SQL Server Setup Help Overview of SQL Server Installation

# 

#### - -

The SQL Server Installation Wizard is Windows Installer-based. It provides a single feature tree for installation of all SQL Server components:

- The Database Engine
- Analysis Services
- Reporting Services
- Integration Services
- Management tools
- Documentation and samples

**Note** SQL Server is available in 32-bit and 64-bit editions. For more information, see <u>Planning a SQL Server Installation</u>.

Whether you use the SQL Server Installation Wizard or the command prompt to install SQL Server, the Setup process involves the following steps:

### <sup>□</sup> In This Section

#### Planning a SQL Server Installation

Describes how to prepare your computer for SQL Server:

- Review hardware and software requirements.
- Review System Configuration Checker requirements and blocking issues.
- Review security considerations for a SQL Server installation.

#### Installing SQL Server 2008

Describes the installation options for SQL Server:

- Install by using the Installation Wizard.
- Install from the command prompt.

#### Configuring SQL Server 2008

Describes how to configure SQL Server by using graphical and command prompt utilities.

### <sup>□</sup> Related Sections

Considerations for Installing the SQL Server Database Engine

Describes how to install and configure the SQL Server Database Engine.

**Considerations for Installing Analysis Services** 

Describes how to install and configure SQL Server Analysis Services.

Considerations for Installing Reporting Services

Describes how to install and configure SQL Server Reporting Services.

**Considerations for Installing Integration Services** 

Describes how to install and configure SQL Server Integration Services.

Considerations for Installing SQL Server Replication

Describes how to install and configure SQL Server Replication.

Considerations for Installing SQL Server Management Tools

Describes how to install and configure SQL Server management tools.

Considerations for Installing SQL Server Samples and Sample Databases

Describes how to install and configure SQL Server samples.

### $\Box$ See Also

### Tasks

How to: Install SQL Server 2008 (Setup)

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help What's New in SQL Server Installation



- -

<u>Overview of SQL Server Installation</u> >

SQL Server 2008 has new Setup architecture for the following scenarios: installation, upgrade, maintenance, failover clustering, and command prompt installations.

### □ See Also

#### Tasks

How to: Install SQL Server 2008 (Setup) How to: Upgrade to SQL Server 2008 (Setup) How to: Create a New SQL Server Failover Cluster (Setup)

#### Concepts

<u>Planning a SQL Server Installation</u> <u>Using Upgrade Advisor to Prepare for Upgrades</u> <u>How to: Install SQL Server 2008 from the Command Prompt</u>

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Quick-Start Installation of SQL Server 2008

## See Also

#### - ▼

#### <u>Overview of SQL Server Installation</u> >

The SQL Server Installation Wizard provides a single feature tree for installation of all the following SQL Server components:

- Database Engine
- Analysis Services
- Reporting Services
- Integration Services
- Replication
- Management tools
- Connectivity components
- Sample databases, samples, and SQL Server Books Online

These additional topics document other ways to install SQL Server 2008:

- To install SQL Server from the command prompt, see <u>How to: Install</u> <u>SQL Server 2008 from the Command Prompt</u>.
- To install a failover cluster, see <u>How to: Create a New SQL Server</u> <u>Failover Cluster (Setup)</u>.
- To upgrade an existing instance of SQL Server to SQL Server 2008, see <u>How to: Upgrade to SQL Server 2008 (Setup</u>).

## <sup>D</sup> Prerequisites

Before you install SQL Server, review topics in <u>Planning a SQL Server</u> <u>Installation</u>.

### <sup>ℤ</sup>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.

### <sup>©</sup> Procedures

To install SQL Server 2008

1.

Insert the SQL Server installation media, and from the root folder, double-click setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click setup.exe.

• If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.

• Windows Installer 4.5 is also required, and may be installed by the Installation Wizard. If you are prompted to restart your computer, restart and then launch the SQL Server 2008 setup.exe again.

• Once prerequisites are installed, the Installation Wizard will launch the SQL Server Installation Center. To create a new installation of SQL Server 2008, click "New Installation or Add Features to an Existing Installation."

• The System Configuration Checker will run a discovery operation on your computer. To continue, click Ok. At this point, Setup log files have been created for your installation. For more information about log files, see <u>How to: View</u> <u>SQL Server 2008 Setup Log Files</u>.

• On the Product key page, click a radio button to indicate whether you are installing a free edition of SQL Server, or whether you have a PID key for a production version of the product. For more information, see <u>Editions and</u> <u>Components of SQL Server 2008</u>.

• On the License Terms page, read the license agreement, and then select the check box to accept the licensing terms and conditions. To continue, click **Next**. To end Setup, click **Cancel**.

• The Installation Wizard will install SQL Server prerequisites if they are not already on the computer. They include the following:

- .NET Framework 2.0
- SQL Server Native Client
- SQL Server Setup Support Files

To install prerequisites, click **Install**.

• The System Configuration Checker will verify the system state of your computer before Setup continues. After the check is complete, click Next to continue.

• On the Feature Selection page, select the components for your installation. A description for each component group appears in the right-hand pane after you select the feature name. You can select any combination of check boxes. For more information, see Editions and Components of SQL Server 2008.

You can also specify a custom directory for shared components using the field at the bottom of this page. To change the installation path for shared components, either update the path name in the field provided at the bottom of the dialog box, or click **Browse** to navigate to an installation directory. The default installation path is C:\Program Files\Microsoft SQL Server\.

• On the Instance Configuration page, specify whether to install a default or a named instance. For more information, see <u>Instance Configuration</u>. To continue, click Next.

**Instance ID suffix** - By default, the instance name is used as the Instance ID suffix. This is used to identify installation directories and registry keys for your instance of SQL Server. This is the case for default instances and named instances. For a default instance, the instance name and instance ID suffix would be MSSQLSERVER. To use a non-default instance ID suffix, select the **Instance ID suffix** check box and provide a value.

### <sup>I</sup>∕ Note:

Typical stand-alone instances of SQL Server 2008, whether default or named instances, do not use a non-default value for the **Instance ID suffix** check box.

**Instance root directory** - By default, the instance root directory is C:\Program Files\Microsoft SQL Server\. To specify a non-default root directory, use the field provided, or click **Browse** to locate an installation folder.

All SQL Server service packs and upgrades will apply to every component of an instance of SQL Server.

**Detected instances and features -** The grid will show instances of SQL Server that are on the computer where Setup is running. If a default instance is already installed on the computer, you must install a named instance of SQL Server 2008. To continue, click **Next**.

• The Disk Space Requirements page calculates the required disk space for the features you specify. It then compares requirements to the available disk space. For more information, see <u>Disk Cost Summary</u>.

• Work flow for the remainder of this topic depends on the features you have specified for your installation. You might not see all of the pages, depending on your selections.

• On the Server Configuration — Service Accounts page, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features you selected to install.

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, are started manually, or are disabled. Microsoft recommends that you configure service accounts individually to provide least privileges for each service, where SQL Server services are granted the minimum permissions they need to complete their tasks. For more information, see <u>SQL Server Configuration - Service Accounts</u> and <u>Setting Up Windows Service Accounts</u>.

To specify the same logon account for all service accounts in this instance of SQL Server, provide credentials in the fields at the bottom of the page.

**Security Note** Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

• Use the **Server Configuration - Collation** tab to specify non-default collations for the Database Engine and Analysis Services. For more information, see <u>SQL Server Configuration - Collation</u>.

• Use the Database Engine Configuration - Account Provisioning page to specify the following:

- Security Mode select Windows Authentication or Mixed Mode Authentication for your instance of SQL Server. If you select Mixed Mode Authentication, you must provide a strong password for the built-in SQL Server system administrator account.
  After a device establishes a successful connection to SQL Server, the security mechanism is the same for both Windows Authentication and Mixed Mode. For more information, see <u>Database Engine Configuration -Account Provisioning</u>.
- SQL Server Administrators You must specify at least one system administrator for the instance of SQL Server. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for the instance of SQL Server. For more information, see <u>Database Engine Configuration Account Provisioning</u>.

When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

• Use the Database Engine Configuration - Data Directories page to specify non-default installation directories. To install to default directories, click **Next**.

#### <sup>I</sup> Important:

If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.

For more information, see <u>Database Engine Configuration - Data Directories</u>.

• Use the Database Engine Configuration - FILESTREAM page to enable FILESTREAM for your instance of SQL Server. For more information about FILESTREAM, see <u>Database Engine Configuration - Filestream</u>. To continue, click Next.

• Use the Analysis Services Configuration - Account Provisioning page to specify users or accounts that will have administrator permissions for Analysis Services. You must specify at least one system administrator for Analysis Services. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for Analysis Services. For more information, see <u>Analysis Services Configuration - Account Provisioning</u>.

When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

• Use the Analysis Services Configuration - Data Directories page to specify non-default installation directories. To install to default directories, click **Next**.

#### <sup>I</sup> Important:

If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.

For more information, see <u>Analysis Services Configuration - Data Directories</u>.

• Use the Reporting Services Configuration page to specify the type of Reporting Services installation to create. Options include the following:

- Native mode default configuration
- SharePoint mode default configuration
- Unconfigured Reporting Services installation

For more information about Reporting Services configuration modes, see <u>Report Server Installation Options</u>.

• On the **Error and Usage Reporting** page, specify the information you would like to send to Microsoft that will help to improve SQL Server. By default, options for error reporting and feature usage are enabled. For more information, see <u>Error and Usage Report Settings</u>.

• The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified.

• The Ready to Install page displays a tree view of installation options that were specified during Setup. To continue, click **Install**.

• During installation, the Installation Progress page provides status so you can monitor installation progress as Setup proceeds.

• After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.

• If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For information about Setup log files, see <u>How to: View SQL Server 2008 Setup Log Files</u>.

### <sup>□</sup> Next Steps

Configure your new SQL Server installation

- To reduce the attackable surface area of a system, SQL Server selectively installs and activates key services and features. For more information, see <u>Understanding Surface Area Configuration</u> on MSDN.
- For more information about using graphical tools and command prompt utilities, see <u>Configuring SQL Server 2008</u>.

### $\Box$ See Also

#### Tasks

How to: Upgrade to SQL Server 2008 (Setup)

#### Concepts

How to: View SQL Server 2008 Setup Log Files How to: Install SQL Server 2008 from the Command Prompt

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Editions and Components of SQL Server 2008

# 

#### <u>Overview of SQL Server Installation</u> >

- -

Installation requirements vary based on your application needs. The different editions of SQL Server accommodate the unique performance, runtime, and price requirements of organizations and individuals. The SQL Server components that you install also depend on your specific requirements. The following sections help you understand how to make the best choice among the editions and components available in SQL Server 2008.

### <sup>□</sup> Server Editions of SQL Server 2008

The following table describes the server editions of SQL Server 2008:

Term	Definition
Enterprise	SQL Server Enterprise is a comprehensive data platform that provides
	enterprise-class scalability, performance, high availability, and
x64 and	advanced business intelligence capabilities for running secure,
	business-critical applications. For more information, see <u>Features</u>
IA64)*	Supported by the Editions of SQL Server 2008.
	SQL Server Standard is a complete data platform that provides ease
	of use and manageability. This includes built-in business intelligence
Standard	capabilities to run departmental applications. For more information,
(x86 and	see <u>Features Supported by the Editions of SQL Server 2008</u> .
(XOO allu x64)	
л0 <del>4</del> )	SQL Server Standard for Small Business includes all of the technical
	components and capabilities of SQL Server Standard and is licensed
	to operate in a small business environment of 75 or fewer computers.

<sup>1</sup>Enterprise is available in SQL Server 2008 Enterprise Evaluation for a 180-day trial period. For more information, see the <u>SQL Server Web site</u>.

## <sup>□</sup> Specialized Editions of SQL Server 2008

Specialized editions of SQL Server 2008 are engineered for specific customer scenarios. The following table describes these editions of SQL Server 2008:

Term	Definition
SQL Server 2008 Developer (x86, x64, and IA64)	SQL Server 2008 Developer lets developers build any kind of application on top of SQL Server. It includes all the functionality of SQL Server 2008 Enterprise, but is licensed for use as a development and test system, not as a production server. SQL Server 2008 Developer is an ideal choice for people who build and test applications. You can upgrade SQL Server 2008 Developer for production use.
Workgroup (x86 and x64)	SQL Server Workgroup is ideal for running branch location databases providing a reliable data management and reporting platform that includes secure, remote synchronization and management capabilities.
Web (x86, x64)	SQL Server 2008 Web is a low total-cost-of-ownership option for Web hosts and Web sites that provide scalability and manageability capabilities for small to large scale Web properties.
SQL Server Express (x86 and x64)	The SQL Server Express database platform is based on SQL Server 2008. It is also a replacement for Microsoft Desktop Engine (MSDE). Integrated with Visual Studio, 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 with Advanced Services (x86 and x64)	SQL Server Express is free and can be redistributed by ISV's (subject to agreement). SQL Server Express is ideal for learning and building desktop and small server applications. This edition is the best choice for independent software vendors, non-professional developers, 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.
Compact 3.5 SP1	SQL Server Compact is a free, embedded database ideal for building

(x86)	stand-alone and occasionally connected applications for mobile
Compact 3.1 (x86)	devices, desktops, and web chefts on an windows platforms.

For more information about the specialized editions of SQL Server 2008, see the <u>SQL Server Web site</u>.

#### Using SQL Server with an Internet Server

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

### <sup>I</sup>Note:

Although you can install an instance of SQL Server on a computer that is 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. For more information about software requirements for SSRS, see <u>Hardware and Software Requirements for Installing SQL Server</u> 2008. For more information about federations, see the **Federated Database Servers** topic in SQL Server Books Online.

### **Using SQL Server with Client/Server Applications**

You can install just the SQL Server client components on a computer that is 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, and SQL Server Books Online. For more information, see <u>How to: Install SQL Server 2008</u> <u>(Setup)</u>.

### <sup>Deciding Among SQL Server Components</sup>

Use the Feature Selection page of the SQL Server Installation Wizard to select the components to include in an installation of SQL Server. By default, none of the features in the tree are selected.

Use the information in the following tables to determine the set of features that best fits your needs:

Server components	Description
SOL Server	SOL Server Database Engine includes the Database Engine, the
Database	core service for storing, processing, and securing data, Replication,
Engine	full-text search, and tools for managing relational and XML data.
Analysis	Analysis Services includes the tools for creating and managing
Services	online analytical processing (OLAP) and data mining applications.
	Reporting Services includes server and client components for
Reporting	creating, managing, and deploying tabular, matrix, graphical, and
Services	free-form reports. Reporting Services is also an extensible platform
	that you can use to develop report applications.
Integration	Integration Services is a set of graphical tools and programmable
Services	objects for moving, copying, and transforming data.

Management tools	Description
SQL Server Management Studio	SQL Server Management Studio is an integrated environment to access, configure, manage, administer, and develop components of SQL Server. Management Studio lets developers and administrators of all skill levels use SQL Server. Internet Explorer 6 SP1 or a later version is required for Management Studio installation.
SQL Server	SQL Server Configuration Manager provides basic configuration
Configuration	management for SQL Server services, server protocols, client
Manager	protocols, and client aliases.
SQL Server	SQL Server Profiler provides a graphical user interface to monitor
Profiler	an instance of the Database Engine or Analysis Services.

Database Engine Tuning Advisor	Database Engine Tuning Advisor helps create optimal sets of indexes, indexed views, and partitions.
Business	The Business Intelligence Development Studio is an IDE for
Intelligence	Analysis Services, Reporting Services, and Integration Services
Development	solutions. Internet Explorer 6 SP1 or a later version is required for
Studio	BI Development Studio installation.
Connectivity Components	Installs components for communication between clients and servers, and network libraries for DB-Library, ODBC, and OLE DB.

Documentation	Description	
SQL Server Books Online	Core documentation for SQL Server.	

### $\Box$ See Also

### Tasks

How to: Install SQL Server 2008 (Setup)

### Concepts

Planning a SQL Server Installation

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Features Supported by the Editions of SQL Server 2008

# 

<u>Overview of SQL Server Installation</u> >

Use the following tables to determine which features are supported by the various editions of Microsoft SQL Server 2008.

### <sup>ℤ</sup>Note:

- -

SQL Server Enterprise is available in an Evaluation edition for a 180-day trial period. For more information about how to obtain an evaluation copy of SQL Server 2008, see this <u>Microsoft Web site</u>.

## <sup>□</sup> Scalability

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
Partitioning	Yes					
Data compression	Yes					
Resource governor	Yes					
Partition table parallelism	Yes					

# <sup>□</sup> High Availability

Enterprise	Standard	Workgroup	Web	Express	Express Advanced
50	16	16	16	16	16
Yes	Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes		
Yes (full)	Yes (safety full only)	Witness only	Witness only	Witness only	Witness only
OS maximum <sup>1</sup>	2 nodes				
Yes	Yes				
Yes	Yes				
Yes	Yes				
Yes					
	Enterprise 50 Yes Yes Yes (full) OS maximum <sup>1</sup> Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes	EnterpriseStandard5016YesYesYesYesYesYesYesYesYes2 nodesYes	EnterpriseStandardWorkgroup501616501616YesYesYesYesYesYesYes (full)YesMitness onlyYes (full)2 nodes7YesYes7YesYes7YesYes7YesYes7YesYes7YesYes1Yes1	EnterpriseStandardWorkgroupWeb5016161650161616YesYesYesYesYesYesYesYesYes (full)YesWitnessWitnessYes (full)Yes210YesYes2101YesYes2101YesYes2101YesYes2101YesYesSS1YesYesSS1YesI1011Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes10111Yes1011 <td< td=""><td>EnterpriseStandardWorkgroupWebExpress5016161616YesYesYesYesYesYesYesYesYesYesYes (full)YesWitness onlyWitness onlyWitness onlyYes (full)YesIIYesYesIIYesYesIIYesYesIIYesYesIIYesYesIIYesYesIIYesI<t< td=""></t<></td></td<>	EnterpriseStandardWorkgroupWebExpress5016161616YesYesYesYesYesYesYesYesYesYesYes (full)YesWitness onlyWitness onlyWitness onlyYes (full)YesIIYesYesIIYesYesIIYesYesIIYesYesIIYesYesIIYesYesIIYesI <t< td=""></t<>

Backup	Yes			
compression				

<sup>1</sup>Windows Server 2003 supports a maximum of 8 failover cluster nodes. Windows Server 2008 supports a maximum of 64 failover cluster nodes.

## <sup>□</sup> Security

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
C2 compliant tracing	Yes	Yes	Yes	Yes	Yes	Yes
SQL auditing foundation	Yes	Yes	Yes	Yes	Yes	Yes
Fine grained auditing	Yes					
Transparent database encryption	Yes					
ISV encryption (off- box key management)	Yes					

# <sup>□</sup> Replication

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advance
Merge replication	Yes	Yes	Subscriber only	Subscriber only	Subscriber only	Subscribe only
Transactional replication	Yes	Yes	Subscriber only	Subscriber only	Subscriber only	Subscribe only
Snapshot replication	Yes	Yes	Subscriber only	Subscriber only	Subscriber only	Subscrib only
SQL Server change tracking	Yes	Yes	Yes	Yes	Yes	Yes
Heterogeneous subscribers	Yes	Yes				
Oracle publishing	Yes					
P2P transactional replication	Yes					

## <sup>□</sup> Manageability

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
User instances					Yes	Yes
Dedicated admin connection	Yes	Yes	Yes	Yes	Yes (Under trace flag)	Yes (Under trace flag)
Policy-based configuration	Yes	Yes	Yes	Yes	Yes	Yes
Policy-based management	Yes	Yes	Yes	Yes	Yes	Yes
Performance data collection and warehouse	Yes	Yes	Yes	Yes		
Standard performance reports	Yes	Yes				
Plan guides	Yes	Yes				
Plan freezing for plan guides	Yes	Yes				
Policy-based best practices	Yes	Yes				
Multi-server policy- based management	Yes	Yes				
Distributed partition views	Yes					
Parallel index operations	Yes					
Automatic query-to- indexed-view matching	Yes					
Parallel database backup checksum check	Yes					

Database mail	Yes	Yes	Yes	Yes		
Database migration tools	Yes	Yes	Yes	Yes	Yes	Yes

# <sup>□</sup> Management Tools

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
SQL management objects (SMO)	Yes	Yes	Yes	Yes	Yes	Yes
SQL Configuration Manager	Yes	Yes	Yes	Yes	Yes	Yes
SQL CMD (command prompt tool)	Yes	Yes	Yes	Yes	Yes	Yes
SQL Server Management Studio	Yes	Yes	Yes	Yes (Express version)		Yes (Express version)
SQL Profiler	Yes	Yes	Yes	Yes		
SQL Server Agent	Yes	Yes	Yes	Yes		
Database Tuning Advisor	Yes	Yes	Yes	Yes		
Microsoft Operations Manager Pack	Yes	Yes	Yes	Yes		

# <sup>□</sup> Development Tools

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
Microsoft Visual Studio Integration	Yes	Yes	Yes	Yes	Yes	Yes
SQL query and edit and design tools	Yes	Yes	Yes			
Intellisense (Transact-SQL and MDX)	Yes	Yes	Yes			
Version control support	Yes	Yes	Yes			
Business Intelligence Development Studio	Yes	Yes				
MDX edit, debug, and design tools	Yes	Yes				

# <sup>©</sup> Programmability

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
Common language runtime (CLR) integration	Yes	Yes	Yes	Yes	Yes	Yes
Native XML support	Yes	Yes	Yes	Yes	Yes	Yes
XML indexing	Yes	Yes	Yes	Yes	Yes	Yes
MERGE and UPSERT capabilities	Yes	Yes	Yes	Yes	Yes	Yes
FILESTREAM support	Yes	Yes	Yes	Yes	Yes	Yes
Date and Time data types	Yes	Yes	Yes	Yes	Yes	Yes
Internationalization support	Yes	Yes	Yes	Yes	Yes	Yes
Full-text search	Yes	Yes	Yes	Yes		Yes
Specification of language in query	Yes	Yes	Yes	Yes		Yes
Service broker (messaging)	Yes	Yes	Yes	Client only	Client only	Client only
XML/A support	Yes	Yes				
Web services (HTTP/SOAP endpoints)	Yes	Yes				

# □ Spatial and Location Services

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
Spatial indexes	Yes	Yes	Yes	Yes	Yes	Yes
Geodetic data type	Yes	Yes	Yes	Yes	Yes	Yes
Advanced spatial libraries	Yes	Yes	Yes	Yes	Yes	Yes
Standards-based spatial support	Yes	Yes	Yes	Yes	Yes	Yes
# Integration Services

Feature	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
SQL Server Import and Export Wizard with basic sources and destinations and Execute SQL task	Yes	Yes	Yes	Yes	Yes	Yes
Integration Services runtime	Yes	Yes	Yes	Yes	Yes	Yes
Integration Services API and object model	Yes	Yes	Yes	Yes		
SSIS Designer including VSTA scripting	Yes	Yes				
Integration Services service, wizards, and command prompt utilities	Yes	Yes				
Basic tasks and transformations	Yes	Yes				
Log providers and logging	Yes	Yes				
Data profiling tools	Yes	Yes				
Additional sources and destinations: Raw File source XML source DataReader destination	νρς	νος				

Raw File destination				
Recordset				
destination				
SQL Server				
Compact destination				
SQL Server destination				
Advanced sources,				
transformations, and				
destinations:				
Data Mining Query				
transformation				
Fuzzy Lookup and				
Fuzzy Grouping				
transformations				
Term Extraction and	N 7			
Term Lookup	Yes			
transformations				
Data Mining Model				
Training destination				
Dimension				
Processing				
destination				
Partition Processing				
destination				

# <sup>□</sup> Data Warehouse Creation

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
Create cubes without a database	Yes	Yes				
Auto-generate staging and data warehouse schema	Yes	Yes				
Attribute relationship designer	Yes	Yes				
Efficient aggregation designers	Yes	Yes				

## <sup>□</sup> Data Warehouse Scale and Performance

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
Change data capture	Yes					
Star join query optimization	Yes					
Scalable read-only AS configuration	Yes					
Proactive caching	Yes					
Auto parallel partition processing	Yes					
Partitioned cubes	Yes					
Distributed partitioned cubes	Yes					

# <sup>©</sup> Multi-Dimensional Analytics

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
SQL Server Analysis Services service	Yes	Yes				
SQL Server Analysis Services backup	Yes	Yes				
General performance/scale improvements	Yes	Yes				
Dimension, attribute relationship, aggregate, and cube design improvements	Yes	Yes				
Personalization extensions	Yes	Yes				
Financial aggregations	Yes					
Partitioned customers	Yes					
Custom rollups	Yes					
Semi-additive measures	Yes					
Writeback dimensions	Yes					
Linked measures and dimensions	Yes					
Binary and compressed XML transport	Yes					
Account intelligence	Yes					

Perspectives	Yes			
Analysis Services				
shared, scalable	Yes			
databases				

# <sup>□</sup> Data Mining

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
Standard algorithms	Yes	Yes				
Data mining tools: wizards, editors, query builders	Yes	Yes				
Cross validation	Yes					
Models on filtered subsets of mining structure data	Yes					
Time series: custom blending between ARTXP and ARIMA models	Yes					
Time series: prediction with new data	Yes					
Unlimited concurrent datamining queries	Yes					
Advanced configuration and tuning for algorithms	Yes					
Algorithm plug-in API	Yes					
Parallel model processing	Yes					
Time series: cross- series prediction	Yes					
Unlimited attributes for association rules	Yes					
Sequence prediction	Yes					

Multiple prediction				
targets for naïve	Yes			
Bayes, neural				
network, and logistic				
regression				

# <sup>E</sup> Reporting

Feature Name	Enterprise	Standard	Workgroup	Web	Express	Express Advanced
Report server	Yes	Yes	Yes	Yes		Yes
Report Designer	Yes	Yes	Yes	Yes		Yes
Report Manager	Yes	Yes	Yes	Yes (Report Manager)		Yes (Report Manager)
Role-based security	Yes	Yes	Yes (Fixed roles)	Yes (Fixed roles)		Yes (Fixed roles)
Ad-hoc reporting (Report builder)	Yes	Yes	Yes			
Word export and enhanced text formatting	Yes	Yes	Yes	Yes		Yes
Enterprise- scale reporting engine	Yes	Yes	Yes	Yes		Yes
IIS-agnostic report deployment	Yes	Yes	Yes	Yes		Yes
Updated management tools	Yes	Yes	Yes	Yes		Yes
Report definition customization extension (RDCE)	Yes	Yes	Yes	Yes		Yes
SharePoint integration	Yes	Yes				

Enhanced SSRS gauges and charting	Yes	Yes	Yes	Yes	Yes
Custom authentication	Yes	Yes	Yes	Yes	
Export to Excel, PDF, and images	Yes	Yes	Yes	Yes	Yes
Remote and non-relational data source support	Yes	Yes			
E-mail and file share delivery	Yes	Yes			
Report history, scheduling, subscriptions, and caching	Yes	Yes			
Data source, delivery, and rendering extensibility	Yes	Yes			
Scale out (Web farms)	Yes				
Infinite click- through	Yes				
Data-driven subscriptions	Yes				
Reporting Services memory limits	Unlimited	Unlimited	4 GB	4 GB	4 GB

## <sup>□</sup> See Also

#### Reference

Version and Edition Upgrades

#### Concepts

Overview of SQL Server Installation Quick-Start Installation of SQL Server 2008

#### **Other Resources**

Product Specifications for SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help About the SQL Server License Terms



#### - ▼

<u>Overview of SQL Server Installation</u> >

The Microsoft Software License Terms are copied to the local computer when SQL Server is installed. When multiple instances of the same edition and language of SQL Server are installed on the same computer, a single copy of the license terms will apply to all instances of that edition and language. The files are written to: %ProgramFiles%\Microsoft SQL Server\100\EULA\ or %ProgramFiles(x86)%\Microsoft SQL Server\100\EULA\.

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

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

- License\_DEV\_ENU.txt for SQL Server 2008 Developer, English language edition.
- License\_STD\_ITA.txt for SQL Server Standard, Italian language edition.
- License\_UA\_JPN.txt for SQL Server 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 editions:

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

The following file name segments are used to identify SQL Server 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

# <sup>□</sup> SQL Server Licensing Options

For information about SQL Server pricing and licensing options, see the <u>Microsoft SQL Server</u> Web site.

## $\Box$ See Also

### Concepts

SQL Server Privacy Statement Legal Notice for Documentation

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help SQL Server Privacy Statement

<u>Overview of SQL Server Installation</u> >

For information about user data collection components in SQL Server 2008, see the <u>SQL Server 2008 privacy statement</u>.

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Legal Notice for Documentation

#### <u>Overview of SQL Server Installation</u> >

This documentation, including sample applications herein, is provided for informational purposes only, and Microsoft and its suppliers make no warranties, either express or implied, in this documentation. 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.

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.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, ActiveX, Excel, IntelliSense, MSDN, MS-DOS, PivotChart, PivotTable, PowerPoint, SharePoint, SQL Server, Visual Basic, Visual C#, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, and Windows Vista 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 Support and Upgrade

SQL Server 2008 software documentation includes SQL Server Books Online, readme files, known issues documents, sample code, sample databases, and Knowledge Base articles published by Microsoft- all contain content about SQL Server features and functionality. This support and upgrade policy applies to SQL Server 2008 documentation, including readme files for SQL Server 2008 releases and service packs; a readme file is considered an extension of SQL Server 2008 Books Online.

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.

In some cases, a particular feature in Microsoft SQL Server 2008 has not been sufficiently tested or developed and is not intended for your use. Such features are often there as placeholders. Therefore, certain features that are included in the software are not documented. Features that are not documented or that are included as a notation for Microsoft internal use are not intended for your use and are not supported by Microsoft Customer Support Services. Unless the Microsoft SQL Server 2008 documentation states otherwise, third party content, including issues that result from using such third party content with Microsoft SQL Server 2008, are not supported by Microsoft Customer Support Services and may also result in loss of data. Microsoft Customer Support Services also does not provide support for databases or applications that leverage or use undocumented entry points like undocumented APIs, including bit not limited to: stored procedures, extended stored procedures, functions, views, tables, columns, properties, or metadata.

Server and database upgrades to future versions of SQL Server 2008 for applications and databases that leverage and use undocumented entry points also may not work and are not supported. Use of SQL Server features and functionality included in Microsoft SQL Server that are not documented in the SQL Server 2008 documentation for your use are not supported and such use may cause loss of data.

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Licensing Considerations for SQL Server 2008

-

<u>Overview of SQL Server Installation</u> >

For information about SQL Server 2008 licensing options, see <u>this Microsoft</u> <u>SQL Server Web site</u>.

## □ See Also

#### Reference

About the SQL Server License Terms

### Concepts

SQL Server Privacy Statement Legal Notice for Documentation

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Planning a SQL Server Installation



#### - -

To install SQL Server 2008, follow these steps:

- Review installation requirements, system configuration checks, and security considerations for a SQL Server 2008 installation.
- Run SQL Server Setup to install or upgrade to SQL Server 2008.
- Use SQL Server utilities to configure SQL Server.

Regardless of whether you use the SQL Server Installation Wizard or the command prompt to install SQL Server, the following topics document the SQL Server planning process.

## □ In This Section

Hardware and Software Requirements for Installing SQL Server 2008

Describes SQL Server hardware and software requirements, operating system support, network and Internet considerations, and hard disk space requirements.

Check Parameters for the System Configuration Checker

Describes requirements for the SQL Server System Configuration Checker.

Security Considerations for a SQL Server Installation

Describes security considerations for a SQL Server installation.

Editions and Components of SQL Server 2008

Describes SQL Server editions and components.

Upgrading to SQL Server 2008

Describes options for upgrading to SQL Server.

Migrating to SQL Server 2008

Describes options for migrating to SQL Server.

Configuring SQL Server 2008

Describes SQL Server configuration tools.

## <sup>□</sup> Related Sections

Considerations for Installing the SQL Server Database Engine

This section describes how to install and configure the SQL Server Database Engine.

**Considerations for Installing Analysis Services** 

This section describes how to install and configure SQL Server Analysis Services.

**Considerations for Installing Reporting Services** 

This section describes how to install and configure SQL Server Reporting Services.

**Considerations for Installing Integration Services** 

This section describes how to install and configure Integration Services.

Considerations for Installing SQL Server Replication

This section describes how to install and configure SQL Server Replication.

Getting Started with SQL Server 2008 Failover Clustering

Describes scenarios for SQL Server failover clustering.

Considerations for Installing SQL Server Management Tools

This section describes how to install and configure SQL Server management tools.

Considerations for Installing SQL Server Samples and Sample Databases

This section describes how to install AdventureWorks samples.

## <sup>□</sup> See Also

#### Tasks

How to: Upgrade to SQL Server 2008 (Setup)

#### Concepts

<u>Quick-Start Installation of SQL Server 2008</u> <u>How to: Install SQL Server 2008 from the Command Prompt</u> <u>Before Installing Failover Clustering</u>

#### **Other Resources**

High Availability Solutions Overview

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

Hardware and Software Requirements for Installing SQL Server 2008



- -

<u>Planning a SQL Server Installation</u> >

The following sections list the minimum hardware and software requirements to install and run SQL Server 2008.

For both 32-bit and 64-bit editions of SQL Server 2008, the following apply:

- SQL Server 2008 Enterprise is available for evaluation during a 180-day trial period. For more information, see the <u>SQL Server: How to Buy</u> Web site.
- Microsoft recommends that you run SQL Server 2008 on computers with the NTFS file format. For upgrades to SQL Server 2008, FAT32 file systems will not be blocked.
- SQL Server Setup will block installations on read-only or compressed drives.
- For information about using SQL Server 2008 tools to prepare for an upgrade to SQL Server 2008, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.
- SQL Server does not install the .NET Framework 3.5 software development kit (SDK). However, the SDK contains tools that are useful when you use the .NET Framework for SQL Server development. You can download the .NET Framework SDK from the <u>.NET Framework</u> Web site.
- Requirements to restart computers during SQL Server Setup: Installation of the .NET Framework requires a restart of the operating system. If Windows Installer installation also requires a restart, Setup will wait until the .NET Framework and Windows Installer components have installed before restarting.
- For more information about features of SQL Server 2008, see the following:
  - Editions and Components of SQL Server 2008

• Features Supported by the Editions of SQL Server 2008

# <sup>□</sup> Hardware and Software Requirements

The following requirements apply to all SQL Server 2008 installations:

Component	Requirement
Framework <sup>2</sup>	<ul> <li>SQL Server Setup installs the following software components required by the product:</li> <li>.NET Framework 3.5<sup>1</sup></li> <li>SQL Server Native Client</li> <li>SQL Server Setup support files</li> </ul>
Software <sup>2</sup>	SQL Server Setup requires Microsoft Windows Installer 4.5 or a later version, and Microsoft Data Access Components (MDAC) 2.8 SP1 or a later version. You can download MDAC 2.8 SP1 from the <u>MDAC downloads</u> Web site. After installing required components, SQL Server Setup will verify that the computer where SQL Server 2008 will be installed also meets all the other requirements for a successful installation. For more information, see <u>Check Parameters for the System</u> <u>Configuration Checker</u> .
Network Software	<ul> <li>Network software requirements for the 64-bit versions of SQL Server 2008 are the same as the requirements for the 32-bit versions.</li> <li>Supported operating systems have built-in network software. Stand-alone named and default instances support the following network protocols: <ul> <li>Shared memory</li> <li>Named Pipes</li> <li>TCP/IP</li> <li>VIA</li> </ul> </li> <li>Note Shared memory and VIA are not supported on failover clusters.</li> </ul>

Virtualization	SQL Server 2008 is supported in virtual machine environments running on the Hyper-V role in Windows Server 2008 Standard, Enterprise and Data Center editions. The virtual machine must run an operating system supported for the specific SQL Server 2008 edition listed later in this topic. In addition to resources required by the parent partition, each virtual machine (child partition) must be provided with sufficient processor resources, memory, and disk resources for its SQL Server 2008 instance. Requirements are listed later in this topic. <sup>3</sup> Within the Hyper-V role on Windows Server 2008, a maximum of four virtual processors can be allocated to virtual machines running Windows Server 2008 32-bit or 64-bit editions. A maximum of 2 virtual processors can be allocated to virtual computers that are running Windows Server 2003 32-bit editions. For virtual computer that host other operating systems, a
	maximum of one virtual processor can be allocated to virtual computers.
	Notes:
	<ul> <li>It is recommended that SQL Server 2008 be shut down before shutting down or the virtual machine.</li> <li>Guest failover clustering (configuring failover clustering in SQL Server 2008) is not supported in a Hyper-V</li> </ul>
	<ul> <li>For more information about the Hyper-V role in Windows Server 2008, see the <u>Windows Server 2008 Web site</u>.</li> </ul>
Internet Software	Microsoft Internet Explorer 6 SP1 or a later version is required for all installations of SQL Server 2008. Internet Explorer 6 SP1 or a later version is required for Microsoft Management Console (MMC), SQL Server Management Studio, Business Intelligence Development Studio, the Report Designer component of Reporting Services, and HTML Help.
Hard Disk	Disk space requirements will vary with the SQL Server 2008 components you install. For more information, see <u>Hard disk</u> <u>space requirements</u> later in this topic.

Drive	A CD or DVD drive, as appropriate, is required for installation from disc.
Display	SQL Server 2008 graphical tools require VGA or higher resolution: at least 1,024x768 pixel resolution.
Other	Pointing device: A Microsoft mouse or compatible pointing
Devices	device is required.

<sup>1</sup>The following .NET Framework versions are required:

- SQL Server 2008 on Windows Server 2003 (64-bit) IA64 .NET Framework 2.0 SP1
- All other editions of SQL Server 2008 .NET Framework 3.5

Installation of .NET Framework requires a restart of the operating system. If Windows Installer installation also requires a restart, Setup will wait until .NET Framework and Windows Installer components have installed before restarting.

<sup>2</sup> SQL Server Setup will not install the following required components for SQL Server Express and SQL Server Express with Advanced Services. You must install these components manually before you run SQL Server Setup:

- SQL Server Express ..NET Framework 2.0 SP2 and Windows installer 4.5. On Windows Vista, use .NET Framework 3.5 SP1.
- SQL Server Express with Advanced Services .NET Framework 3.5 SP1, Windows Installer 4.5, and Windows PowerShell 1.0.

<sup>3</sup>As with all virtualization technologies, SQL Server 2008 running in a Windows Server 2008 Hyper-V virtual computer will be slower than on a physical computer with the same physical resources.

## Processor, Memory, and Operating System Requirements

To view system requirements, select an edition of SQL Server 2008:

#### Server Editions of SQL Server 2008

- SQL Server 2008 Enterprise (64-bit) IA64
- <u>SQL Server 2008 Enterprise (64-bit) x64</u>
- SQL Server 2008 Standard (64-bit) x64
- <u>SQL Server 2008 Enterprise (32-bit</u>)
- <u>SQL Server 2008 Standard (32-bit</u>)

#### **Specialized Editions of SQL Server 2008**

- SQL Server 2008 Developer (64-bit) IA64
- SQL Server 2008 Developer (64-bit) x64
- <u>SQL Server 2008 Workgroup (64-bit) x64</u>
- SQL Server 2008 Web (64-bit) x64
- SQL Server 2008 Express (64-bit) x64
- SQL Server 2008 Express Advanced (64-bit) x64
- SQL Server 2008 Developer (32-bit)
- SQL Server 2008 Workgroup (32-bit)
- <u>SQL Server 2008 Web (32-bit)</u>
- <u>SQL Server 2008 Express and Express with Advanced Services (32-bit)</u>

#### Additional information in this topic

- Hard disk space requirements
- <u>Cross language support</u>
- Support for installing SQL Server 2008 on a domain controller

## <sup>□</sup> Server Editions (64-bit and 32-bit)

#### SQL Server 2008 Enterprise (64-bit) IA64

The following table shows system requirements for SQL Server 2008 Enterprise (64-bit) IA64. Note that this edition of SQL Server 2008 is not available in the following localized versions: Italian, Spanish, Brazilian (Portuguese), or Russian.

Component	Requirement
Processor	<ul> <li>Processor type:</li> <li>Itanium processor or faster</li> <li>Processor speed:</li> <li>Recommended: 1.0 GHz or faster</li> </ul>
Operating System	Windows Server 2008 64-bit Itanium <sup>1,2</sup> Windows Server 2003 SP2 64-bit Itanium Data Center <sup>1,2</sup> Windows Server 2003 SP2 64-bit Itanium Enterprise <sup>1</sup>
Memory	<ul> <li>RAM:</li> <li>Minimum: 512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>1</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Windows that enables 32-bit applications to run natively in 32-bit mode. Applications function in 32-bit mode, even though the underlying operating system is running on the 64-bit operating system.

<sup>2</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core

installations.

## SQL Server 2008 Enterprise (64-bit) x64

The following table shows system requirements for SQL Server 2008 Enterprise (64-bit) x64:

Component	Requirement
Processor	<ul> <li>Processor type:</li> <li>Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support</li> <li>Processor speed:</li> <li>Minimum: 1.4 GHz</li> <li>Recommended: 2.0 GHz or faster</li> </ul>
	Windows XP Professional 2003 64-bit x64 <sup>3</sup> Windows Server 2003 SP2 64-bit x64 Standard <sup>1</sup> Windows Server 2003 SP2 64-bit x64 Data Center <sup>1</sup>
Operating System	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>1</sup> Windows Vista 64-bit x64 Ultimate <sup>3</sup> Windows Vista 64-bit x64 Home Premium <sup>3</sup> Windows Vista 64-bit x64 Home Basic <sup>3</sup> Windows Vista 64-bit x64 Enterprise <sup>3</sup> Windows Vista 64-bit x64 Business <sup>3</sup> Windows Server 2008 64-bit x64 Standard

	Windows Server 2008 64-bit x64 Standard without Hyper-V <sup>2,4</sup>
	Windows Server 2008 64-bit x64 Data Center
	Windows Server 2008 64-bit x64 Data Center without Hyper-V <sup>1,2</sup>
	Windows Server 2008 64-bit x64 Enterprise,
	Windows Server 2008 64-bit x64 Enterprise without Hyper-V <sup>1,2</sup>
Memory	<ul> <li>RAM:</li> <li>Minimum:512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>1</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Windows that enables 32-bit applications to run natively in 32-bit mode. Applications function in 32-bit mode even though the underlying operating system is running on the 64-bit operating system.

<sup>2</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

<sup>3</sup>This version of Windows supports SQL Server 2008 Enterprise Evaluation, not Enterprise.

#### SQL Server 2008 Standard (64-bit) x64

The following table shows system requirements for SQL Server 2008 Standard (64-bit) x64:

Component	Requirement
	Processor type:
	• Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support

Processor	Processor speed:
	Minimum: 1.4 GHz
	• Recommended: 2.0 GHZ of faster
	Windows XP Professional x64
	Windows Server 2003 SP2 64-bit x64 Standard <sup>1,2</sup>
	Windows Server 2003 SP2 64-bit x64 Data Center <sup>1,2</sup>
	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>1,2</sup>
	Windows Vista Ultimate x64
	Windows Vista Enterprise x64
Operating System	Windows Vista Business x64
	Windows Server 2008 x64 Web <sup>1</sup>
	Windows Server 2008 x64 Standard, Windows Server 2008 x64 Standard without Hyper- $V^{1,2}$
	Windows Server 2008 x64 Data Center, Windows Server 2008 x64 Data Center without Hyper-V <sup>1,2</sup>
	Windows Server 2008 x64 Enterprise, Windows Server 2008 x64 Enterprise without Hyper-V <sup>1,2</sup>
	RAM:
Memory	<ul> <li>Minimum:512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>1</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Microsoft Windows that enables 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 operating system.

<sup>2</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

## SQL Server 2008 Enterprise (32-bit)

The following table shows the system requirements for SQL Server 2008 Enterprise (32-bit):

Component	Requirement
	Processor type:
	• Pentium III-compatible processor or faster
Processor	Processor speed:
	<ul><li>Minimum: 1.0 GHz</li><li>Recommended: 2.0 GHz or faster</li></ul>
	Windows XP Professional SP2 <sup>1</sup>
	Windows XP SP2 Tablet <sup>1</sup>
	Windows XP SP2 Media Center 2002 <sup>1</sup>
	Windows XP SP2 Media Center 2004 <sup>1</sup>
	Windows XP Media Center 2005 <sup>1</sup>
	Windows XP Professional Reduced Media <sup>1</sup>
	Windows Server 2003 Small Business Server R2 Standard
	Windows Server 2003 Small Business Server R2 Premium
	Windows Server 2003 SP2 Standard

	Windows Server 2003 SP2 Enterprise
	Windows Server 2003 SP2 Data Center
	Windows Server 2003 Small Business Server SP2 Standard
	Windows Server 2003 Small Business Server SP2 Premium
	Windows Server 2003 SP2 64-bit x64 Standard <sup>2</sup>
	Windows Server 2003 SP2 64-bit x64 Data Center <sup>2</sup>
	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>2</sup>
	Windows Vista Ultimate <sup>1</sup>
	Windows Vista Home Premium <sup>1</sup>
	Windows Vista Home Basic <sup>1</sup>
Operating System	Windows Vista Enterprise <sup>1</sup>
	Windows Vista Business <sup>1</sup>
	Windows Vista 64-bit x64 Ultimate <sup>1,2</sup>
	Windows Vista 64-bit x64 Home Premium <sup>1,2</sup>
	Windows Vista 64-bit x64 Home Basic <sup>1,2</sup>
	Windows Vista 64-bit x64 Enterprise <sup>1,2</sup>
	Windows Vista 64-bit x64 Business <sup>1,2</sup>
	Windows Server 2008 Standard (with and without Hyper-V)
	Windows Server 2008 Web

	Windows Server 2008 Data Center
	Windows Server 2008 Data Center without Hyper-V
	Windows Server 2008 Enterprise
	Windows Server 2008 Enterprise without Hyper-V
	Windows Server 2008 x64 Standard
	Windows Server 2008 x64 Standard without Hyper-V <sup>2</sup>
	Windows Server 2008 x64 Data Center
	Windows Server 2008 x64 Data Center without Hyper-V <sup>2</sup>
	Windows Server 2008 x64 Enterprise
	Windows Server 2008 x64 Enterprise without Hyper-V <sup>2</sup>
	RAM:
Memory	<ul> <li>Minimum: 512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>1</sup>This version of Windows supports SQL Server 2008 Enterprise Evaluation, not Enterprise.

<sup>2</sup>This edition of SQL Server can be installed to the Windows on Windows (WOW64) 32-bit subsystem of a 64-bit server.

## SQL Server 2008 Standard (32-bit)

The following table shows the system requirements for SQL Server 2008 Standard (32-bit):

Component	Requirement

	Processor type:
	Pentium III-compatible processor or faster
Processor	Processor speed:
	<ul><li>Minimum: 1.0 GHz</li><li>Recommended: 2.0 GHz or faster</li></ul>
	Windows XP Professional SP2
	Windows XP SP2 Tablet
	Windows XP x64 Professional 2003 <sup>1</sup>
	Windows XP SP2 Media Center 2002
	Windows XP SP2 Media Center 2004
	Windows XP Media Center 2005
	Windows XP Professional Reduced Media
	Windows Server 2003 Small Business Server R2 Standard
	Windows Server 2003 Small Business Server R2 Premium
	Windows Server 2003 SP2 Standard
	Windows Server 2003 SP2 Enterprise
	Windows Server 2003 SP2 Data Center
	Windows Server 2003 SP2 Small Business Server Standard
	Windows Server 2003 SP2 Small Business Server Premium
	Windows Server 2003 SP2 64-bit x64 Standard <sup>1</sup>
	Windows Server 2003 SP2 64-bit x64 Data Center <sup>1</sup>

	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>1</sup>
Operating System	Windows Vista Ultimate
	Windows Vista Enterprise
	Windows Vista Business
	Windows Vista Ultimate x64 <sup>1</sup>
	Windows Vista Enterprise x64 <sup>1</sup>
	Windows Vista Business x64 <sup>1</sup>
	Windows Server 2008 Web
	Windows Server 2008 Standard Server
	Windows Server 2008 Standard Server without Hyper-V
	Windows Server 2008 Data Center
	Windows Server 2008 Data Center without Hyper-V
	Windows Server 2008 Enterprise
	Windows Server 2008 Enterprise without Hyper-V
	Windows Server 2008 x64 Standard
	Windows Server 2008 x64 Standard without Hyper-V <sup>1</sup>
	Windows Server 2008 x64 Data Center
	Windows Server 2008 x64 Data Center without Hyper-V $^1$
	Windows Server 2008 x64 Enterprise
	Windows Server 2008 x64 Enterprise without Hyper-V <sup>1</sup>
	KAM:

Memory	<ul> <li>Minimum: 512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>1</sup>This edition of SQL Server can be installed to the Windows on Windows (WOW64) 32-bit subsystem of a 64-bit server.

# <sup>□</sup> Specialized Editions (64-bit and 32-bit)

# SQL Server 2008 Developer (64-bit) IA64

The following table shows system requirements for SQL Server 2008 Developer (64-bit) IA64. Note that this edition of SQL Server 2008 is not available in the following localized versions: Italian, Spanish, Brazilian (Portuguese), or Russian.

Component	Requirement
Processor	<ul> <li>Processor type:</li> <li>Minimum: Itanium processor or faster</li> </ul>
	Processor speed:
	• Minimum: 1.0 GHz or faster
	Windows Server 2003 SP2 64-bit Itanium Data Center <sup>2</sup>
Operating System	Windows Server 2003 SP2 64-bit Itanium Enterprise <sup>2</sup>
	Windows Server 2008 64-bit Itanium Edition <sup>2</sup>
Memory	<ul> <li>RAM:</li> <li>Minimum: 512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>2</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Microsoft Windows that enables 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 operating system.

<sup>4</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core

installations.

# SQL Server 2008 Developer (64-bit) x64

The following table shows system requirements for SQL Server 2008 Developer (64-bit) x64:

Requirement
<ul> <li>Processor type:</li> <li>Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support</li> </ul>
<ul><li>Processor speed:</li><li>Minimum: 1.4 GHz</li><li>Recommended: 2.0 GHz or faster</li></ul>
Windows XP x64 Professional 2003
Windows Server 2003 SP2 64-bit x64 Standard <sup>2</sup>
Windows Server 2003 SP2 64-bit x64 Data Center <sup>2</sup>
Windows Server 2003 SP2 64-bit x64 Enterprise <sup>2</sup>
Windows Vista Ultimate x64
Windows Vista Home Premium x64
Windows Vista Home Basic x64
Windows Vista Enterprise x64
Windows Vista Business x64
Windows Server 2008 64-bit x64 Standard, Windows Server 2008 64-bit x64 Standard without Hyper-V <sup>2</sup>

	Windows Server 2008 64-bit x64 Data Center, Windows Server 2008 64-bit x64 Data Center without Hyper-V <sup>2</sup>
	Windows Server 2008 64-bit x64 Enterprise, Windows Server 2008 64-bit x64 Enterprise without Hyper-V <sup>2</sup>
	Windows Server 2008 x64 Web <sup>2</sup>
Memory	RAM: • Minimum:512 MB • Recommended: 2.048 GB or more • Maximum: Operating system maximum

<sup>2</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Microsoft Windows that enables 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 operating system.

<sup>4</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

## SQL Server 2008 Workgroup (64-bit) x64

The following table shows system requirements for SQL Server 2008 Workgroup (64-bit) x64:

Component	Requirement
Processor	<ul> <li>Processor type:</li> <li>Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support</li> <li>Processor speed:</li> <li>Minimum: 1.4 GHz</li> <li>Recommended: 2.0 GHz or faster</li> </ul>

	Windows XP x64 Professional 2003
	Windows Server 2003 SP2 64-bit x64 Standard <sup>2</sup>
	Windows Server 2003 SP2 64-bit x64 Data Center <sup>2</sup>
	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>2</sup>
	Windows Vista Ultimate x64
	Windows Vista Home Premium x64
Operating	Windows Vista Home Basic x64
System	Windows Vista Enterprise x64
	Windows Vista Business x64
	Windows Server 2008 x64 Web <sup>2</sup>
	Windows Server 2008 64-bit x64 Standard, Windows Server 2008 64-bit x64 Standard without Hyper-V <sup>2</sup>
	Windows Server 2008 64-bit x64 Data Center, Windows Server 2008 64-bit x64 Data Center without Hyper-V <sup>2</sup>
	Windows Server 2008 64-bit x64 Enterprise, Windows Server 2008 64-bit x64 Enterprise without Hyper-V <sup>2</sup>
	RAM:
Memory	<ul> <li>Minimum:512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>2</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Microsoft Windows that enables 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 operating system.

<sup>4</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

### SQL Server 2008 Web (64-bit) x64

The following table shows system requirements for SQL Server 2008 Web (64-bit) x64:

Component	Requirement
Processor	<ul> <li>Processor type:</li> <li>Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support</li> <li>Processor speed:</li> </ul>
	<ul><li>Minimum: 1.4 GHz</li><li>Recommended: 2.0 GHz or faster</li></ul>
	Windows XP x64 Professional 2003
	Windows Server 2003 SP2 64-bit x64 Standard <sup>2</sup>
	Windows Server 2003 SP2 64-bit x64 Data Center <sup>2</sup>
	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>2</sup>
	Windows Vista Ultimate x64
Operating	Windows Vista Enterprise x64
System	Windows Vista Business x64
	Windows Server 2008 x64 Web
	Windows Server 2008 x64 Standard, Windows Server 2008 x64 Standard without Hyper-V <sup>2</sup>

	Windows Server 2008 x64 Data Center, Windows Server 2008 x64 Data Center without Hyper-V <sup>2</sup>
	Windows Server 2008 x64 Enterprise, Windows Server 2008 x64 Enterprise without Hyper-V <sup>2</sup>
Memory	<ul> <li>RAM:</li> <li>Minimum:512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>2</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Microsoft Windows that enables 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 operating system.

<sup>4</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

# SQL Server 2008 Express Advanced (64-bit) x64

The following table shows system requirements for SQL Server 2008 Express Advanced (64-bit) x64:

Component	Requirement
Processor	<ul> <li>Processor type:</li> <li>Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support</li> <li>Processor speed:</li> <li>Minimum: 1.4 GHz</li> <li>Recommended: 2.0 GHz or faster</li> </ul>
	Windows Server 2003 x64

	Windows Server 2003 SP2 64-bit x64 Standard <sup>1</sup>
	Windows Server 2003 SP2 64-bit x64 Data Center <sup>1</sup>
	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>1</sup>
	Windows Vista Ultimate x64
	Windows Vista Home Premium x64
	Windows Vista Home Basic x64
Operating	Windows Vista Enterprise x64
System	Windows Vista Business x64
	Windows Server 2008 64-bit x64 Web <sup>1</sup>
	Windows Server 2008 64-bit x64 Standard
	Windows Server 2008 64-bit x64 Standard without Hyper-V <sup>1</sup>
	Windows Server 2008 64-bit x64 Data Center
	Windows Server 2008 64-bit x64 Data Center without Hyper-V <sup>1</sup>
	Windows Server 2008 64-bit x64 Enterprise
	Windows Server 2008 64-bit x64 Enterprise without Hyper-V $^1$
	RAM:
Memory	<ul> <li>Minimum:512 MB</li> <li>Recommended: 1 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>2</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Windows that enables 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 operating system.

# SQL Server Developer (32-bit)

The following table shows the system requirements for SQL Server 2008 Developer (32-bit):

Component	Requirement
	Processor type:
	• Pentium III-compatible processor or faster
Processor	Processor speed:
	• Minimum: 1.0 GHz
	Recommended: 2.0 GHz or faster
	Windows XP Home Edition SP2
	Windows XP Professional SP2
	Windows XP Tablet SP2
	Windows XP Professional x64 SP2 <sup>1</sup>
	Windows XP SP2 Media Center 2002
	Windows XP SP2 Media Center 2004
	Windows XP Media Center 2005
	Windows XP Professional Reduced Media
	Windows XP Home Edition Reduced Media
	Windows Server 2003 Small Business Server R2 Standard
	Windows Server 2003 Small Business Server R2 Premium

Windows Server 2003 SP2 Standard

Windows Server 2003 SP2 Enterprise

Windows Server 2003 SP2 Data Center

Windows Server 2003 SP2 Small Business Server Standard

Windows Server 2003 SP2 Small Business Server Premium

Windows Server 2003 SP2 64-bit x64 Standard<sup>2</sup>

Windows Server 2003 SP2 64-bit x64 Data Center<sup>2</sup>

Windows Server 2003 SP2 64-bit x64 Enterprise<sup>2</sup>

Windows Vista Ultimate

Windows Vista Home Premium

Operating System Windows Vista Home Basic

Windows Vista Starter Edition

Windows Vista Enterprise

Windows Vista Business

Windows Vista Ultimate 64-bit x64<sup>2</sup>

Windows Vista Home Premium 64-bit x64<sup>2</sup>

Windows Vista Home Basic 64-bit x64<sup>2</sup>

Windows Vista Enterprise 64-bit x64<sup>2</sup>

Windows Vista Business 64-bit x64<sup>2</sup>

Windows Server 2008 Web

	Windows Server 2008 Standard
	Windows Server 2008 Standard without Hyper-V
	Windows Server 2008 Enterprise
	Windows Server 2008 Enterprise without Hyper-V
	Windows Server 2008 Data Center
	Windows Server 2008 Data Center without Hyper-V
	Windows Server 2008 64-bit x64 Standard
	Windows Server 2008 64-bit x64 Standard without Hyper-V <sup>2</sup>
	Windows Server 2008 Data Center
	Windows Server 2008 Data Center without Hyper-V <sup>2</sup>
	Windows Server 2008 Enterprise
	Windows Server 2008 Enterprise without Hyper-V <sup>2</sup>
	RAM:
Memory	<ul> <li>Minimum: 512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>1</sup>This edition of SQL Server 2008 can be installed to the Windows on Windows (WOW64) 32-bit subsystem of a 64-bit server.

<sup>2</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

# SQL Server Workgroup (32-bit)

The following table shows the system requirements for SQL Server 2008

Workgroup (32-bit):

Component	Requirement
	Processor type:
	• Pentium III-compatible processor or faster
Processor	Processor speed:
	<ul><li>Minimum: 1.0 GHz</li><li>Recommended: 2.0 GHz or faster</li></ul>
	Windows XP Professional SP2
	Windows XP SP2 Tablet
	Windows XP Professional 64-bit x64 <sup>3</sup>
	Windows XP SP2 Media Center 2002
	Windows XP SP2 Media Center 2004
	Windows XP Media Center 2005
	Windows XP Professional Reduced Media
	Windows Server 2003 Small Business Server R2 Standard
	Windows Server 2003 Small Business Server R2 Premium
	Windows Server 2003 SP2 Standard
	Windows Server 2003 SP2 Enterprise
	Windows Server 2003 SP2 Data Center
	Windows Server 2003 SP2 Small Business Server Standard
	Windows Server 2003 SP2 Small Business Server Premium

	Windows Server 2003 64-bit x64 Standard <sup>3</sup>
	Windows Server 2003 64-bit x64 Data Center <sup>3</sup>
	Windows Server 2003 64-bit x64 Enterprise <sup>3</sup>
Operating	Windows Vista Vista Ultimate
System	Windows Vista Enterprise
	Windows Vista Business
	Windows Vista 64-bit x64 Ultimate <sup>3</sup>
	Windows Vista 64-bit x64 Enterprise <sup>3</sup>
	Windows Vista 64-bit x64 Business <sup>3</sup>
	Windows Server 2008 2008 Web
	Windows Server 2008 Standard
	Windows Server 2008 Standard without Hyper-V
	Windows Server 2008 Data Center
	Windows Server 2008 Data Center without Hyper-V
	Windows Server 2008 Enterprise
	Windows Server 2008 Enterprise without Hyper-V
	Windows Server 2008 64-bit x64 Standard
	Windows Server 2008 64-bit x64 Standard without Hyper-V <sup>3</sup>
	Windows Server 2008 64-bit x64 Data Center
	Windows Server 2008 64-bit x64 Data Center without Hyper-

	V <sup>3</sup> Windows Server 2008 64-bit x64 Enterprise
	Windows Server 2008 64-bit x64 Enterprise without Hyper-V $^3$
	RAM:
Memory	<ul> <li>Minimum: 512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>3</sup>This edition of SQL Server 2008 can be installed to the Windows on Windows (WOW64) 32-bit subsystem of a 64-bit server.

<sup>5</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

## SQL Server Web (32-bit)

The following table shows the system requirements for SQL Server 2008 Web (32-bit):

Component	Requirement
Processor	<ul> <li>Processor type:</li> <li>Pentium III-compatible processor or faster</li> <li>Processor speed:</li> <li>Minimum: 1.0 GHz</li> <li>Recommended: 2.0 GHz or faster</li> </ul>
	Windows XP Professional SP2 Windows XP SP2 Tablet Windows XP Professional XP x64 <sup>3</sup>

	Windows XP SP2 Media Center 2002
	Windows XP SP2 Media Center 2004
	Windows XP Media Center 2005
	Windows XP Professional Reduced Media
	Windows Server 2003 Small Business Server R2 Standard
	Windows Server 2003 Small Business Server R2 Premium
	Windows Server 2003 SP2 Standard
	Windows Server 2003 SP2 Enterprise
	Windows Server 2003 SP2 Data Center
	Windows Server 2003 SP2 Small Business Server Standard
	Windows Server 2003 SP2 Small Business Server Premium
	Windows Server 2003 SP2 64-bit x64 Standard <sup>3</sup>
	Windows Server 2003 SP2 64-bit x64 Data Center <sup>3</sup>
	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>3</sup>
	Windows Vista Ultimate
Operating System	Windows Vista Enterprise
	Windows Vista Business
	Windows Vista Ultimate x64 <sup>3</sup>
	Windows Vista Enterprise x64 <sup>3</sup>
	Windows Vista Business x64 <sup>3</sup>

	Windows Server 2008 Web
	Windows Server 2008 Standard Server
	Windows Server 2008 Standard Server without Hyper-V
	Windows Server 2008 Data Center
	Windows Server 2008 Data Center without Hyper-V
	Windows Server 2008 Enterprise
	Windows Server 2008 Enterprise without Hyper-V
	Windows Server 2008 x64 Standard
	Windows Server 2008 x64 Standard without Hyper-V <sup>3</sup>
	Windows Server 2008 x64 Data Center
	Windows Server 2008 x64 Data Center without Hyper-V <sup>3</sup>
	Windows Server 2008 x64 Enterprise
	Windows Server 2008 x64 Enterprise without Hyper-V <sup>3</sup>
	RAM:
Memory	<ul> <li>Minimum: 512 MB</li> <li>Recommended: 2.048 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>3</sup>This edition of SQL Server 2008 can be installed to the Windows on Windows (WOW64) 32-bit subsystem of a 64-bit server.

<sup>5</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

# SQL Server Express (32-bit) and Express with Advanced Services

# (32-bit)

The following table shows the system requirements for SQL Server 2008 Express (32-bit) and Express with Advanced Services (32-bit):

Component	Requirement
	Processor type:
	Pentium III-compatible processor or faster
Processor	Processor speed:
	• Minimum: 1.0 GHz
	Recommended: 2.0 GHz or faster
	Windows XP SP2 Home
	Windows XP SP2 Professional
	Windows XP SP2 Tablet
	Windows XP SP2 Media Center 2002
	Windows XP SP2 Media Center 2004
	Windows XP Media Center 2005
	Windows Server 2003 Reduced Media
	Windows XP Home Edition Reduced Media
	Windows Server 2003 Small Business Server R2 Standard
	Windows Server 2003 Small Business Server R2 Premium
	Windows Server 2003 SP2 Standard
	Windows Server 2003 2003 SP2 Enterprise
	Windows Server 2003 SP2 Data Center

	Windows Server 2003 SP2 Web Edition
Operating System	Windows Server 2003 SP2 Small Business Server Standard
	Windows Server 2003 SP2 Small Business Server Premium
	Windows Server 2003 SP2 64-bit x64 Standard <sup>3</sup>
	Windows Server 2003 SP2 64-bit x64 Data Center <sup>3</sup>
	Windows Server 2003 SP2 64-bit x64 Enterprise <sup>3</sup>
	Windows Vista Ultimate
	Windows Vista Home Premium
	Windows Vista Home Basic
	Windows Vista Enterprise
	Windows Vista Business
	Windows Vista Ultimate 64-bit x64 <sup>8</sup>
	Windows Vista Home Premium 64-bit x64 <sup>8</sup>
	Windows Vista Home Basic 64-bit x64 <sup>8</sup>
	Windows Vista Enterprise 64-bit x64 <sup>8</sup>
	Windows Vista Business 64-bit x64 <sup>8</sup>
	Windows Server 2008 Standard Server
	Windows Server 2008 Standard Server without Hyper-V
	Windows Server 2008 Enterprise
	Windows Server 2008 Enterprise without Hyper-V

	Windows Server 2008 Data Center		
	Windows Server 2008 Data Center without Hyper-V		
	Windows Server 2008 Web Edition		
	Windows Server 2008 64-bit x64 Web Edition <sup>3</sup>		
	Windows Server 2008 64-bit x64 Standard		
	Windows Server 2008 64-bit x64 Standard without Hyper-V <sup>3</sup>		
	Windows Server 2008 64-bit x64 Data Center		
Windows Server 2008 64-bit x64 Data Center without V <sup>3</sup>			
	Windows Server 2008 64-bit x64 Enterprise		
	Windows Server 2008 64-bit x64 Enterprise without Hyper-V <sup>3</sup>		
	Windows XP Embedded SP2 feature pack 2007 <sup>7</sup>		
	Windows Embedded for Point of Service SP2 <sup>7</sup>		
	RAM:		
Memory	<ul> <li>Minimum: 256 MB</li> <li>Recommended: 1.024 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>		

<sup>3</sup>This edition of SQL Server 2008 can be installed to the Windows on Windows (WOW64) 32-bit subsystem of a 64-bit server.

<sup>5</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

<sup>6</sup> SQL Server 2008 support on Windows Server 2003 R2 is the same as support

on Windows Server 2003 SP2.

<sup>7</sup>Supports SQL Server 2008 Express, not SQL Server 2008 Express with Advanced Services.

<sup>8</sup> SQL Server 2008 Express Advanced can be installed to the Windows on Windows (WOW64) 32-bit subsystem of a 64-bit server with this operating system.

### SQL Server 2008 Express x64 (64-bit)

The following table shows system requirements for SQL Server 2008 Express x64 (64-bit):

Component	Requirement
Processor	<ul> <li>Processor type:</li> <li>Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support</li> <li>Processor speed:</li> <li>Minimum: 1.4 GHz</li> <li>Recommended: 2.0 GHz or faster</li> </ul>
	Windows Server 2003 x64 Windows Server 2003 SP2 64-bit x64 Standard Windows Server 2003 SP2 64-bit x64 Data Center Windows Server 2003 SP2 64-bit x64 Enterprise Windows Vista Ultimate x64 Windows Vista Home Premium x64 Windows Vista Home Basic x64

Operating	Windows Vista Enterprise x64
System	Windows Vista Business x64
	Windows Server 2008 64-bit x64 Web
	Windows Server 2008 64-bit x64 Standard
	Windows Server 2008 64-bit x64 Standard without Hyper-V
	Windows Server 2008 64-bit x64 Data Center
	Windows Server 2008 64-bit x64 Data Center without Hyper-V
	Windows Server 2008 64-bit x64 Enterprise
	Windows Server 2008 64-bit x64 Enterprise without Hyper-V
	RAM:
Memory	<ul> <li>Minimum:256 MB</li> <li>Recommended: 1.024 GB or more</li> <li>Maximum: Operating system maximum</li> </ul>

<sup>2</sup>Management Tools are supported in WOW64, a feature of 64-bit editions of Microsoft Windows that enables 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 operating system.

<sup>4</sup> SQL Server 2008 is not supported on Windows Server 2008 Server Core installations.

## **Cross-Language Support**

- The English-language version of SQL Server 2008 is supported on all localized versions of supported operating systems.
- Localized versions of SQL Server 2008 are supported on localized operating systems that are the same language as the localized SQL Server 2008 version.

- Localized versions of SQL Server can be upgraded to localized versions of SQL Server 2008 of the same language.
- Localized versions of SQL Server cannot be upgraded to the Englishlanguage version of SQL Server 2008.
- Localized versions of SQL Server cannot be upgraded to localized SQL Server 2008 versions of a different localized language.
- Localized versions of SQL Server 2008 are also supported on Englishlanguage versions of supported operating systems by using the Windows Multilingual User Interface Pack (MUI) settings. However, you must verify certain operating system settings before you install a localized version of SQL Server 2008 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 SQL Server 2008 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 before you install SQL Server 2008.

# **Extended System Support**

SQL Server 2008 64-bit versions include support for extended systems, also known as Windows on Windows (WOW64). WOW64 is a feature of 64-bit editions of Windows that enables 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 operating system.

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

During installation of SQL Server 2008, Windows Installer creates temporary files on the system drive. Before you run Setup to install or upgrade SQL Server, verify that you have at least 2.0 GB of available disk space on the system drive for these files. This requirement applies even if you install SQL Server components to a non-default drive.

Actual hard disk space requirements depend on your system configuration and

the features that you decide to install. The following table provides disk space requirements for SQL Server 2008 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
Integration Services	120 MB
Client Components	850 MB
SQL Server Books Online and SQL Server Compact Books Online	240 MB

## **Installing SQL Server on a Domain Controller**

For security reasons, Microsoft recommends that you do not install SQL Server 2008 on a domain controller. SQL Server Setup will not block installation on a computer that is a domain controller, but the following limitations apply:

- On Windows Server 2003, SQL Server services can run under a domain account or a local system account.
- You cannot run SQL Server services on a domain controller under a local service account or a network service account.
- After SQL Server is installed on a computer, you cannot change the computer from a domain member to a domain controller. You must uninstall SQL Server before you change the host computer to a domain controller.
- After SQL Server is installed on a computer, you cannot change the computer from a domain controller to a domain member. You must uninstall SQL Server before you change the host computer to a domain member.
- SQL Server failover cluster instances are not supported where cluster nodes are domain controllers.
- SQL Server Setup cannot create security groups or provision SQL Server service accounts on a read-only domain controller. In this scenario, Setup will fail.

# **Supported Clients (64-Bit)**

SQL Server 2008 64-bit client components can be installed on Windows Server 2003 (64-bit).

# □ See Also

### Reference

<u>Check Parameters for the System Configuration Checker</u> <u>Editions and Components of SQL Server 2008</u>

## Concepts

<u>Planning a SQL Server Installation</u> <u>Security Considerations for a SQL Server Installation</u> <u>Quick-Start Installation of SQL Server 2008</u>

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Check Parameters for the System Configuration Checker

# 

# Ξ 🖌

#### <u>Planning a SQL Server Installation</u> >

During SQL Server Setup, the System Configuration Checker (SCC) scans the computer where SQL Server will be installed. The SCC checks for conditions that prevent a successful SQL Server installation. Before Setup starts the SQL Server Installation Wizard, the SCC retrieves the status of each item. It then compares the result with required conditions and provides guidance for removal of blocking issues.

# <sup>□</sup> Additional Rule topics

See the following topics for scenario-specific Setup rules:

- <u>Feature Installation Rules</u>
- Upgrade and Repair Rules Check
  Edition Upgrade Rules
  Uninstallation rules

# $\Box$ See Also

### Reference

Hardware and Software Requirements for Installing SQL Server 2008 Version and Edition Upgrades

## Concepts

Security Considerations for a SQL Server Installation

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Security Considerations for a SQL Server Installation

- -

<u>Planning a SQL Server Installation</u> >

Security is important 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 consider 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
- <u>Use firewalls</u>
- <u>Isolate services</u>
- <u>Configure a secure file system</u>
- <u>Disable NetBIOS and server message block</u>

# **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 accessible only to authorized 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 do not connect your SQL Servers directly to the Internet.
- Back up all data regularly and secure the backups in an off-site location.

# **Use Firewalls**

Firewalls are important to help secure the SQL Server installation. Firewalls will be most effective if you follow these guidelines:

- Put a firewall between the server and the Internet. Enable your firewall. If your firewall is turned off, turn it on. If your firewall is turned on, do not turn it off.
- Divide the network into security zones separated by firewalls. Block all traffic, and then selectively admit only what is required.
- In a multi-tier environment, use multiple firewalls to create screened subnets.
- When you are installing the server inside a Windows domain, configure

interior firewalls to allow Windows Authentication.

• If your application uses distributed transactions, you might have to configure the firewall to allow Microsoft Distributed Transaction Coordinator (MS DTC) traffic to flow between separate MS DTC instances. You will also have to configure the firewall to allow traffic to flow between the MS DTC and resource managers such as SQL Server.

For more information about the default Windows firewall settings, and a description of the TCP ports that affect the Database Engine, Analysis Services, Reporting Services, and Integration Services, see <u>Configuring the Windows</u> <u>Firewall to Allow SQL Server Access</u>.

# **Isolate Services**

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

• Run separate SQL Server services under separate Windows accounts. Whenever possible, use separate, low-rights Windows or Local user accounts for each SQL Server service. For more information, see <u>Setting</u> <u>Up Windows Service Accounts</u>.

# **Configure a Secure File System**

Using the correct file system increases security. For SQL Server installations, you should do the following tasks:

• Use the NTFS file system (NTFS). NTFS is the preferred file system for installations of SQL Server because it is more stable and recoverable than FAT file systems. NTFS also enables security options like file and directory access control lists (ACLs) and Encrypting File System (EFS) file encryption. During installation, SQL Server will set appropriate ACLs on registry keys and files if it detects NTFS. These permissions should not be changed. Future releases of SQL Server might not support installation on computers with FAT file systems.

#### <sup>II</sup>Note:

If you use EFS, database files will be encrypted under the identity of the account running SQL Server. Only this account will be able to decrypt the
files. If you must change the account that runs SQL Server, you should first decrypt the files under the old account and then re-encrypt them under the new account.

• Use a redundant array of independent disks (RAID) for critical data files.

#### **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 reduce the threat of user enumeration.

# □ 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 by using the lowest possible permissions.
- Associate SQL Server services with low privileged Windows local user accounts, or domain user accounts.
- For more information, see <u>Setting Up Windows Service Accounts</u>.

#### 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 for password strength and expiration.
- Always use strong passwords for all SQL Server logins. For more information, see the <u>SQL Server 2008 Security Overview for Database</u> <u>Administrators</u> white paper.

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Setting Up Windows Service Accounts

# 

#### -

#### <u>Planning a SQL Server Installation</u> >

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

#### Security Note:

Always run SQL Server services by using the lowest possible user rights. Use a specific user account or domain account instead of a shared account for SQL Server services. Use separate accounts for different SQL Server services. Do not grant additional permissions to the SQL Server service account. Permissions will be granted through group membership or granted directly to a service SID.

Depending on the components that you decide to install, SQL Server 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 enables automation of some administrative tasks.

### <sup>ℤ</sup>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. For more information about how to customize account information for each service, see <u>How to: Install</u> <u>SQL Server 2008 (Setup)</u>.

- Analysis Services Provides online analytical processing (OLAP) and data mining functionality for business intelligence applications.
- Reporting Services Manages, executes, creates, schedules, and delivers reports.
- Integration Services Provides management support for Integration Services package storage and execution.
- SQL Server Browser The name resolution service that provides SQL Server connection information for client computers.
- Full-text search Quickly creates full-text indexes on content and properties of structured and semistructured data to provide document filtering and word-breaking for SQL Server.
- 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:

- <u>Configuring Service StartupType</u>
- <u>Using Startup Accounts for SQL Server Services</u>
- <u>Using Service SIDs for SQL Server Services</u>
- Identifying Instance-Aware and Instance-Unaware Services
- Reviewing NT Rights and Privileges Granted for SQL Server Service <u>Accounts</u>
- <u>Reviewing Access Control Lists Created for SQL Server Service</u>
   <u>Accounts</u>
- <u>Reviewing Windows Permissions for SQL Server Services</u>
- <u>Reviewing Additional Considerations</u>
- Localized Service Names

# <sup>□</sup> Configuring Service Startup Type

During SQL Server Setup, you can configure the startup type - diabled, manual, or automatic - for some SQL Server services. The following table shows the SQL Server services that can be configured during installation. For unattended installations, you can use the switches in a configuration file or at a command prompt.

SQL Server service name	Configurable in Installation Wizard?	Switches for unattended installations <sup>1</sup>
		SQLACCOUNT,
MSSQLSERVER	Yes	SQLPASSWORD,
		SQLAUTOSTART
	Yes	AGTACCOUNT,
SQLServerAgent <sup>2</sup>		AGTPASSWORD,
		AGTAUTOSTART
		ASACCOUNT,
MSSQLServerOLAPService	Yes	ASPASSWORD,
		ASAUTOSTART
ReportServer	Yes	RSACCOUNT,
		RSPASSWORD,
		RSAUTOSTART
		ISSVSAccount,
Integration Services	Yes	ISSVCPassword,
		ISSVCStartupType

<sup>1</sup>For more information and sample syntax for unattended installations, see <u>How</u> to: Install SQL Server 2008 from the Command Prompt.

<sup>2</sup>The SQL Server Agent service is disabled on instances of SQL Server Express and SQL Server Express with Advanced Services.

## <sup>©</sup> Using Startup Accounts for SQL Server Services

To start and run, each service in SQL Server must have a user account that is configured during installation. User accounts can be built-in system accounts, local user 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 automatically started by the operating system.

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

SQL Server service name	Optional accounts	Startup type	Default state following Setup
SQL Server	SQL Server Express: Domain User, Local System, Network Service <sup>1</sup> All other editions: Domain User, Local System, Network Service <sup>1</sup>	Automatic <sup>2</sup>	Started Stopped only if user chooses not to autostart
SQL Server Agent	Domain User, Local System, Network Service <sup>1</sup>	Manual <sup>3</sup> Automatic only if user chooses to autostart	Stopped Started only if user chooses to autostart
Analysis Services	Domain User, Network Service, Local Service,	Automatic	Started Stopped only if user

	Local System		chooses not to autostart
Reporting Services	Domain User, Local System, Network Service, Local Service	Automatic	Started Stopped only if user chooses not to autostart
Integration Services	Domain User, Local System, Network Service, Local Service	Automatic	Started Stopped only if user chooses not to autostart.
Full-Text Search	Use an account different than the account for the SQL Server service. The account will default to Local Service on Windows Server 2008 and Windows Vista.	Automatic	Started Stopped only if an account is not specified on Windows Server 2003 or Windows XP.
SQL Server Browser	Local Service	Disabled <sup>4</sup> Automatic only if user chooses to autostart.	Stopped Started only if user chooses to autostart.
SQL Server Active Directory Helper	Local System, Network Service	Disabled	Stopped
SQL Writer	Local System	Automatic	Started

<sup>1</sup> **Important** We recommend that you do not use the Network Service account for SQL Server services. Network Service is a shareable account, and is appropriate for use as a SQL Server service account only if you can ensure that no other services that use the account are installed on the computer. Domain User accounts that are not a Windows administrator are more appropriate for SQL Server services. Be aware that the Local Service account is not supported

for SQL Server Agent.

<sup>2</sup>Set as manual in failover cluster configurations.

<sup>3</sup>The SQL Server Agent service is disabled on instances of SQL Server Express and SQL Server Express with Advanced Services.

<sup>4</sup>By default, for failover cluster installations and named instances, the SQL Server Browser is set to start automatically after Setup finishes.

### Using a Domain User Account

If the service must interact with network services, access domain resources like file shares, or if it uses linked server connections to other computers running SQL Server, you might use a minimally privileged domain server account. Many server-to-server activities can be performed only with a domain user account.

### **Using the Local Service Account**

The Local Service account is a built-in account that has the same level of access to resources and objects as members of the 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. Be aware that the Local Service account is not supported for the SQL Server or SQL Server Agent services.

### Using a Local User Account

If the computer is not part of a domain, a local user account without Windows administrator permissions is recommended.

### **Using the Network Service Account**

The Network Service account is a built-in account that has more access to resources and objects than members of the Users group. Services that run as the Network Service account access network resources by using the credentials of the computer account.

#### <sup>Important:</sup>

We recommend that you do not use the Network Service account for SQL Server services. Network Service is a shareable account, and is appropriate for use as a SQL Server service account only if you can ensure that no other services that use the account are installed on the computer.

### **Using the Local System Account**

Use caution when you assign Local System permissions to SQL Server service accounts. Local System is a very high-privileged account.

### **Using Service SIDs for SQL Server Service Configurations**

Service security IDs (SIDs) are available in SQL Server 2008 on Windows Server 2008 and Windows Vista operating systems to enable service isolation. Service isolation provides services a way to access specific objects without having to either run in a high-privilege account or weaken the security protection of the object. A SQL Server service can use this identity to restrict access to its resources by other services or applications.

A service isolates an object for its exclusive use by securing the resource by using an access control entry that contains a service SID. This per-service SID is derived from the service name and is unique to that service, for example, a service SID name for a SQL Server service might be NT Service\MSSQL\$<InstanceName>. After a SID has been enabled for a service, the SID is added to the local security group in Windows. The service owner can modify the access control list for an object to give access to the SID. For example, a registry key in **HKEY\_LOCAL\_MACHINE\SOFTWARE** would usually be available only to services that have administrative credentials. By adding the per-service SID to the access control list of the key, the service can run in a lower-privilege account, but still have access to the key.

During installation,SQL Server Setup creates a service group for each component of SQL Server. On Windows Server 2003 and Windows XP, service accounts for SQL Server services are added to the local service groups and function on these operating systems as they have in previous SQL Server releases; SQL Server services connect to an instance of SQL Server as members of the local group. On Windows Server 2008 and Windows Vista, the service SID is added to the local security group instead of the SQL Server service account.

#### **Supported Service Configurations**

Installing SQL Server 2008 on a Windows Server 2008 or Windows Vista domain controller requires SQL Server services to be provisioned with a service SID. SQL Server 2008 is not supported where failover cluster nodes are domain controllers The following table shows the service configurations for new and upgraded installations on Windows Server 2008 or Windows Vista.

New installation	Upgrade
<ul> <li>Stand-alone instance - Service group with service SID</li> </ul>	<ul> <li>Stand-alone instance - Service group with service SID</li> </ul>
• Failover cluster instance - Service SID	• Failover cluster instance - Service group with service account

Installing SQL Server 2008 on a Windows Server 2003 or Windows XP domain controller requires the domain for SQL Server services to be provisioned with a domain account. SQL Server 2008 is not supported where failover cluster nodes are domain controllers. The following table shows the service configurations for new and upgraded installations on Windows Server 2003 or Windows XP.

New installation	Upgrade
<ul> <li>Stand-alone instance- Service group with service account</li> <li>Failover cluster instance - Domain service group with service account</li> </ul>	<ul> <li>Stand-alone instance - Service group with service account</li> <li>Failover cluster instance - Service group with service account</li> </ul>

### **Changing User Accounts**

To change the service accounts, password, service startup type, or other properties of any SQL Server–related service, use SQL Server Configuration

Manager. Note that renaming the instance of SQL Server does not rename SQL Server security groups. The security groups will continue to function with the old names after the renaming operation.

## Identifying Instance-Aware and Instance-Unaware Services

Instance-aware services are associated with a specific instance of SQL Server, and have their own registry hives. You can install multiple copies of instanceaware services by running SQL Server Setup for 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 SQL Server include the following:

- SQL Server
- SQL Server Agent Be aware that the SQL Server Agent service is disabled on instances of SQL Server Express and SQL Server Express with Advanced Services.
- Analysis Services
- Reporting Services
- Full-text search

Instance-unaware services in SQL Server include the following:

- 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 local service groups for different SQL Server services and adds the service accounts or service SIDs to them as appropriate. These groups simplify granting the permissions that are required to run SQL Server services and other executables, and help secure SQL Server files. For standalone instances of SQL Server on Windows Vista and Windows Server 2008 operating systems, service SIDs are automatically added to the group. For failover cluster instances on Windows Server 2008, Setup does not create a default user group. Instead, Setup adds SQL Server resources to the access control lists by service SID. Users can also use domain groups and domain accounts to run SQL Server services. Note that in order to utilized domain groups, they must be created before you run SQL Server Setup.

Unique service group names are required when you install SQL Server on a domain controller, and only domain accounts can be used for these groups.

The following table shows the user groups that SQL Server Setup creates are granted specific Windows NT user rights.

SQL Server service	User group
SQL	Default instance: SQLServerMSSQLUser\$ <i>ComputerName</i> \$MSSQLSERVER

Server	Named instance: SQLServerMSSQLUser\$ <i>ComputerName</i> \$ <i>Instance</i> N
SQL Server Agent <sup>3</sup>	Default instance: SQLServerSQLAgentUser\$ <i>ComputerName</i> \$MSSQLSERVER Named instance: SQLServerSQLAgentUser\$ <i>ComputerName</i> \$InstanceName
Analysis Services	Default instance: SQLServerMSOLAPUser\$ <i>ComputerName</i> \$MSSQLSERVER Named instance: SQLServerMSOLAPUser\$ <i>ComputerName</i> \$ <i>InstanceName</i>
SSRS	Default instance: SQLServerReportServerUser\$ <i>ComputerName</i> \$MSRS10.MSSQLSEF Named instance: SQLServerReportServerUser\$ <i>ComputerName</i> \$MSRS10. <i>InstanceNan</i>

Integration Services	Default or named instance: SQLServerDTSUser\$ <i>ComputerName</i>
Full-text search	Default instance: SQLServerFDHostUser\$ <i>ComputerName</i> \$MSSQL10.MSSQLSERVER Named instance: SQLServerFDHostUser\$ <i>ComputerName</i> \$MSSQL10. <i>InstanceName</i>
SQL Server Browser	Default or named instance: SQLServerSQLBrowserUser\$ <i>ComputerN</i>
SQL Server Active Directory Helper	Default or named instance: SQLServerMSSQLServerADHelperUser\$ <i>ComputerName</i>
SQL Writer	N/A

<sup>1</sup>This permission is granted by default to all SQL Server services.

<sup>2</sup> SQL Server Setup does not check or grant permissions for this service.

<sup>3</sup>The SQL Server Agent service is disabled on instances of SQL Server Express and SQL Server Express with Advanced Services.

## Reviewing Access Control Lists Created for SQL Server Service Accounts

SQL Server service accounts must have access to resources. Access control lists (ACLs) are set at the user group level.

### <sup>I</sup>Important:

For failover cluster installations, resources on shared disks must be set to an ACL for a local account.

Service account <sup>1</sup> for	Files and folders
MSSQLServer	Instid\MSSQL\backup
	Instid\MSSQL\binn
	Instid\MSSQL\data
	Instid\MSSQL\FTData
	Instid\MSSQL\Install
	Instid\MSSQL\Log
	Instid\MSSQL\Repldata
	100\shared
	Instid\MSSQL\Template Data (SQL Server Express
SQLServerAgent <sup>2</sup>	Instid\MSSQL\binn
	Instid\MSSQL\binn

The following table shows the ACLs that are set by SQL Server Setup:

	Instid\MSSQL\Log
	100\com
	100\shared
	100\shared\Errordumps
FTS	Instid\MSSQL\FTData
	Instid\MSSQL\FTRef
	100\shared
	100\shared\Errordumps
	Instid\MSSQL\Install
	Instid\MSSQL\jobs
MSSQLServerOLAPservice	100\shared\ASConfig
	Instid\OLAP
	Instid\Olap\Data
	Instid\Olap\Log
	Instid\OLAP\Backup
	Instid\OLAP\Temp
	100\shared\Errordumps

SQLServerReportServerUser	Instid\Reporting Services\Log Files
	Instid\Reporting Services\ReportServer
	Instid\Reportingservices\Reportserver\global.asax
	Instid\Reportingservices\Reportserver\Reportserver
	Instid\Reporting Services\reportManager
	Instid\Reporting Services\RSTempfiles
	100\shared
	100\shared\Errordumps
MSDTSServer100	100\dts\binn\MsDtsSrvr.ini.xml
	100\dts\binn
	100\shared
	100\shared\Errordumps
SQL Server Browser	100\shared\ASConfig
	100\shared
	100\shared\Errordumps
MSADHelper	N/A (Runs under Network Service account)
SQLWriter	N/A (Runs as local system)
User	Instid\MSSQL\binn
	Instid\Reporting Services\ReportServer

Instid\Reportingservices\Reportserver\global.asax
Instid\Reporting Services\ReportManager
Instid\Reporting Services\ReportManager\pages
Instid\Reporting Services\ReportManager\Styles
100\dts
100\tools
90\tools
80\tools
100\sdk
Microsoft SQL Server\100\Setup Bootstrap

<sup>1</sup>In some scenarios, such as failover clustering and installing SQL Server on a domain controller, ACLs will be set for the SQL Server service SID instead of being set for the SQL Server service.

<sup>2</sup>The SQL Server Agent service is disabled on instances of SQL Server Express and SQL Server Express with Advanced Services.

Some access control permissions might have to be granted to built-in accounts or other SQL Server service accounts. The following table lists additional ACLs that are set by SQL Server Setup.

Requesting component	Account	Resource	
MSSQLServer	Performance Log Users	Instid\MSSQL\binn	Ι
	Performance Monitor Users	Instid\MSSQL\binn	Ι
	Performance Log Users,	\WINNT\system32\sqlctr100.dll	F

	Performance Monitor Users	
	Administrator	\\.\root\Microsoft\SqlServer\ServerEvents\
	only	<sql_instance_name>1</sql_instance_name>
	Administrators, System	\tools\binn\schemas\sqlserver\2003\03\showplanF
	Users	\tools\binn\schemas\sqlserver\2003\03\showplanF
Reporting Services	<report server<br="">Web Service Account&gt;</report>	<pre><install>\Reporting Services\LogFiles F F F F F F F F F F F F F F F F F F F</install></pre>
	Report Manager Application pool identity, ASP.NET account,	<install>\Reporting Services\ReportManager, <install>\Reporting Services\ReportManager\Pages\*.*, <install>\Reporting Services\ReportManager\Styles\*.*, <install>\Reporting</install></install></install></install>
	Everyone Report Manager	Services\ReportManager\webctrl_client\1_0\*.* <install>\Reporting F</install>

Applicati pool iden	on Services\ReportManager\Pages\*.*
<pre><report account="" s="" serv="" web=""></report></pre>	Server ice < <i>install</i> >\Reporting Services\ReportServer
<report s<br="">Web Serv Account&gt;</report>	Gerver ice Services\ReportServer\global.asax
Everyone	< <i>install</i> >\Reporting Services\ReportServer\global.asax
Network	<install>\Reporting</install>
Everyone	<pre></pre>

ReportServer Windows Services Account	<pre><install>\Reporting Services\ReportServer\RSReportServer.config F F F F F F F F F F F F F F F F F F F</install></pre>
Everyone	Report Server keys (Instid hive) F
Terminal Services User	Report Server keys (Instid hive)
Power Users	Report Server keys (Instid hive)

	ľ
	Γ
	F

<sup>1</sup>This is the WMI provider namespace.

## Reviewing Windows Permissions for SQL Server Services

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

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

SOL Server	Default instance:	Executes jobs, monitors SQL Serv
Agent	SQLServerAgent	
(InstanceName) <sup>1</sup>	Named instance:	The path of the executable file is
	SQLAgent\$ <i>InstanceName</i>	\MSSQL\Binn\sqlagent.exe.

Analysis Services	Default instance: MSSQLServerOLAPService	The service that provides online ar (OLAP) and data mining functiona intelligence applications.
Services (InstanceName)	Named instance: MSOLAP\$ <i>InstanceName</i>	The path of the executable file is \OLAP\Bin\msmdsrv.exe.
Reporting	Default instance: ReportServer	Manages, executes, creates, schedı reports.
Services	Named instance: ReportServer\$ <i>InstanceName</i>	The path of the executable is \Report Services\ReportServer\Bin\Report
Integration Services	Default or named instance: MSDTSServer	Provides management support for package storage and execution. The path of the executable file is \DTS\Binn\msdtssrvr.exe.
Full-text search	Default or named instance: SQL Full-text Filter Daemon Launcher	Service to launch the full-text filte perform document filtering and wc Server full-text search.
SQL Server Browser	Default or named instance: SQLBrowser	The name resolution service that p connection information for client c is shared across multiple SQL Serv The path of the executable file is < Files\Microsoft SQL Server\100\S]
SQL Server Active Directory Helper	Default or named instance: MSSQLServerADHelper.	Publishes and manages SQL Serve Active Directory. The path of the executable is <i><driv< i=""> Files\Microsoft SQL Server\100\S</driv<></i>
		Allows backup and restore applica

SQL Writer	SQLWriter	Volume Shadow Copy Service fran single instance of the SQL Writer s of SQL Server on the server.
		The path of the executable file is < Files\Microsoft SQL Server\100\S

<sup>1</sup>The SQL Server Agent service is disabled on instances of SQL Server Express and SQL Server Express with Advanced Services.

# Reviewing Additional Considerations

The following table shows the permissions that are required for SQL Server services to provide additional functionality.

Service/Application	Functionality	Required permission
SQL Server (MSSQLSERVER)	Write to a mail slot using <b>xp_sendmail</b> .	Network write permissions.
SQL Server (MSSQLSERVER)	Run <b>xp_cmdshell</b> 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 <b>Administrators</b> local group.
Database Engine Tuning Advisor	Tunes databases for optimal query performance.	On first use, a user who has system administrative credentials must initialize the application. After initialization, <b>dbo</b> 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 Books Online.

#### <sup>I</sup>Important:

Before you upgrade SQL Server, enable Windows Authentication for SQL Server Agent and verify the required default configuration: that the SQL Server Agent service account is a member of the SQL Server **sysadmin** group.

# <sup>□</sup> Localized Service Names

The following table shows service names that are displayed by localized versions of Windows.

Language	Name for Local Service	Name for Network Service	Nam
English Simplified Chinese Traditional Chinese Korean Japanese	NT AUTHORITY\LOCAL SERVICE	NT AUTHORITY\NETWORK SERVICE	NT AUT
German	NT- AUTORITÄT\LOKALER DIENST	NT- AUTORITÄT\NETZWERKDIENST	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

### □ See Also

#### Reference

Database Engine Configuration - Account Provisioning Analysis Services Configuration - Account Provisioning

### Concepts

<u>Security Considerations for a SQL Server Installation</u> <u>File Locations for Default and Named Instances of SQL Server</u>

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Network Protocols and Network Libraries



- -

<u>Planning a SQL Server Installation</u> >

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 is configured for TCP/IP port 1433, and named pipe \\.\pipe\sql\query. SQL Server 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 is installed. To enable, disable, and configure network protocols after Setup is complete, use the SQL Server Network Configuration area of the SQL Server Configuration Manager.

# <sup>□</sup> Types of Endpoints

SQL Server 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.

Transact-SQL endpoints created by SQL Server Setup are as follows:

- Transact-SQL local machine
- Transact-SQL named pipes
- Transact-SQL default TCP
- Transact-SQL default VIA

For more information about Transact-SQL endpoints, see "Network Protocols and TDS Endpoints" in SQL Server Books Online.

For more information about SQL Server network configurations, see the following topics in SQL Server Books Online:

- "Server Network Configuration"
- "Configuring Server Network Protocols and Net-Libraries"
- "Default SQL Server Network Configuration"
- "Net-Libraries and Network Protocols"

### $\Box$ See Also

### **Concepts** <u>Planning a SQL Server Installation</u>

### **Other Resources Understanding Surface Area Configuration**

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help File Locations for Default and Named Instances of SQL Server

See Also

#### - ▼

#### <u>Planning a SQL Server Installation</u> >

An installation of SQL Server 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 of SQL Server 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 instance of SQL Server.

### <sup>I</sup> Important:

Program files and data files cannot be installed on a removable disk drive, cannot be installed on a file system that uses compression, cannot be installed to a directory where system files are located, and cannot be installed on shared drives on a failover cluster instance.

Do not delete any of the following directories or their contents: Binn, Data, Ftdata, HTML, or 1033. You can delete other directories, if necessary; however, you might not be able to retrieve any lost functionality or data without uninstalling and then reinstalling SQL Server.Do not delete or modify any of the .htm files in the HTML directory. They are required for SQL Server tools to function properly.

## <sup>□</sup> Shared Files for All Instances of SQL Server

Common files used by all instances on a single computer are installed in the folder *<drive>*:\Program Files\Microsoft SQL Server\100\, where *<drive>* is the drive letter where components are installed. Normally this is drive C.
# <sup>□</sup> 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 default instance ID is constructed by using the following format:

- MSSQL for the Database Engine, followed by the major version number and a period, followed by the instance name.
- MSAS for Analysis Services, followed by the major version number and a period, followed by the instance name.
- MSRS for Reporting Services, followed by the major version number and a period, followed by the instance name.

Examples of default instance IDs in this release of SQL Server are as follows:

- MSSQL10.MSSQLSERVER for a default instance of SQL Server.
- MSAS10.MSSQLSERVER for a default instance of Analysis Services.
- MSSQL10.MyInstance for a named instance of SQL Server named "MyInstance."

The directory structure for a SQL Server 2008 named instance that includes the Database Engine and Analysis Services, named "MyInstance", and installed to the default directories would be as follows:

- C:\Program Files\Microsoft SQL Server\MSSQL10.MyInstance\
- C:\Program Files\Microsoft SQL Server\ASSQL10.MyInstance\

You can specify any value for the instance ID, but avoid special characters and reserved keywords. For more information, see <u>Reserved Keywords (Setup</u>).

You can specify a non-default instance ID during SQL Server Setup. Instead of <Program Files>\Microsoft SQL Server, a <custom path>\Microsoft SQL Server is used if the user chooses to change the default installation directory. Note that instance IDs that begin with an underscore (\_) or that contain the number sign (#) or the dollar sign (\$) are not supported.

## <sup>I</sup>Note:

Integration Services and client components are not instance aware and, therefore are not assigned an instance ID. By default, non-instance-aware components are installed to a single directory: *<drive>*:\Program Files\Microsoft SQL Server\100\. 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.

SQL Server Analysis Services is the only SQL Server component that supports instance renaming after installation. If an instance of Analysis Services is renamed, the instance ID will not change. After instance renaming is complete, directories and registry keys will continue to use the instance ID created during installation.

The registry hive is created under HKLM\Software\Microsoft\Microsoft SQL Server\<*Instance\_ID*> for instance-aware components. For example,

- HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10.MyInstance
- HKLM\Software\Microsoft\Microsoft SQL Server\ASSQL10.MyInstance
- 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"="MSSQL10"
- [HKEY\_LOCAL\_MACHINE\Software\Microsoft\Microsoft SQL Server\Instance Names\OLAP] "InstanceName"="ASSQL10"
- [HKEY\_LOCAL\_MACHINE\Software\Microsoft\Microsoft SQL Server\Instance Names\RS] "InstanceName"="RSSQL10"

# <sup>□</sup> 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 userconfigurable destination folder:

Component	Default path <sup>1, 2</sup>	Configurable <sup>3</sup> or Fixed Path
Database Engine server components	\Program Files\Microsoft SQL Server\MSSQL10. <instanceid>\</instanceid>	Configurable
Database Engine data files	\Program Files\Microsoft SQL Server\MSSQL10. <instanceid>\</instanceid>	Configurable
Analysis Services server	\Program Files\Microsoft SQL Server\MSAS10. <instanceid>\</instanceid>	Configurable
Analysis Services data files	\Program Files\Microsoft SQL Server\MSAS10. <instanceid>\</instanceid>	Configurable
Reporting Services report server	\Program Files\Microsoft SQL Server\MSRS10. <instanceid>\Reporting Services\ReportServer\Bin\</instanceid>	Configurable
Reporting Services report manager	\Program Files\Microsoft SQL Server\MSRS10. <instanceid>\Reporting Services\ReportManager\</instanceid>	Fixed path
Integration Services	<install directory="">\100\DTS\</install>	Configurable <sup>4</sup>
Client Components	<install directory="">\100\Tools\</install>	Configurable <sup>4</sup>
Replication and server- side COM objects	< <i>drive</i> >:\Program Files\Microsoft SQL Server\100\COM\ <sup>5</sup>	Fixed path
Integration Services component DLLs for the Data Transformation Run- time engine, the Data Transformation Pipeline engine, and the <b>dtexec</b>	< <i>drive</i> >:\Program Files\Microsoft SQL Server\100\DTS\Binn	Fixed path

command prompt utility		
DLLs that provide managed connection support for Integration Services	< <i>drive</i> >:\Program Files\Microsoft SQL Server\100\DTS\Connections	Fixed path
DLLs for each type of enumerator that Integration Services supports	< <i>drive</i> >:\Program Files\Microsoft SQL Server\100\DTS\ForEachEnumerators	Fixed path
SQL Server Browser Service, WMI providers	< <i>drive</i> >:\Program Files\Microsoft SQL Server\100\Shared\	Fixed path
Components that are shared between all instances of SQL Server	< <i>drive</i> >:\Program Files\Microsoft SQL Server\100\Shared\	Fixed path

 $^1Ensure that the \Program Files Microsoft SQL Server folder is protected with limited permissions.$ 

<sup>2</sup>The default drive for these locations is *systemdrive*, normally drive C.

<sup>3</sup>Installation paths for child features are determined by the installation path of the parent feature.

<sup>4</sup>A single installation path is shared between Integration 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.

<sup>5</sup>This directory is used by all instances of SQL Server on a computer. If you apply an update to any of the instances on the computer, any changes to files in this folder will affect all instances on the computer. When you add 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.

## <sup>I</sup>Note:

For clustered configurations, you must select a local drive that is available on

every node of the cluster.

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 ID for tools and other shared files. Setup also does not use any instance ID for the Analysis Services program and data files, although it does use the instance ID 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\MSSQL10\MSSQL\", instancespecific directories are added to the end of that path.

Customers who choose 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>\MSSQL10\MSSQL\MSSQL10\MSSQL\Data\*. Instead, to use the USESYSDB feature, set an installation path for the SQL Data Files feature instead of the Database Engine feature.

## <sup>I</sup>Note:

Data files are always expected to be found in a child directory named Data. For example, specify C:\Program Files\Microsoft SQL Server\MSSQL10\ 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\MSSQL10\MSSQL\Data.

## $\Box$ See Also

## Reference

Database Engine Configuration - Data Directories Analysis Services Configuration - Data Directories

## Concepts

Planning a SQL Server Installation

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Working with Multiple Versions and Instances of SQL Server



- -

<u>Planning a SQL Server Installation</u> >

SQL Server supports multiple instances of the Database Engine, Analysis Services, and Reporting Services on the same computer. You can also upgrade earlier versions of SQL Server, or install SQL Server on a computer where earlier SQL Server versions are already installed. For supported upgrade scenarios, see <u>Version and Edition Upgrades</u>.

# Using SQL Server Side-By-Side with Previous Versions of SQL Server

You can install SQL Server on a computer that is already running instances of an earlier SQL Server version. If a default instance already exists on the computer, SQL Server must be installed as a named instance.

The following table shows side-by-side support for SQL Server 2008:

Existing instance of SQL Server 2008	Side-by-side support
	SQL Server 2000 (32-bit)
COL Convert 2000 (22 bit)	SQL Server 2000 (64-bit) x64
SQL Server 2006 (52-Dit)	SQL Server 2005 (32-bit)
	SQL Server 2005 (64-bit) x64
SQL Server 2008 (64-bit) IA64	SQL Server 2005 (64-bit) IA64
	SQL Server 2000 (32-bit)
SOI Sorver 2008 (64 bit) $x64$	SQL Server 2000 (64-bit) x64
SQL Server 2000 (04-Dit) x04	SQL Server 2005 (32-bit)
	SQL Server 2005 (64-bit) x64

## □ See Also

## Tasks

How to: Install SQL Server 2008 (Setup)

## Reference

Hardware and Software Requirements for Installing SQL Server 2008 Version and Edition Upgrades

## Concepts

Upgrading to SQL Server 2008 Backward Compatibility Using Upgrade Advisor to Prepare for Upgrades

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Collation and Unicode Support



#### - -

<u>Planning a SQL Server Installation</u> >

As databases expand to support a growing global market, users must be able to work with character data in meaningful ways. Collations let users sort and compare strings according to their own conventions. Collations are a critical part of creating a database and manipulating data.

SQL Server 2008 is fully aligned with the collations in Windows Server 2008.

# <sup>□</sup> What's New in SQL Server 2008 Collations

The following collation changes are new in SQL Server 2008:

SQL Server 2008 has introduced new collations that are in full alignment with collations that Windows Server 2008 provides. These 80 new collations are denoted by \*\_100 version references. They provide users with the most up-to-date and linguistically accurate cultural sorting conventions. Support includes the following:

- New East Asian government standards.
- Liguistically correct surrogates.
- Chinese minority scripts.
- Unicode 5.0 case table.
- Weighting has been added to previously non-weighted characters that would have compared equally.

New versions are being added to the existing Windows collations from SQL Server 2000 and SQL Server 2005 to reflect these changes. All current collations will be maintained in SQL Server 2005 for baclward compatibility. No changes have been made to the SQL\_\* collations.

Deprecation of the Korean\_Wansung\_Unicode, Hindi\_CI\_AS, Macedonian\_CI\_AS, Lithuanian\_Classic\_CI\_AS, Cyrillic\_90\_CI\_AS, and Azeri\_Latin\_90\_CI\_AS Windows collations, and deprecation of the SQL\_ALTDiction\_CP1253\_CS\_AS SQL collation. These collations were supported SQL Server 2005, and are supported in SQL Server 2008. However, they will not be displayed in the Setup collations list, and will not appear when you use the system function **::fn\_helpcollations()** to query the list of supported collations in SQL Server 2008.

## $\Box$ See Also

### Reference

Collation Settings in Setup Selecting a SQL Server Collation

## Concepts

Collation and International Terminology

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Collation and International Terminology



### - -

<u>Planning a SQL Server Installation</u> > <u>Collation and Unicode Support</u> >

Collations in SQL Server provide sorting rules, case, and accent sensitivity properties for your data. Collations that are used with character data types such as char and varchar dictate the code page and corresponding characters that can be represented for that data type. Whether you are installing a new instance of SQL Server, restoring a database backup, or connecting server to client databases, it is important that you understand the locale requirements, sorting order, and case and accent sensitivity of the data you will be working with.

When you select a collation for your server, database, column, or expression, you are assigning certain characteristics to your data that will affect the results of many operations in your database. For example, when you construct a query by using ORDER BY, the sort order of your result set might be dependent on the collation that is applied to the database or dictated in a COLLATE clause at the expression level of the query.

A collation can contain any or all of the following characteristics:

- Case sensitivity
- Accent sensitivity
- Kana sensitivity
- Width sensitivity

To best use collation support in SQL Server, you must understand the terms that are defined in this topic, and how they relate to the characteristics of your data.

# <sup>□</sup> Terms

- <u>Collation</u>
- <u>Locale</u>
- <u>Unicode</u>
- <u>Code page</u>
- Data types
- <u>Sort order</u>

## 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 supports storing objects with different collations in a single database. For non-Unicode columns, the collation setting specifies the code page for the data and which characters can be represented. Data that is moved between non-Unicode columns must be converted from the source code page to the destination code page.

Transact-SQL statement results can vary when the statement is run in the context of different databases that have different collation settings. If it is possible, use a standardized collation for your organization. This way, you do not have 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, 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 sensitivity, case sensitivity, accent sensitivity, Kana-sensitive, and width sensitivity.

SQL Server collations include the following collation sets:

### Windows collations

Windows collations define rules for storing character data that are based on an associated Windows system locale. For a Windows collation, comparison of non-Unicode data is implemented by using the same algorithm as Unicode data. The base Windows collation rules specify which alphabet or language is used when dictionary sorting is applied, and the code page that is 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 lets developers sort strings in their applications by using the same rules that are used by SQL Server. For more information, see <u>Collation Settings in Setup</u>.

#### **Binary collations**

Binary collations sort data based on the sequence of coded values that are defined by the locale and data type. They are case sensitive. A binary collation in SQL Server defines the locale and the ANSI code page to be used. This enforces a binary sort order. Because they are relatively simple, binary collations help improve application performance. For non-Unicode data types, data comparisons are based on the code points that are defined in the ANSI code page. For Unicode data types, data comparisons are based on the code points that are defined 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 they are used on Unicode data.

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 set of pure codepoint comparison collations. You can migrate to the binary collations to take advantage of true code-point comparisons. Youo should use binary collations to develop new applications. The BIN2 suffix identifies collation names that implement the new code-point collation semantics. There is also a comparison flag that corresponds to BIN2 for the new binary sort. For more information, see <u>Guidelines for Using BIN and BIN2 Collations</u>.

#### SQL Server

SQL Server collations (SQL\_\*) provide sort order compatibility with earlier

versions of SQL Server. The dictionary sorting rules for non-Unicode data are not compatible with any sorting routine that is provided by Windows operating systems. However, sorting 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 will see different results for comparisons of the same data, depending on the underlying data type. For more information, see <u>Using SQL Server</u> <u>Collations</u>.

## <sup> </</sup> Note:

When you upgrade an English-language instance of SQL Server, SQL Server collations (SQL\_\*) 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 the following are true:

- Your application code depends on the behavior of previous SQL Server collations.
- You are going to use SQL Server replication with existing installations of SQL Server 6.5 or SQL Server 7.0.
- You must store character data that reflects multiple languages.

Setting collations are supported at the following levels of an instance of SQL Server:

### Server-level collations

The default collation is set during SQL Server setup, and also becomes the default collation of the system databases. Note that Unicode-only collations cannot be selected during SQL Server setup because they are not supported as server-level collations.

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 recreating the object. Instead of changing the default collation of an instance of SQL Server, you can specify the collation at the time you create a new database or database column.

To query the server collation for an instance of SQL Server, 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, you can use the COLLATE clause of the CREATE DATABASE statement to specify the default database collation. If no collation is specified, the database is assigned the default collation of the **model** database. The collation of the **model** database is the same as the default collation of the instance of SQL Server.

The collation of a user database can be changed by using an ALTER DATABASE statement similar to the following:

Copy Code

ALTER DATABASE myDB COLLATE Greek\_CS\_AI

The current collation of a database can be retrieved by using a statement similar to the following:

SELECT CONVERT (varchar, DATABASEPROPERTYEX('database\_name','co]

Altering the database-level collation does not affect column-level or expression-level collations.

Column-level collations

When you create a table, you can specify collations for each characterstring column by using the COLLATE clause of the CREATE TABLE statement. If no collation is specified, the column is assigned the default collation of the database. The collation of a column can be changed by using an ALTER TABLE statement similar to the following:

ALTER TABLE myTable ALTER COLUMN mycol NVARCHAR(10) COLLATE Gree

Expression-level collations

Expression-level collations are set when a statement is run, and they affect the way a result set is returned. This enables ORDER BY sort results to be locale-specific. Use a COLLATE clause such as the following to implement expression-level collations:

SELECT name FROM customer ORDER BY name COLLATE Latin1\_General\_(

#### Back to Top

#### Locale

A locale is a set of information that is associated with a place or a culture. This can include the name and identifier of the spoken language, the script that is used to write the language, and cultural conventions. Collations can be associated with one or more locales.

#### Back to Top

### Unicode

If you store character data that reflects multiple languages, always use Unicode data types (**nchar**, **nvarchar**, and **ntext**) instead of the non-Unicode data types (**char**, **varchar**, and **text**).

Significant limitations are associated with non-Unicode data types. This is because a non-Unicode computer will be limited to use of a single code page. You might experience performance gain by using Unicode because fewer codepage conversions are required. Unicode collations must be selected individually at the database, column or expression level because they are not supported at the server level. The code pages that a client uses are determined by your operating system settings. To set client code pages on the Windows 2000, Windows XP, Windows Server 2003, or Windows Server 2008 operating systems, use **Regional Settings** in Control Panel.

When you move data from a server to a client, your server collation might not be recognized by older client drivers. This can happen when you move data from a Unicode server to a non-Unicode client. Your best option might be to upgrade the client operating system so that the underlying system collations are updated. If your client has database client software installed, you might consider applying a service update to the database client software.

You can also try to use a different collation for the data on the server. Choose a collation that will map to a code page on the client. For more information, see the "Setting and Changing Collations" topic in SQL Server Books Online.

To evaluate issues that are related to using Unicode or non-Unicode data types, test your scenario to measure performance differences in your environment. It is a good practice to standardize the collation that is used on systems across your organization, and deploy Unicode servers and clients wherever possible. For more information about Unicode, see the <u>Unicode Consortium Web site</u>.

In many situations, SQL Server will interact with other servers or clients, and your organization might use multiple data access standards between applications and server instances. SQL Server clients are one of two main types:

- **Unicode clients** that use OLE DB and Open Database Connectivity (ODBC) version 3.7 or a later version.
- **Non-Unicode clients** that use DB-Library and ODBC version 3.6 or an earlier version.

The following table provides information about using multilingual data with various combinations of Unicode and non-Unicode servers.

Server	Client	Benefits or Limitations
		Because Unicode data will be used throughout the system, this
UnicodeUnicode		scenario provides the best performance and protection from
	corruption of retrieved data. This is the situation with ActiveX	
		Data Objects (ADO), OLE DB, and ODBC version 3.7 or a

		later version.
Unicode	Non- Unicode	In this scenario, especially with connections between a server that is running a newer operating system and a client that is running an older version of SQL Server, or on an older operating system, there can be limitations or errors when you move data to a client computer. Unicode data on the server will try to map to a corresponding code page on the non- Unicode client to convert the data.
Non- Unicode	Unicode	This is not an ideal configuration for using multilingual data. You cannot 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-	Non-	This is a very limiting scenario for multilingual data. You can
Unicode	Unicode	use only a single code page.

If you experience errors or difficulties when moving data from a server to a client, specifically from a Unicode server to a non-Unicode client, your server collation might not be recognized by older client drivers. In this situation, your best option might be to upgrade the client operating system so that the underlying system collations are updated. If your client has database client software installed, you might consider applying a service update to the database client software.

You can also try to use a different collation for the data on the server. Choose a collation that will map to a code page on the client. For more information about changing collations, see the "Setting and Changing Collations in SQL Server" topic in SQL Server Books Online. For more information about changing collations, see the "SQL Server Best Practices Collation Change" white paper. For more information about migrating non-Unicode datatypes to Unicode, see the "SQL Server Best Practices Migration to Unicode" whitepape.

### **Related Topics: Unicode Basics**

### Back to Top

## **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 Windows code page is commonly referred to as a *character set* or *charset*. Code pages are used to provide support for the character sets and keyboard layouts that are used by different Windows system locales.

All Windows Server 2008 Unicode collations are Unicode 5.0-based.

Back to Top

## **Data Types**

A data type is a definition that specifies a range of values, the operations that can be performed on the values, and how the values are stored in memory. Defining data types lets SQL Server predictably manipulate data. Non-Unicode character data types are **char**, **varchar**, and **text**. Unicode data types are **nchar**, **nvarchar**, and **ntext**. We recommend that you use Unicode data types in your applications, especially if you store character data that reflects multiple languages.

For more information about migrating non-Unicode datatypes to Unicode, see the "SQL Server Best Practices Migration to Unicode" <u>whitepaper</u>.

### Related Topics: Data Types (Database Engine), Data Types (Transact-SQL), Integration Services Data Types

Back to Top

## Sort Order

Sort order specifies how data values are sorted. This affects the results of data comparison. Data is sorted by using collations, and it can be optimized by using indexes.

Related Topics: <u>Windows Collation Sorting Styles</u>, Indexes

Back to Top

## <sup>□</sup> See Also

## Reference

Collation and Unicode Support

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Collation Settings in Setup



#### - -

<u>Planning a SQL Server Installation</u> > <u>Collation and Unicode Support</u> >

Many server-to-server activities can fail or yield inconsistent results if collation settings are not consistent across servers. To avoid this, select a Windows locale or SQL Server collation (SQL\_\*) to match the collation settings in other instances of SQL Server.

SQL Server 2008 supports setting collations at the following levels:

- Server
- Database
- Column
- Expression

For more information, see <u>Collation and International Terminology</u> and <u>How to:</u> <u>Install SQL Server 2008 from the Command Prompt</u>.

# <sup>□</sup> 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. For more information, see <u>Windows Collation</u> <u>Sorting Styles</u> and <u>Guidelines for Using BIN and BIN2 Collations</u>.

# □ SQL Server Collations (SQL\_\*)

Select this option to match settings with English-language versions SQL Server 2005 or earlier versions. For more information, see <u>Using SQL Server</u> <u>Collations</u>.

# <sup>□</sup> Windows system locale

The default installation settings are determined by the Windows system locale. The server-level collation can either be changed during setup, or by changing the Windows system locale before installation,. The default collations are set to the oldest available version that is associated with each specific locale. This is due to backwards compatibility reasons. Therefore, these are not always the recommended collations. To take advantage of the SQL Server 2008 features, change the default installation settings for Windows collations..

To set the Windows locale for operating systems before Windows Vista, in Contol Panel, find the Windows system locale name on the **Advanced** tab in **Regional and Language Options**. In Windows Vista, use the **Formats** tab.

## **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).

## Sort Order

Select sort order options to use with the selected collation designator. Binary is the fastest sorting order and is case-sensitive, but can yield unexpected sort orders. If Binary is selected, the Case-sensitive, Accent-sensitive, Kanasensitive, and Width-sensitive options are not available. For more information, see <u>Windows Collation Sorting Styles</u>.

# <sup>□</sup> Default Collations in SQL Server Setup

In Control Panel, find the Windows system locale name on the **Advanced** tab in **Regional and Language Options**. In Windows Vista, use the **Formats**" tab. The following table shows the corresponding collation designator and code page to match collation settings with an existing Windows locale:

Windows Locale	Windows LCID	Default	Cod Page
Afrikaans (South Africa)	0x0436	Latin1_General_CI_AS	1252
Albanian (Albania)	0x041c	Albanian_CI_AS	1252
Alsatian (France)	0x0484	Latin1_General_100_CI_AS	1252
Amharic (Ethiopia)	0x045e	Latin1_General_100_CI_AS	1252
Arabic (Algeria)	0x1401	Arabic_CI_AS	125
Arabic (Bahrain)	0x3c01	Arabic_CI_AS	1250
Arabic (Egypt)	0x0c01	Arabic_CI_AS	1250
Arabic (Iraq)	0x0801	Arabic_CI_AS	1250
Arabic (Jordan)	0x2c01	Arabic_CI_AS	1250
Arabic (Kuwait)	0x3401	Arabic_CI_AS	125
Arabic (Lebanon)	0x3001	Arabic_CI_AS	1250
Arabic (Libya)	0x1001	Arabic_CI_AS	1250
Arabic (Morocco)	0x1801	Arabic_CI_AS	125(
Arabic (Oman)	0x2001	Arabic_CI_AS	1250
Arabic (Qatar)	0x4001	Arabic_CI_AS	1250
Arabic (Saudi Arabia)	0x0401	Arabic_CI_AS	1250
Arabic (Syria)	0x2801	Arabic_CI_AS	1250
Arabic (Tunisia)	0x1c01	Arabic_CI_AS	125
Arabic (U.A.E.)	0x3801	Arabic_CI_AS	125
Arabic (Yemen)	0x2401	Arabic_CI_AS	125

Armenian (Armenia)	0x042b	Cyrillic_General_100_CI_AS	125:
Assamese (India)	0x044d	Not available at server level	Uni
Azeri (Azerbaijan, Cyrillic)	0x082c	Azeri_Cyrillic_90_CI_AS	125:
Azeri (Azerbaijan, Latin)	0x042c	Azeri_Latin_90_CI_AS	1254
Bashkir (Russia)	0x046d	Bashkir_100_CI_AS	125
Basque (Basque)	0x042d	Latin1_General_100_CI_AS	1252
Belarusian (Belarus)	0x0423	Cyrillic_General_CI_AS	<b>125</b> :
Bengali (Bangladesh)	0x0845	Not available at server level	Unic
Bengali (India)	0x0445	Not available at server level	Uni
Bosnian (Bosnia and Herzegovina, Cyrillic)	0x201a	Bosnian_Cyrillic_100_CI_AS	125:
Bosnian (Bosnia and Herzegovina, Latin)	0x141a	Bosnian_Latin_100_CI_AS	125(
Breton (France)	0x047e	Breton_100_CI_AS	1252
Bulgarian (Bulgaria)	0x0402	Cyrillic_General_CI_AS	<b>125</b> :
Catalan (Catalan)	0x0403	Latin1_General_CI_AS	1252
Chinese (Hong Kong SAR, PRC)	0x0c04	Chinese_Hong_Kong_Stroke_90_CI_AS	950
Chinese (Macao SAR)	0x1404	Chinese_Traditional_Pinyin_100_CI_AS	950
Chinese (Macau)	0x21404	Chinese_Traditional_Stroke_Order_100_CI_AS	950
Chinese (PRC)	0x0804	Chinese_PRC_CI_AS	936
Chinese (PRC)	0x20804	Chinese_PRC_Stroke_CI_AS	936
Chinese (Singapore)	0x1004	Chinese_PRC_CI_AS	936
Chinese			

(Singapore)	0x21004	Chinese_Simplified_Stroke_Order_100_CI_AS	936
Chinese (Taiwan)	0x30404	Chinese_Taiwan_Bopomofo_CI_AS	950
Chinese (Taiwan)	0x0404	Chinese_Taiwan_Stroke_CI_AS	950
Corsican (France)	0x0483	Corsican_100_CI_AS	1252
Croatian (Bosnia and Herzegovina, Latin)	0x101a	Croatian_100_CI_AS	125(
Croatian (Croatia)	0x041a	Croatian_CI_AS	125(
Czech (Czech Republic)	0x0405	Czech_CI_AS	125(
Danish (Denmark)	0x0406	Danish_Norwegian_CI_AS	1252
Dari (Afghanistan)	0x048c	Dari_100_CI_AS	125(
Divehi (Maldives)	0x0465	Not available at server level	Unic
Dutch (Belgium)	0x0813	Latin1_General_CI_AS	1252
Dutch (Netherlands)	0x0413	Latin1_General_CI_AS	1252
English (Australia)	0x0c09	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 (India)	0x4009	Latin1_General_100_CI_AS	1252
English (Ireland)	0x1809	Latin1_General_CI_AS	1252
English (Jamaica)	0x2009	Latin1_General_CI_AS	1252
English (Malaysia)	0x4409	Latin1_General_100_CI_AS	1252
English (New Zealand)	0x1409	Latin1_General_CI_AS	1252
English	0x3409	Latin1_General_CI_AS	1252

(Philippines)			
English (Singapore)	0x4809	Latin1_General_100_CI_AS	1252
English (South Africa)	0x1c09	Latin1_General_CI_AS	1252
English (Trinidad and Tobago)	0x2c09	Latin1_General_CI_AS	1252
English (United Kingdom)	0x0809	Latin1_General_CI_AS	1252
English (United States)	0x0409	SQL_Latin1_General_CP1_CI_AS	1252
English (Zimbabwe)	0x3009	Latin1_General_CI_AS	1252
Estonian (Estonia)	0x0425	Estonian_CI_AS	125
Faroese (Faroe Islands)	0x0438	Latin1_General_CI_AS	1252
Filipino (Philippines)	0x0464	Latin1_General_100_CI_AS	1252
Finnish (Finland)	0x040b	Finnish_Swedish_CI_AS	1252
French (Belgium)	0x080c	French_CI_AS	1252
French (Canada)	0x0c0c	French_CI_AS	1252
French (France)	0x040c	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
Frisian (Netherlands)	0x0462	Frisian_100_CI_AS	1252
Galician (Spain)	0x0456	Latin1_General_CI_AS	1252
Georgian (Georgia)	0x0437	Cyrillic_General_100_CI_AS	125:
Georgian (Georgia)	0x10437	Georgian_Modern_Sort_CI_AS	
German - Phone			

Book Sort (DIN)	0x10407	German_PhoneBook_CI_AS	1252
German (Austria)	0x0c07	Latin1_General_CI_AS	1252
German (Germany)	0x0407	Latin1_General_CI_AS	1252
German (Liechtenstein)	0x1407	Latin1_General_CI_AS	1252
German (Luxembourg)	0x1007	Latin1_General_CI_AS	1252
German (Switzerland)	0x0807	Latin1_General_CI_AS	1252
Greek (Greece)	0x0408	Greek_CI_AS	1253
Greenlandic (Greenland)	0x046f	Danish_Greenlandic_100_CI_AS	1252
Gujarati (India)	0x0447	Not available at server level	Unic
Hausa (Nigeria, Latin)	0x0468	Latin1_General_100_CI_AS	1252
Hebrew (Israel)	0x040d	Hebrew_CI_AS	125
Hindi (India)	0x0439	Not available at server level	Unic
Hungarian (Hungary)	0x040e	Hungarian_CI_AS	125(
Hungarian Technical Sort	0x1040e	Hungarian_Technical_CI_AS	125(
Icelandic (Iceland)	0x040f	Icelandic_CI_AS	1252
Igbo (Nigeria)	0x0470	Latin1_General_100_CI_AS	1252
Indonesian (Indonesia)	0x0421	Latin1_General_CI_AS	1252
Inuktitut (Canada, Latin)	0x085d	Latin1_General_100_CI_AS	1252
Inuktitut (Syllabics) Canada	0x045d	Latin1_General_100_CI_AS	1252
Irish (Ireland)	0x083c	Latin1_General_100_CI_AS	1252
Italian (Italy)	0x0410	Latin1_General_CI_AS	1252
Italian (Switzerland)	0x0810	Latin1_General_CI_AS	1252

Japanese (Japan XJIS)	0x0411	Japanese_CI_AS	932
Japanese (Japan)	0x040411	Japanese_Bushu_Kakusu_100_CI_AS	932
Kannada (India)	0x044b	Not available at server level	Unic
Kazakh (Kazakhstan)	0x043f	Kazakh_90_CI_AS	125:
Khmer (Cambodia)	0x0453	Not available at server level	Unio
K'iche (Guatemala)	0x0486	Modern_Spanish_100_CI_AS	1252
Kinyarwanda (Rwanda)	0x0487	Latin1_General_100_CI_AS	1252
Konkani (India)	0x0457	Not available at server level	Unic
Korean (Korea Dictionary Sort)	0x0412	Korean_Wansung_CI_AS	949
Kyrgyz (Kyrgyzstan)	0x0440	Cyrillic_General_CI_AS	125:
Lao (Lao PDR)	0x0454	Not available at server level	Unic
Latvian (Latvia)	0x0426	Latvian_CI_AS	125
Lithuanian (Lithuania)	0x0427	Lithuanian_CI_AS	125′
Lower Sorbian (Germany)	0x082e	Latin1_General_100_CI_AS	1252
Luxembourgish (Luxembourg)	0x046e	Latin1_General_100_CI_AS	1252
Macedonian (Macedonia, FYROM)	0x042f	Macedonian_FYROM_90_CI_AS	125:
Malay (Brunei Darussalam)	0x083e	Latin1_General_CI_AS	1252
Malay (Malaysia)	0x043e	Latin1_General_CI_AS	1252
Malayalam (India)	0x044c	Not available at server level	Unic
Maltese (Malta)	0x043a	Maltese_100_CI_AS	1252
Maori (New Zealand)	0x0481	Maori_100_CI_AS	1252

Mapudungun (Chile)	0x047a	Mapudungan_100_CI_AS	1252
Marathi (India)	0x044e	Not available at server level	Unic
Mohawk (Canada)	0x047c	Mohawk_100_CI_AS	1252
Mongolian (Mongolia)	0x0450	Cyrillic_General_CI_AS	125:
Mongolian (PRC)	0x0850	Cyrillic_General_100_CI_AS	125:
Nepali (Nepal)	0x0461	Not available at server level	Unic
Norwegian (Bokmål, Norway)	0x0414	Norwegian_100_CI_AS	1252
Norwegian (Nynorsk, Norway)	0x0814	Norwegian_100_CI_AS	1252
Occitan (France)	0x0482	French_100_CI_AS	1252
Oriya (India)	0x0448	Not available at server level	Unic
Pashto (Afghanistan)	0x0463	Not available at server level	Unic
Persian (Iran)	0x0429	Persian_100_CI_AS	1250
Polish (Poland)	0x0415	Polish_CI_AS	125(
Portuguese (Brazil)	0x0416	Latin1_General_CI_AS	1252
Portuguese (Portugal)	0x0816	Latin1_General_CI_AS	1252
Punjabi (India)	0x0446	Not available at server level	Unic
Quechua (Bolivia)	0x046b	Latin1_General_100_CI_AS	1252
Quechua (Ecuador)	0x086b	Latin1_General_100_CI_AS	1252
Quechua (Peru)	0x0c6b	Latin1_General_100_CI_AS	1252
Romanian (Romania)	0x0418	Romanian_CI_AS	1250
Romansh (Switzerland)	0x0417	Romansh_100_CI_AS	1252
Russian (Russia)	0x0419	Cyrillic_General_CI_AS	125

Sami (Inari, Finland)	0x243b	Sami_Sweden_Finland_100_CI_AS	1252
Sami (Lule, Norway)	0x103b	Sami_Norway_100_CI_AS	1252
Sami (Lule, Sweden)	0x143b	Sami_Sweden_Finland_100_CI_AS	1252
Sami (Northern, Finland)	0x0c3b	Sami_Sweden_Finland_100_CI_AS	1252
Sami (Northern, Norway)	0x043b	Sami_Norway_100_CI_AS	1252
Sami (Northern, Sweden)	0x083b	Sami_Sweden_Finland_100_CI_AS	1252
Sami (Skolt, Finland)	0x203b	Sami_Sweden_Finland_100_CI_AS	1252
Sami (Southern, Norway)	0x183b	Sami_Norway_100_CI_AS	1252
Sami (Southern, Sweden)	0x1c3b	Sami_Sweden_Finland_100_CI_AS	1252
Sanskrit (India)	0x044f	Not available at server level	Unic
Serbian (Bosnia and Herzegovina, Cyrillic)	0x1c1a	Cyrillic_General_CI_AS	125:
Serbian (Bosnia and Herzegovina, Latin)	0x181a	Latin1_General_CI_AS	125(
Serbian (Serbia, Cyrillic)	0x0c1a	Cyrillic_General_CI_AS	125:
Serbian (Serbia, Latin)	0x081a	Latin1_General_CI_AS	125(
Sesotho sa Leboa/Northern Sotho (South Africa)	0x046c	Latin1_General_100_CI_AS	1252
Setswana/Tswana (South Africa)	0x0432	Latin1_General_100_CI_AS	1252
Sinhala (Sri Lanka)	0x045b	Not available at server level	Unic

Slovak (Slovakia)	0x041b	Slovak_CI_AS	1250
Slovenian (Slovenia)	0x0424	Slovenian_CI_AS	125(
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 (Mexico)	0x080a	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 (Spain)	0x0c0a	Modern_Spanish_CI_AS	1252
Spanish (Spain, Traditional Sort)	0x040a	Traditional_Spanish_CI_AS	1252

Spanish (United States)	0x540a	Latin1_General_100_CI_AS	1252
Spanish (Uruguay)	0x380a	Modern_Spanish_CI_AS	1252
Spanish (Venezuela)	0x200a	Modern_Spanish_CI_AS	1252
Swahili (Kenya)	0x0441	Latin1_General_100_CI_AS	1252
Swedish (Finland)	0x081d	Finnish_Swedish_CI_AS	1252
Swedish (Sweden)	0x041d	Finnish_Swedish_CI_AS	1252
Syriac (Syria)	0x045a	Not available at server level	Uni
Tajik (Tajikistan)	0x0428	Cyrillic_General_100_CI_AS	1251
Tamazight (Algeria, Latin)	0x085f	Tamazight_100_CI_AS	1252
Tamil (India)	0x0449	Not available at server level	Uni
Tatar (Russia)	0x0444	Cyrillic_General_CI_AS	125
Telugu (India)	0x044a	Not available at server level	Unic
Thai (Thailand)	0x041e	Thai_CI_AS	874
Tibetan (PRC)	0x0451	Not available at server level	Unic
Turkish (Turkey)	0x041f	Turkish_CI_AS	1254
Turkmen (Turkmenistan)	0x0442	Turkmen_100_CI_AS	1250
Uighur (PRC)	0x0480	Uighur_100_CI_AS	125(
Ukrainian (Ukraine)	0x0422	Ukrainian_100_CI_AS	125:
Upper Sorbian (Germany)	0x042e	Upper_Sorbian_100_CI_AS	1252
Urdu (Pakistan)	0x0420	Urdu_100_CI_AS	1250
Uzbek (Uzbekistan, Cyrillic)	0x0843	Cyrillic_General_CI_AS	125:
Uzbek (Uzbekistan, Latin)	0x0443	Uzbek_Latin_90_CI_AS	1254
Vietnamese (Vietnam)	0x042a	Vietnamese_CI_AS	1258
Welsh (United Kingdom)	0x0452	Welsh_100_CI_AS	1252
----------------------------------	--------	--------------------------	------
Wolof (Senegal)	0x0488	French_100_CI_AS	1252
Xhosa/isiXhosa (South Africa)	0x0434	Latin1_General_100_CI_AS	1252
Yakut (Russia)	0x0485	Yakut_100_CI_AS	125
Yi (PRC)	0x0478	Latin1_General_100_CI_AS	1252
Yoruba (Nigeria)	0x046a	Latin1_General_100_CI_AS	1252
Zulu/isiZulu (South Africa)	0x0435	Latin1_General_100_CI_AS	1252

## □ See Also

#### Reference

Collation and Unicode Support Windows Collation Sorting Styles

#### Concepts

Using SQL Server Collations

SQL Server Setup Help Windows Collation Sorting Styles



#### - -

<u>Planning a SQL Server Installation</u> > <u>Collation and Unicode Support</u> >

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.

## <sup>ℤ</sup>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 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 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-pointto-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 set of pure codepoint comparison collations. Customers can choose to migrate to the binary collations to take advantage of true code-point comparisons, and they should utilize the binary collations for development of new applications. The BIN2 suffix identifies collation names that implement the code-point collation semantics. There is also a comparison flag that corresponds to BIN2 for the new binary sort. For more information, see <u>Guidelines for Using BIN and BIN2</u> <u>Collations</u>.

The following table describes Windows collation sort order options for SQL Server.

Sort order (suffix)	Sort order description
Binary (_BIN) <sup>1</sup>	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 <u>Guidelines for Using BIN and BIN2 Collations</u> . If this option is not selected, SQL Server follows sorting and comparison rules as defined in dictionaries for the associated language or alphabet.
Binary- code point (_BIN2) <sup>1</sup>	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. 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 <u>Guidelines for Using BIN and BIN2</u> <u>Collations</u> .

Case- sensitive (_CS)	Distinguishes between uppercase and lowercase letters. If selected, lowercase letters sort ahead of their uppercase versions. If this option is not selected, SQL Server considers the uppercase and lowercase versions of letters to be identical for sorting purposes.
Accent- sensitive (_AS)	Distinguishes between accented and unaccented characters. For example, 'a' is not equal to '?'. If this option is not selected, SQL Server considers the accented and unaccented versions of letters to be identical for sorting purposes.
Kana- sensitive (_KS)	Distinguishes between the two types of Japanese kana characters: Hiragana and Katakana. If this option is not selected, SQL Server considers Hiragana and Katakana characters to be equal for sorting purposes.
Width- sensitive (_WS)	Distinguishes between a single-byte character and the same character when represented as a double-byte character. 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.

<sup>1</sup> If selected, the case-sensitive, accent-sensitive, kana-sensitive, and width-sensitive options are not available.

# <sup>□</sup> 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 <sup>1</sup>	Binary sort.	
_BIN2 <sup>1</sup>	Binary-code point sort order.	
_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-
_C5_A5_K5_W5	sensitive
CS AS MS	Case-sensitive, accent-sensitive, kana-insensitive, width-
_C3_A3_W3	sensitive

<sup>1</sup> If BIN or Binary-code point is selected, the Case-sensitive, Accent-sensitive, Kana-sensitive, and Width-sensitive options are not available.

## $\Box$ See Also

#### Reference

Collation Settings in Setup

#### Concepts

**Guidelines for Using BIN and BIN2 Collations** 

#### SQL Server Setup Help Guidelines for Using BIN and BIN2 Collations



#### - -

<u>Planning a SQL Server Installation</u> > <u>Collation and Unicode Support</u> >

The following considerations will help you to decide whether BIN or BIN2 binary collations are appropriate for your SQL Server implementation.

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 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, those ending with "\_BIN", 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.

# <sup>□</sup> Guidelines for Using BIN Collations

If your SQL Server applications interact with older versions of SQL Server that use binary collations, continue to use binary. Binary collations might be a more suitable choice for mixed environments.

## <sup>□</sup> Guidelines for Using BIN2 Collations

Binary collations in this release of SQL Server include a set of pure code-point comparison collations. You can migrate to the binary collations to take advantage of true code-point comparisons, and you should use the new binary collations for development of new applications. The BIN2 suffix identifies collation names that implement the new code-point collation semantics. There is also a comparison flag that corresponds to BIN2 for the binary sort. Advantages include simpler application development and clearer semantics.

## $\Box$ See Also

#### Reference

Collation and Unicode Support Collation Settings in Setup

SQL Server Setup Help Using SQL Server Collations



#### - -

<u>Planning a SQL Server Installation</u> > <u>Collation and Unicode Support</u> >

SQL Server 2008 collation settings depend on the type of installation. Typically, you should choose a SQL Server collation that supports the Windows system locale most commonly used at your organization. To identify your Windows system locale on computers that are running Windows 2000, click **Regional Settings** in Control Panel, and then click **Advanced**. In Windows Vista, use the **Formats** tab. To identify your Windows system locale on computers that are running Windows Server 2003, click **Regional and Language Options** in Control Panel, and then click the **Regional Options** tab. In most cases, a computer runs the Windows system locale that matches the language requirements of the user. Therefore, SQL Server Setup automatically detects the Windows system locale and selects the appropriate SQL Server collation. For backward compatibility, the default English-language (US) collation is SQL\_Latin1\_General\*.

SQL Server 2008 collations control the following:

- The code page that is used to store non-Unicode data in SQL Server.
- The rules that govern how SQL Server sorts and compares characters that are stored in non-Unicode data types. SQL Server Setup will detect the Windows collation on the computer where SQL Server is being installed. Then, it will automatically select the collation for your instance of SQL Server. Sorts and comparisons might be different for Unicode and non-Unicode characters.

Use the following table to determine which , if any, collation you should choose.

#### <sup>I</sup>∕ Note:

If you upgrade from an earlier version of a SQL Server collation setting, no collation choice is required.

Task	Collation	
Install on a new system with no requirements for synchronizing with any existing system.	Use the collation that is identified by Setup, and then choose the desired binary, case, or other option. When SQL Server Setup detects that the computer is running the U.S. English system locale, Setup automatical selects the <b>Dictionary order, case-insensitive, for use wi</b> <b>1252 character set</b> SQL Server 2008 collation. To select the equivalent Windows collation, select <b>Collation designator</b> , select <b>Latin1_General</b> , and the cle both <b>case-sensitive</b> and <b>accent-sensitive</b> .	
Upgrade an installation of SQL Server 6.5 or SQL Server 7.0 to a default instance of SQL Server, or to install a default instance of SQL Server 2008 that will version- switch with an installation of SQL Server 6.5.	Use the SQL Server 2008 collation that Setup selects.	
Synchronize with an existing instance of SQL Server.	Select <b>SERVERPROPERTY(N'Collation')</b> from the Code Editor on the existing instance, and specify <i>Collation</i> . For more information, see <b>SERVERPROPERTY (Transact- SQL)</b> . 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.It consists of the collation designator name followed by a description of the binary, case, accent, kana,	

	and width-sensitivity options that are specified. Select the same Windows collation designator and sorting options in SQL Server Setup. For more information, see <u>Windows Collation Sorting</u> <u>Styles</u> .
Synchronize with an existing installation of SQL Server 6.5, SQL Server 7.0, or SQL Server 2000.	Run <b>sp_helpsort</b> on the existing system. Then, use the sort ID to select a SQL Server collation to make your instance of SQL Server compatible with an existing installation. For more information, see <u>Selecting a SQL Server</u> <u>Collation</u> .
Synchronize with the Windows system locale of another computer.	In Control Panel, find the locale name under <b>Regional</b> <b>Options</b> (Windows 2000, or Windows Server 2003), or <b>Regional and Language Options</b> (Windows XP), and then use the table that is provided in <u>Collation Settings in Setup</u> . Set the sorting options as described in <u>Windows Collation</u> <u>Sorting Styles</u> .

# <sup>ℤ</sup>Note:

When you perform an action that depends on collations, the SQL Server collation that is used by the referenced object must use a code page that is supported by the operating system.

## □ See Also

#### Reference

Selecting a SQL Server Collation

#### **Other Resources**

Setting Client Code Pages SQL Server Collation Name (Transact-SQL)

SQL Server Setup Help Selecting a SQL Server Collation

# <u> See Also</u> See Also See See Also See Also

#### - -

<u>Planning a SQL Server Installation</u> > <u>Collation and Unicode Support</u> > <u>Using SQL Server Collations</u> >

This topic describes SQL Server 2008 collation options for instances of SQL Server that require compatibility with versions of SQL Server that do not use collations.

The following SQL 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
44	noaccent.850	Dictionary order, case- insensitive, accent-	SQL_Latin1_General_Cp850_CI

		insensitive	
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_C
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_Scandainavian_Pref_Cp850
59	scandict.850	Scandinavian dictionary sort order, case- sensitive	SQL_Scandainavian_Cp850_CS_
60	scannocase.850	Case-insensitive Scandinavian dictionary sort order, without case preference	SQL_Scandainavian_Cp850_CI_
61	altnocs.850	Alternate dictionary 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_A
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- sensitive	SQL_Latin1_General_Cp1251_C
		Dictionary order, case-	

106	nocase.251	insensitive	SQL_Latin1_General_Cp1251_C
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_C
114	nocase.253	Dictionary order, case- insensitive	SQL_Latin1_General_Cp1253_C
121	graltdct.253	Dictionary order, case- sensitive, accent- sensitive	SQL_Latin1_General_Cp1253_C
124	grnoacce.253	Dictionary order, case- insensitive, accent- insensitive	SQL_Latin1_General_Cp1253_C
128	binary.254	Binary	SQL_Latin1_General_1254_BIN
129	diction.254	Dictionary order, case- sensitive	SQL_Latin1_General_Cp1254_C
130	nocase.254	Dictionary order, case- insensitive	SQL_Latin1_General_Cp1254_C
136	binary.255	Binary	SQL_Latin1_General_1255_BIN
137	diction.255	Dictionary order, case- sensitive	SQL_Latin1_General_Cp1255_C
138	nocase.255	Dictionary order, case- insensitive	SQL_Latin1_General_Cp1255_C
144	binary.256	Binary	SQL_Latin1_General_1256_BIN
145	diction.256	Dictionary order, case- sensitive	SQL_Latin1_General_Cp1256_C
146	nocase.256	Dictionary order, case- insensitive	SQL_Latin1_General_Cp1256_C
152	binary.257	Binary	SQL_Latin1_General_1257_BIN
153	diction.257	Dictionary order, case- sensitive	SQL_Latin1_General_Cp1257_C
154	nocase.257	Dictionary order, case- 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	lvidic.257	Latvian dictionary order, case-sensitive	SQL_Latvian_Cp1257_CS_AS_F
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_(
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

## <sup>I</sup>Note:

Each SQL collation name has an equivalent string in Transact-SQL code. For more information, see **SQL Server Collation Name (Transact-SQL)**.

## $\Box$ See Also

#### Reference

Collation Settings in Setup

#### Concepts

Using SQL Server Collations

SQL Server Setup Help Product Specifications for SQL Server 2008

<u>Planning a SQL Server Installation</u> >

The following topics provide product specification details for this release.

Editions and Components of SQL Server 2008

Selecting components and features to install.

Maximum Capacity Specifications for SQL Server 2008

Maximum capacities for various SQL Server objects.

Maximum Number of Processors Supported by the Editions of SQL Server 2008 The maximum number of processors supported by each edition of SQL Server.

#### SQL Server Setup Help Maximum Capacity Specifications for SQL Server 2008

#### - -

<u>Planning a SQL Server Installation</u> > <u>Product Specifications for SQL Server 2008</u> >

The following tables specify the maximum sizes and numbers of various objects defined in SQL Server components.

# <sup>□</sup> Database Engine Objects

The following table specifies the maximum sizes and numbers of various objects defined in SQL Server databases or referenced in Transact-SQL statements.

SQL Server Database Engine object	Maximum sizes/numbers SQL Server (32-bit)	Maximum sizes/numbers SQL Server (64-bit)
Batch size <sup>1</sup>	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 <sup>2</sup>	900	900
Bytes per foreign key	900	900
Bytes per primary key	900	900
Bytes per row <sup>8</sup>	8,060	8,060
Bytes in source text of a stored procedure	Lesser of batch size or 250 MB	Lesser of batch size or 250 MB
Bytes per <b>varchar(max), varbinary(max), xml, text,</b> or <b>image</b> column	2^31-1	2^31-1
Characters per <b>ntext</b> or <b>nvarchar(max)</b> 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 <sup>7</sup>	16	16

Columns per foreign key	16	16
Columns per primary key	16	16
Columns per nonwide table	1,024	1,024
Columns per wide table	30,000	30,000
Columns per SELECT statement	4,096	4,096
Columns per INSERT statement	4096	4096
		Maximum
Connections per client	Maximum value of configured	value of
Connections per chent	connections	configured
		connections
Database size	524.272 terabytes	524,272
		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	253	253
Identifier length (in characters)	128	128
racialitici lengui (in characters)	50 instances on a stand-alone	120
	server for all SOL Server	50 instances
	editions except for	on a stand-
	Workgroup. Workgroup	alone server.
Instances per computer	supports a maximum of 16	
	instances per computer.	25 instances on a failover
	SQL Server supports 25	cluster.
	instances on a failover cluster.	
I ongth of a string containing SOL		65,536 *
	65,536 * Network packet size	Network
statements (batch size) <sup>2</sup>		packet size
Locks per connection		Maximum
	Maximum locks per server	locks per
		server
Locks per instance of SQL		Limited only

Server <sup>5</sup>	Up to 2,147,483,647	by memory
Nested stored procedure levels <sup>6</sup>	32	32
Nested subqueries	32	32
Nested trigger levels	32	32
Nonclustered indexes per table	999	999
Number of distinct expressions in the GROUP BY clause when any of the following are present: CUBE, ROLLUP, GROUPING SETS, WITH CUBE, WITH ROLLUP	32	32
Number of grouping sets generated by operators in the GROUP BY clause	4,096	4,096
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 <sup>3</sup>	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	30,000	30,000
Tables per SELECT statement	256	256
Triggers per table <sup>3</sup>	Limited by number of objects in a database	Limited by number of objects in a database
Columns per UPDATE statement (Wide Tables)	4096	4096
User connections	32,767	32,767

XML indexes 249 249	
---------------------	--

<sup>1</sup>Network 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 KB, and is controlled by the **network packet size** configuration option.

<sup>2</sup>The maximum number of bytes in any index key cannot exceed 900 in SQL Server. 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, you can include nonkey columns in a nonclustered index to avoid the maximum index key size of 900 bytes.

<sup>3</sup>Database objects include objects such as tables, views, stored procedures, userdefined functions, triggers, rules, defaults, and constraints. The sum of the number of all objects in a database cannot exceed 2,147,483,647.

<sup>4</sup>Although 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.

<sup>5</sup>This value is for static lock allocation. Dynamic locks are limited only by memory.

<sup>6</sup>If a stored procedure accesses more than 8 databases, or more than 2 databases in interleaving, you will receive an error.

<sup>7</sup>If 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, 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**.

<sup>8</sup> SQL Server 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 Books Online.

# Replication Objects

The following table specifies the maximum sizes and numbers of various objects defined in SQL Server Replication.

SQL Server Replication object	Maximum sizes/numbers SQL Server (32-bit)	Maximum sizes/numbers SQL Server (64-bit)
Articles (merge publication)	256	256
Articles (snapshot or transactional publication)	32,767	32,767
Columns in a table <sup>1</sup> (merge publication)	246	246
Columns in a table <sup>2</sup> (SQL Server snapshot or transactional publication)	1,000	1,000
Columns in a table <sup>2</sup> (Oracle snapshot or transactional publication)	995	995
Bytes for a column used in a row filter (merge publication)	1,024	1,024
Bytes for a column used in a row filter (snapshot or transactional publication)	8,000	8,000

<sup>1</sup>If 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**.

<sup>2</sup>The 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.

## $\Box$ See Also

#### Reference

Hardware and Software Requirements for Installing SQL Server 2008 Check Parameters for the System Configuration Checker

SQL Server Setup Help

Maximum Number of Processors Supported by the Editions of SQL Server 2008



#### - -

<u>Planning a SQL Server Installation</u> > <u>Product Specifications for SQL Server 2008</u> >

SQL Server supports the specified number of processor sockets multiplied by the number of logical CPUs in each socket. For example, the following is considered a single processor for purposes of this table:

- A single-core, hyper-threaded processor with 2 logical CPUs per socket.
- A dual-core processor with 2 logical CPUs.
- A quad-core processor with 4 logical CPUs.

SQL Server is licensed per processor socket, and not per logical CPU basis.

SQL Server 2008 edition	Number of processors supported
Enterprise	Operating system maximum
Standard	4
Web	4
Workgroup	2
Express	1
Express with Advanced Services	1

## □ See Also

#### Reference

Memory Supported by the Editions of SQL Server 2008 Hardware and Software Requirements for Installing SQL Server 2008

#### **Other Resources**

Product Specifications for SQL Server 2008

#### SQL Server Setup Help Memory Supported by the Editions of SQL Server 2008

- -

<u>Planning a SQL Server Installation</u> > <u>Product Specifications for SQL Server 2008</u> >

The following table specifies the maximum memory support for each edition of SQL Server 2008.

SQL Server edition	Maximum memory supported
Enterprise	OS maximum
Developer	OS maximum
Standard	OS maximum
Web	OS maximum
Workgroup	4 GB (64-bit), OS maximum (32-bit)
Express	1 GB
Express with Advanced Services	1 GB

## □ See Also

#### Reference

Hardware and Software Requirements for Installing SQL Server 2008

#### **Other Resources**

Product Specifications for SQL Server 2008

SQL Server Setup Help High Availability Solutions Overview



#### - -

<u>Planning a SQL Server Installation</u> >

This section introduces several SQL Server 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 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 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 Enterprise and SQL Server Developer, and, with some restrictions, in SQL Server Standard. For more information about failover clustering,
see <u>Getting Started with SQL Server 2008 Failover Clustering</u> and <u>Installing a SQL Server 2008 Failover Cluster</u>.

• 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, you can use it indirectly for reporting 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 highperformance 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 several seconds. Database mirroring partners and witnesses are supported by SQL Server 2005 Standard SP1 and later versions, and by SQL Server 2005 Enterprise Edition SP1 and later versions. But the partners must use the same edition, and **asynchronous database mirroring (highperformance mode)** is supported only by SQL Server 2005 Enterprise Edition SP1 and later versions. Witnesses are also supported by SQL

Edition SP1 and later versions. Witnesses are also supported by SQL Server 2005 Workgroup SP1 and later versions and by SQL Server 2005 Express Edition SP1 and later versions. For more information about database mirroring, see **Database Mirroring**.

• Log shipping

Like database mirroring, log shipping operates at the database level. You can use log shipping to maintain one or more warm standby databases for a corresponding production database that is referred to as the *primary database*. Standby databases are also referred to as *secondary databases*. Each secondary database is created by restoring a database backup of the primary database with no recovery, or with standby. Restoring with standby lets you use the resulting secondary database for limited reporting.

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 set 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-todate 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 the SQL Server Enterprise, Standard, and Workgroup editions. For more information about log shipping, see **Log Shipping Overview** and **Log Shipping Administration**.

• Replication

Replication uses a publish-subscribe model. This lets a primary server, referred to as the Publisher, distribute data to one or more secondary servers, or Subscribers. Replication enables real-time availability and scalability across these servers. It supports filtering to provide a subset of data at Subscribers, and also allows for 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 usually used for high availability. For more information,

#### see Improving Scalability and Availability.

Replication is supported in all editions of SQL Server. Replication publishing is not available with SQL Server Express or SQL Server Compact 3.5.

#### <sup>Important:</sup>

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

• Scalable shared databases

The scalable shared database feature lets you scale out a read-only database built exclusively for reporting. The reporting database must reside on a set of dedicated, read-only volumes whose primary purpose is hosting the database. By using commodity hardware for servers and volumes, you can scale out a reporting database that provides the same view of the reporting data on multiple reporting servers. This feature also allows a smooth update path for the reporting database. For more information, see **Scalable Shared Databases Overview**.

# <sup>□</sup> In This Section

Торіс	Description
Selecting a High Availability	Presents considerations for selecting a high
<u>Solution</u>	availability solution.

# $\Box$ See Also

#### Reference

Hardware and Software Requirements for Installing SQL Server 2008

#### **Other Resources**

Database Mirroring Overview Log Shipping Overview SQL Server Replication

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Selecting a High Availability Solution

- -

<u>Planning a SQL Server Installation</u> > <u>High Availability Solutions Overview</u> >

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 signed 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

# <sup>ℤ</sup>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 it operates 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.

• 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.

Asynchronous database mirroring has the potential advantage over log shipping 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.

## <sup>✓</sup> Note:

For information about how to use log shipping with database mirroring, see **Database Mirroring and Log Shipping**.

#### • Replication

Replication offers the following benefits:

- Allows filtering in 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.

## <sup>I</sup>Note:

For information about how to use other high-availability solutions

with replication, see **Replication and Database Mirroring** and **Replication and Log Shipping**.

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Getting Started with SQL Server 2008 Failover Clustering



#### - -

<u>Planning a SQL Server Installation</u> > <u>High Availability Solutions Overview</u> >

Failover clustering in SQL Server 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.

Failover clustering has a new architecture and new work flow for all Setup scenarios in SQL Server 2008. The two options for installation are Integrated installation and Advanced/Enterprise installation. Integrated installation creates and configures a single-node SQL Server failover cluster instance. Additional nodes are added using add node functionality in Setup. For example, for Integrated installation, you run Setup to create a single-node failover cluster. Then, you run Setup again for each node you want to add to the cluster. Advanced/Enterprise installation consists of two steps. The Prepare step prepares all nodes of the failover cluster to be operational. Nodes are defined and prepared during this initial step. After you prepare the nodes, the Complete step is run on the active node—the node that owns the shared disk—to complete the failover cluster instance and make it operational.

# <sup>□</sup> 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 <u>Installing a SQL Server 2008</u> <u>Failover Cluster</u>.
- Allow one failover cluster node to fail over to any other node in the failover cluster configuration. For more information, see <u>Installing a SQL Server</u> <u>2008 Failover Cluster</u>.
- 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 the Microsoft Search service with failover clustering. For more information, see <u>Using SQL Server Tools with</u> Failover Clustering.

# <sup>□</sup> 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.Use SQL Server Setup in a failover cluster to do the following:

- Install SQL Server to create a failover cluster. 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 <u>Before Installing Failover Clustering</u>. All nodes in a failover cluster must be of the same platform, either 32-bit or 64-bit, and must run the same operating system edition and version. Also, 64-bit SQL Server editions must be installed on 64-bit hardware running the 64-bit versions of Windows operating systems. There is no WOW64 support for failover clustering in this release.
- Add or remove nodes from a failover cluster configuration without affecting the other cluster nodes. For more information, see <u>How to: Add or</u> <u>Remove Nodes in a SQL Server Failover Cluster (Setup)</u>.
- Specify multiple IP addresses for each failover cluster instance. You can specify multiple IP addresses for each server, allowing you to use all available network IP subnets. SQL Server limits you to one IP address per network. This provides alternative network connections if one subnet fails.

# <sup>□</sup> Failover Clustering Support

In SQL Server, the number of nodes supported for failover clustering depends on the operating system and the edition of SQL Server. For a list of tools, features, and services supported with SQL Server failover clustering, see **Features Supported by the Editions of SQL Server 2005**.

# $\Box$ See Also

#### Tasks

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

#### Concepts

Before Installing Failover Clustering How to: Install SQL Server 2008 from the Command Prompt

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Before Installing Failover Clustering



#### - -

<u>Planning a SQL Server Installation</u> > <u>High Availability Solutions Overview</u> > <u>Getting Started with SQL</u> <u>Server 2008 Failover Clustering</u> >

Before you install a SQL Server failover cluster, you must select the hardware and the operating system on which SQL Server 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**

• For Windows Server 2003, your hardware must be listed on the <u>Windows</u> <u>Catalog and Hardware Compatibility List</u>. The hardware system must appear under the category of a cluster solution. Windows Server 2008 has a utility that assesses hardware compatibility for failover cluster solutions.

<sup>I</sup> 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 Windows Catalog and Windows Hardware Compatibility List, specify "cluster" as the category. All other categories are for OEM use. For more information, see <u>The Microsoft support policy for server clusters, the Hardware Compatibility List, and the Windows Server Catalog</u>.

- 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 Windows Catalog and Hardware Compatibility List.
- 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 <u>Windows clustering and geographically separate sites</u> in the Microsoft Knowledge Base.
- SAN configurations are also supported on Windows 2000 Advanced Server and Datacenter Server editions. The 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

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 <u>The Datacenter Program and Windows 2000</u> <u>Datacenter Server Product</u> in the Microsoft Knowledge Base.

- If you deploy a SQL Server failover cluster on iSCSI technology components, we recommend that you use appropriate caution. For more information, see <u>Support for SQL Server 2000 on iSCSI technology</u> <u>components</u> in the Microsoft Knowledge Base.
- For support information, see <u>SQL Server support policy for Microsoft</u> <u>Clustering</u> in the Microsoft Knowledge Base.
- 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 Microsoft Knowledge Base article, <u>Quorum Drive Configuration</u> <u>Information</u>.

• To install a SQL Server 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, see <u>Hardware and</u> <u>Software Requirements for Installing SQL Server 2008</u>.

- 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.
- SQL Server 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.

Additional mount point considerations for SQL Server 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. Volume GUIDs are not supported in this release.
- 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 validates drive configuration as part of a failover cluster installation.
- SQL Server Setup automatically sets dependencies between the SQL Server cluster group and the disks that will be in the failover cluster. Do not set dependencies for disks before Setup.
- To enable Kerberos authentication with SQL Server 2008, see <u>How to use</u> <u>Kerberos authentication in SQL Server</u> in the Microsoft Knowledge Base.

#### **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 Enterprise or SQL Server Standard in conjunction with MSCS. SQL Server Enterprise supports failover clusters with up to 8 nodes. SQL Server Standard supports 2-node failover clusters.

For more information about installing and configuring MSCS on Windows Server 2003, see <u>Server clusters</u>.

- The resource DLL for the SQL Server service exports two functions used by MSCS Cluster manager to check for availability of the SQL Server resource. There is a simple check, LooksAlive, that queries the service status through the Windows NT Service Control Manager. There is also a more stringent check, IsAlive, that connects to SQL Server as a user probe to perform a simple query. By default, LooksAlive is fired every 5 seconds, and IsAlive is fired every 60 seconds. The LooksAlive and IsAlive polling intervals can be changed in MSCS Cluster Administrator from the **Advanced** tab for the SQL Server resource or by using the Cluster.exe command prompt utility.
- MSCS must be able to verify that the failover clustered instance is running by using the IsAlive check. This requires connecting to the server by using a trusted connection. By default, the account that runs the cluster service is not configured as an administrator on nodes in the cluster, and the BUILTIN\Administrators group does not have permission to log into SQL Server. These settings change only if you change permissions on the cluster nodes.

Ensure that the group or 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 **SELECT** *@@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 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, Workstation Components, or if you will use distributed transactions, you must install MSDTC. Note that MSDTC is not required for Analysis Services-only instances.

## **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.

#### **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, <u>Recommended private</u> <u>"Heartbeat" configuration on a cluster server</u>.
- 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 installations in side-by-side configurations with previous versions, SQL Server services must use accounts found only in the global domains group. Additionally, accounts used by SQL Server services must not appear in the local Administrators group. Failure to comply with this guideline will result in unexpected security behavior.
- Windows Server 2003 cluster nodes in an environment where there are no pre-existing Windows Server 2003 domain controllers, see <u>Windows 2000</u> and <u>Windows Server 2003 cluster nodes as domain controllers</u>.
- 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.

#### <sup>I</sup>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 Microsoft Knowledge Base article, <u>Antivirus software may cause problems with cluster services</u>.
- 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 (')
  - Ampersand (&)

Also verify that existing cluster group names do not contain unsupported characters.

## Network, Port, and Firewall 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.

### <sup>Important:</sup>

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 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.
- To ensure correct failover cluster functionality, add exceptions to firewall configuration settings for the SQL Server port, SQL Browser port, File and Printer Sharing (TCP 139/445 and UDP 137/138), and Remote Procedure Call (TCP port 135).
- The Remote Registry service must be up and running.
- Remote Administration must be enabled.
- For the SQL Server port, use SQL Server Configuration Manager to check the SQL Server network configuration for the TCP/IP protocol for the instance you want to unblock. You must enable the TCP port for IPALL if you want to connect to SQL Server using TCP after installation. By default, SQL Browser listens on UDP port 1434.

## **Other Considerations**

- To create a failover cluster, you must be a local administrator with permissions to log on 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.
- On Windows Server 2008, service SIDs are generated automatically for use with SQL Server 2008 services. For SQL Server 2008 failover cluster instances upgraded from SQL Server 2000 or SQL Server 2005, existing domain groups and ACL configurations will be preserved.
- On Windows Server 2003, create domain groups for the clustered services that will be installed as part of your SQL Server failover cluster. The SQL Server service, SQL Server Agent service, Analysis Services service, and iFTS service must run as domain accounts that are members of the 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 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 <u>Check Parameters for the System Configuration</u> <u>Checker</u>.
- Consider whether the SQL Server tools, features, and components you want to use are supported with failover clustering. For more information, see <u>Getting Started with SQL Server 2008 Failover Clustering</u>.
- Consider how you will monitor and maintain your failover cluster to achieve your high availability goals. For more information, see <u>Maintaining</u> <u>a Failover Cluster</u> and <u>Using SQL Server Tools with Failover Clustering</u>.
- To reduce the time required to install a SQL Server failover cluster, you can pre-install .NET Framework 2.0 2.0 on all failover cluster nodes before running SQL Server Setup.

# □ See Also

#### Reference

Hardware and Software Requirements for Installing SQL Server 2008 Check Parameters for the System Configuration Checker

#### Concepts

Installing a SQL Server 2008 Failover Cluster Security Considerations for a SQL Server Installation

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Installing a SQL Server 2008 Failover Cluster



#### - -

<u>Planning a SQL Server Installation</u> > <u>High Availability Solutions Overview</u> > <u>Getting Started with SQL</u> <u>Server 2008 Failover Clustering</u> >

To install a SQL Server 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.

# <sup>□</sup> 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. 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 that includes SQL Server, SQL Server Agent, the Full-text Search (FTS) service, and Replication. You can install a failover cluster with SQL Server only, Analysis Services only, or SQL Server and Analysis Services.

# <sup>□</sup> 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 2008. Each instance of SQL Server requires just one cluster disk for data files.

# <sup>□</sup> 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:
  - <u>Before Installing Failover Clustering</u>
  - 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. Run the SQL Server Setup program to start your failover cluster installation. Failover clustering has a new architecture and new work flow for all Setup scenarios in SQL Server 2008. The two options for installation are Integrated installation and Advanced/Enterprise installation. Integrated installation creates and configures a single-node SQL Server failover cluster instance. Additional nodes are added by using the Add Node functionality in Setup. For example, for Integrated installation, you run Setup to create a single-node failover cluster. Then, you run Setup again for each node you want to add to the cluster. Advanced/Enterprise installation consists of two steps. The Prepare step prepares all nodes of the failover cluster to be operational. Nodes are defined and prepared during this initial step. After you prepare the nodes, the Complete step is run on the active node—the node that owns the shared disk—to complete the failover cluster instance and make it operational.

# $\Box$ See Also

#### Tasks

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

#### Concepts

Before Installing Failover Clustering How to: Install SQL Server 2008 from the Command Prompt

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Upgrading a SQL Server 2008 Failover Cluster



#### - ▼

High Availability Solutions Overview > Getting Started with SQL Server 2008 Failover Clustering > Installing a SQL Server 2008 Failover Cluster >

SQL Server supports upgrade of the Database Engine and Analysis Services from SQL Server 2000 and SQL Server 2005 failover clusters separately on all failover cluster nodes.

Support details are as follows:

- Upgrade is supported both through the user interface and from the command prompt. For more information, see <u>How to: Upgrade a SQL</u> <u>Server Failover Cluster Instance (Setup)</u> and <u>How to: Install SQL Server</u> <u>2008 from the Command Prompt</u>.
- Upgrading from SQL Server 2005 You can run upgrade from the command prompt on each failover cluster node, or by using the Setup UI to upgrade each cluster node. If Full-text search and Replication features do not exist on the instance being upgraded, they will be installed automatically with no option to omit them.
- Upgrading from SQL Server 2005 Similar to upgrade from SQL Server 2005: You can run upgrade from the command prompt on each failover cluster node, or by using the Setup UI to upgrade each cluster node. Supported for 32-bit scenarios only. Failover cluster upgrades from SQL Server 2000 (64-bit) are not supported. For more information, see <u>Version and Edition Upgrades</u>.
- Service pack installation you must apply SQL Server service packs and patches to SQL Server 2008 failover clusters separately on all nodes.
- The following scenarios are not supported in SQL Server 2008:
  - You cannot migrate from a stand-alone instance of SQL Server to a SQL Server 2008 failover cluster.
  - Add features to a SQL Server 2008 failover cluster. For example, you cannot add the Database Engine to an existing Analysis Services-only failover cluster.
  - You cannot downgrade a SQL Server 2008 failover cluster node

to a stand-alone instance of SQL Server 2008.

• For more information, see <u>Getting Started with SQL Server 2008</u> <u>Failover Clustering</u>.

# □ See Also

#### Tasks

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

### Reference

Version and Edition Upgrades

#### Concepts

How to: Install SQL Server 2008 from the Command Prompt

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help Maintaining a Failover Cluster



#### - -

<u>Planning a SQL Server Installation</u> > <u>High Availability Solutions Overview</u> > <u>Getting Started with SQL</u> <u>Server 2008 Failover Clustering</u> >

After you have installed a SQL Server 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.

# <sup>□</sup> 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 by running SQL Server Setup on the computer that you want to add to the cluster. For more information, see <u>How to: Create a New SQL Server Failover Cluster (Setup)</u> and <u>How to: Add or Remove Nodes in a SQL Server Failover Cluster (Setup)</u>.

# Removing a Node from an Existing Failover Cluster Instance

You can remove a node from a failover cluster configuration by running SQL Server Setup on the computer that you want to remove from the cluster. 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 <u>How to: Add or Remove Nodes in a SQL Server Failover Cluster (Setup)</u>.
## 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 Configuration Manager to find out the pipe name.

# <sup>□</sup> 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 by 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.

### <sup>I</sup>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.

On Windows Server 2008, service SIDs are used for SQL Server service accounts. For more information, see <u>Setting Up Windows Service Accounts</u>.

## □ 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 Failover Cluster Instance (Setup)

### Concepts

Before Installing Failover Clustering

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Optimizing Failover Cluster Performance



- -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Maintaining a Failover Cluster</u> >

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

# <sup>□</sup> 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.

# <sup>□</sup> 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.

## <sup>□</sup> 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.

# <sup>□</sup> 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.

# <sup>□</sup> Fixed IP Ports and Fixed Memory Settings

Use fixed IP ports and fixed memory settings.

## $\Box$ See Also

## **Other Resources** Setting Server Configuration Options

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Using SQL Server Tools with Failover Clustering



- -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Maintaining a Failover Cluster</u> >

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

# <sup>□</sup> 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 SQL Server must run on the same system account on all failover cluster nodes.

### <sup>I</sup>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 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.

# <sup>B</sup> 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 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.

# <sup>□</sup> 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.

# <sup>□</sup> 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.

# <sup>□</sup> 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.

# <sup>B</sup> SQL Mail

SQL Mail can be configured remotely by Configuration Manager.

## $\Box$ See Also

## **Other Resources**

**Features Supported by Editions of SQL Server 2005** 

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Failover Cluster Troubleshooting



#### - -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Maintaining a Failover Cluster</u> >

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.

## <sup>□</sup> Basic Troubleshooting Steps

When working with SQL Server failover clustering, remember that the server cluster consists of a failover cluster instance that runs under Microsoft Cluster Services (MSCS). The instance of SQL Server might 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, <u>Recommended</u> <u>Private "Heartbeat" Configuration on a Cluster Server</u>.
- 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.

# <sup>□</sup> 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 <u>How to: Create a New SQL Server Failover</u> <u>Cluster (Setup)</u> and <u>How to: Recover from Failover Cluster Failure in</u> <u>Scenario 1</u>.

• 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 <u>How to: Create a New SQL Server Failover</u> <u>Cluster (Setup)</u> and <u>How to: Recover from Failover Cluster Failure in Scenario 2</u>.

# 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

**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 <u>How to: View SQL Server 2008 Setup Log Files</u>.

**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.

# **Problem: SQL Server 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:

- 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 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 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 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 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.

## <sup>I</sup>Note:

For installations of SQL Server 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.

# 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 Serverdependent 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.

## $\Box$ See Also

### Concepts

How to: View SQL Server 2008 Setup Log Files

### **Other Resources**

How Extended Stored Procedures Work Execution Characteristics of Extended Stored Procedures

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Failover Clustering How-to Topics



#### - -

<u>Planning a SQL Server Installation</u> > <u>High Availability Solutions Overview</u> > <u>Getting Started with SQL</u> <u>Server 2008 Failover Clustering</u> >

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

## $\Box$ See Also

### Concepts

<u>Getting Started with SQL Server 2008 Failover Clustering</u> <u>Before Installing Failover Clustering</u>

© 2008 Microsoft Corporation. All rights reserved.

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

# 

#### - ▼

High Availability Solutions Overview > Getting Started with SQL Server 2008 Failover Clustering > Failover Clustering How-to Topics >

The SQL Server Installation Wizard lets you install a SQL Server failover cluster on a single node.

Use the following procedure to create a failover cluster instance.

To add nodes to or remove nodes from an existing failover cluster instance, see <u>How to: Add or Remove Nodes in a SQL Server Failover Cluster (Setup)</u>.

For remote installation, see <u>Version and Edition Upgrades</u>.

To install a stand-alone instance of SQL Server , see <u>How to: Install SQL Server</u> <u>2008 (Setup)</u>.

# <sup>D</sup> Prerequisites

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

- <u>Planning a SQL Server Installation</u>
- Before Installing Failover Clustering
- <u>Security Considerations for a SQL Server Installation</u>
- It is very important that you also read and explicitly follow instructions in the <u>recommended private "Heartbeat" configuration on a cluster server</u> Knowledge Base article.

### <sup>ℤ</sup>Note:

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

## <sup>□</sup> To install SQL Server 2008

- 1. Insert the SQL Server installation media, and from the root folder, double-click setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click setup.exe.
- 2. If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.
- 3. Windows Installer 4.5 is also required, and may be installed by the Installation Wizard. If you are prompted to restart your computer, restart and then start SQL Server 2008 Setup again.
- 4. After prerequisites are installed, the Installation Wizard will start the SQL Server Installation Center. To create a new installation of SQL Server 2008, click **New Installation or Add Features to an Existing**

#### Installation.

- 5. The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information, see <u>How to: View SQL</u> <u>Server 2008 Setup Log Files</u>.
- 6. On the Product key page, click a radio button to indicate whether you are installing a free edition of SQL Server, or whether you have a PID key for a production version of the product. For more information, see Editions and Components of SQL Server 2008.
- 7. On the License Terms page, read the license agreement, and then select the check box to accept the licensing terms and conditions. To continue, click **Next**. To end Setup, click **Cancel**.
- 8. The Installation Wizard will install SQL Server prerequisites if they are not already on the computer. They include the following:
  - .NET Framework 2.0
  - SQL Server Native Client
  - SQL Server Setup Support Files

To install prerequisites, click Install.

- 9. The System Configuration Checker will verify the system state of your computer before Setup continues. After the check is complete, click **Next** to continue.
- 10. On the Feature Selection page, select the components for your installation. A description for each component group appears in the right-hand pane after you select the feature name. You can select any combination of check boxes, but only the Database Engine and Analysis Services support failover clustering. Other components will run on a single failover cluster node.

You can specify a custom directory for shared components using the field at the bottom of this page. To change the installation path for shared components, either update the path name in the field provided at the bottom of the dialog box, or click **Browse** to navigate to an

installation directory. The default installation path is C:\Program Files\Microsoft SQL Server\.

11. On the Instance Configuration page, specify whether to install a default or a named instance. For more information, see <u>Instance Configuration</u>.

**Instance ID suffix** — By default, the instance name is used as the Instance ID suffix. This is used to identify installation directories and registry keys for your instance of SQL Server. This is the case for default instances and named instances. For a default instance, the instance name and instance ID suffix would be MSSQLSERVER. To use a non-default instance ID suffix, select the **Instance ID suffix** check box and provide a value.

## <sup>I</sup>Note:

Typical stand-alone instances of SQL Server 2008, whether default or named instances, do not use a non-default value for the **Instance ID suffix** check box.

**Instance root directory** — By default, the instance root directory is C:\Program Files\Microsoft SQL Server\. To specify a non-default root directory, use the field provided, or click **Browse** to locate an installation folder.

All SQL Server service packs and upgrades will apply to every component of an instance of SQL Server.

**Detected instances and features** - The grid will show instances of SQL Server that are on the computer where Setup is running. If a default instance is already installed on the computer, you must install a named instance of SQL Server 2008. To continue, click **Next**.

- 12. Use the Cluster Resource Group page to specify the disk where SQL Server virtual server resources will be located. To specify the SQL Server cluster resource group name. You have two options:
  - Use the drop-down box to specify an existing group to use.
  - Type the name of a new group to create. Note that the name "Available storage" is not a valid group name.

- 13. On the Cluster Disk Selection page, select the shared cluster disk resource for your SQL Server failover cluster. The cluster disk is where the SQL Server data will be placed. Only one disk can be specified. If you select the group containing the cluster quorum resource, a warning will be displayed. It is recommended that you do not install to the cluster quorum resource. The Available shared disks grid displays a list of available disks, whether each is qualified as a shared disk, and a description of each disk resource. To continue, click Next.
- 14. On the Cluster Network Selection page, specify the network resources for your failover cluster instance:
  - **SQL Server Failover Cluster Network Name** This is the name used to identify your failover cluster instance on the network.
  - **Network Settings** Specify the IP type and IP address for your failover cluster instance.

To continue, click Next.

- 15. Use the Cluster Security Policy page to specify global or local domain groups for clustered services. All resource permissions are controlled by domain-level groups that include SQL Server service accounts as group members. For information about service security IDs (SID) functionality on Windows Server 2008, see <u>Setting Up Windows</u> <u>Service Accounts</u>. On Windows Server 2003, specify domain groups for SQL Server services. To continue, click **Next**.
- 16. The Disk Space Requirements page calculates the required disk space for the features you specify, and compares requirements to the available disk space on the computer where Setup is running. For more information, see <u>Disk Cost Summary</u>.
- 17. Work flow for the remainder of this topic depends on the features you have specified for your installation. You might not see all of the pages, depending on your selections.
- 18. On the Server Configuration Service Accounts page, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features you selected to install.
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, are started manually, or are disabled. Microsoft recommends that you configure service accounts individually to provide least privileges for each service, where SQL Server services are granted the minimum permissions they need to complete their tasks. For more information, see <u>SQL Server</u> <u>Configuration - Service Accounts</u> and <u>Setting Up Windows Service</u> <u>Accounts</u>.

To specify the same logon account for all service accounts in this instance of SQL Server, provide credentials in the fields at the bottom of the page.

**Security Note** Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

- 19. Use the **Server Configuration Collation** tab to specify non-default collations for the Database Engine and Analysis Services. For more information, see <u>SQL Server Configuration Collation</u>.
- 20. Use the Database Engine Configuration Account Provisioning page to specify the following:
  - Security Mode select Windows Authentication or Mixed Mode Authentication for your instance of SQL Server. If you select Mixed Mode Authentication, you must provide a strong password for the built-in SQL Server system administrator account. After a device establishes a successful connection to SQL Server, the security mechanism is the same for both Windows Authentication and Mixed Mode. For more information, see Database Engine Configuration - Account Provisioning.
  - SQL Server Administrators You must specify at least one system administrator for the instance of SQL Server. To add the account under which SQL Server Setup is running, click Add Current User. To add or remove accounts from the list of system administrators, click Add or Remove, and then edit the list of

users, groups, or computers that will have administrator privileges for the instance of SQL Server. For more information, see <u>Database</u> <u>Engine Configuration - Account Provisioning</u>.

When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

21. Use the Database Engine Configuration - Data Directories page to specify non-default installation directories. To install to default directories, click **Next**.

#### <sup>I</sup> Important:

If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.

For more information, see <u>Database Engine Configuration - Data</u> <u>Directories</u>.

- 22. Use the Database Engine Configuration FILESTREAM page to enable FILESTREAM for your instance of SQL Server. For more information about FILESTREAM, see <u>Database Engine Configuration Filestream</u>. To continue, click Next.
- 23. Use the Analysis Services Configuration Account Provisioning page to specify users or accounts that will have administrator permissions for Analysis Services. You must specify at least one system administrator for Analysis Services. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for Analysis Services. For more information, see <u>Analysis Services Configuration Account Provisioning</u>.

When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

24. Use the Analysis Services Configuration — Data Directories page to specify non-default installation directories. To install to default directories, click **Next**.

#### <sup>I</sup>Important:

If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.

For more information, see <u>Analysis Services Configuration - Data</u> <u>Directories</u>.

- 25. Use the Reporting Services Configuration page to specify the type of Reporting Services installation to create. Options include the following:
  - Native mode default configuration
  - SharePoint mode default configuration
  - Unconfigured Reporting Services installation

For more information about Reporting Services configuration modes, see <u>Report Server Installation Options</u>.

- 26. On the Error and Usage Reporting page, specify the information you would like to send to Microsoft that will help to improve SQL Server. By default, options for error reporting and feature usage are enabled. For more information, see <u>Error and Usage Report Settings</u>.
- 27. The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified.
- 28. The Ready to Install page displays a tree view of installation options that were specified during Setup. To continue, click **Install**.
- 29. During installation, the Installation Progress page provides status so you can monitor installation progress as Setup proceeds.
- 30. After installation, the Complete page provides a link to the summary log

file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.

31. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For information about Setup log files, see <u>How to: View SQL</u> <u>Server 2008 Setup Log Files</u>.

# <sup>□</sup> Next Steps

**Configure your new SQL Server installation** — To reduce the attackable surface area of a system, SQL Server selectively installs and activates key services and features. For more information, see <u>Understanding Surface Area</u> <u>Configuration</u> on MSDN.

For information about log file locations, see <u>How to: View SQL Server 2008</u> <u>Setup Log Files</u>.

#### Concepts

How to: Install SQL Server 2008 from the Command Prompt

SQL Server Setup Help

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



#### - -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Failover Clustering How-to Topics</u> >

Use this procedure to manage nodes to an existing SQL Server failover cluster instance.

#### <sup>Important:</sup>

To update or remove a SQL Server failover cluster, you must be a local administrator with permission to log in as a service on all nodes of the failover cluster. 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 view procedural steps to add or remove nodes, select one of the following operations:

- Add a node to an existing SQL Server 2008 failover cluster
- <u>Remove a node from an existing SQL Server 2008 failover cluster</u>

# □ Add Node

#### <sup>□</sup> To add a node to an existing SQL Server 2008 failover cluster

1.

Insert the SQL Server installation media, and from the root folder, double-click setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click setup.exe.

• If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.

• Windows Installer 4.5 is also required, and may be installed by the Installation Wizard. If you are prompted to restart your computer, restart, and then start SQL Server 2008 setup.exe again.

• When prerequisites are installed, the Installation Wizard will launch the SQL Server Installation Center. To add a node to an existing failover cluster instance, click **Installation** in the left-hand pane. Then, select **Add node to a SQL Server failover cluster**.

• The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information about log files, see <u>How to: View SQL Server</u> <u>2008 Setup Log Files</u>.

• On the Product key page, specify the PID key for a production version of the product. Note that the product key you enter for this installation must be for the same SQL Server 2008 edition as that which is installed on the active node.

• On the License Terms page, read the license agreement, and then select the check box to accept the licensing terms and conditions. To continue, click **Next**.

To end Setup, click **Cancel**.

• The Installation Wizard will install SQL Server prerequisites if they are not already on the computer. They include the following:

- .NET Framework 2.0
- SQL Server Native Client
- SQL Server Setup Support Files

To install prerequisites, click **Install**.

• The System Configuration Checker will verify the system state of your computer before Setup continues. After the check is complete, click **Next** to continue.

• On the Cluster Node Configuration page, use the drop-down box to specify the name of the SQL Server 2008 failover cluster instance that will be modified during this Setup operation.

• On the Server Configuration — Service Accounts page, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features you selected to install. For failover cluster installations, account name and startup type infromation will be pre-populated on this page based on settings provided for the active node. You must provide passwords for each account. For more information, see <u>SQL Server</u> <u>Configuration - Service Accounts</u> and <u>Setting Up Windows Service Accounts</u>.

**Security Note** Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

• The Disk Space Requirements page calculates the required disk space for the features you specify, and compares requirements to the available disk space on the computer where Setup is running. For more information, see <u>Disk Cost</u> <u>Summary</u>.

• On the Error and Usage Reporting page, specify the information you would like to send to Microsoft that will help to improve SQL Server. By default, options for error reporting and feature usage are enabled. For more information,

see Error and Usage Report Settings.

• The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified.

• The Ready to Install page displays a tree view of installation options that were specified during Setup.

• During installation, the Installation Progress page provides status so you can monitor installation progress as Setup proceeds.

• After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.

• If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For information about Setup log files, see <u>How to: View SQL Server 2008 Setup Log Files</u>.

# <sup>□</sup> Remove Node

# <sup>□</sup> To remove a node from an existing SQL Server 2008 failover cluster

1.

Insert the SQL Server installation media. From the root folder, double-click setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click setup.exe.

• If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.

• Windows Installer 4.5 is also required, and may be installed by the Installation Wizard. If you are prompted to restart your computer, restart and then start SQL Server 2008 setup.exe again.

• When prerequisites are installed, the Installation Wizard will launch the SQL Server Installation Center. To add a node to an existing failover cluster instance, click **Installation** in the left-hand pane, and then select **Add node to a SQL Server failover cluster**.

• The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information about log files, see <u>How to: View SQL Server</u> <u>2008 Setup Log Files</u>.

• The Installation Wizard will install SQL Server prerequisites if they are not already on the computer. They include the following:

- .NET Framework 2.0
- SQL Server Native Client

• SQL Server Setup Support Files

To install prerequisites, click **Install**.

• The System Configuration Checker will verify the system state of your computer before Setup continues. After the check is complete, click Next to continue.

• On the Cluster Node Configuration page, use the drop-down box to specify the name of the SQL Server 2008 failover cluster instance that will be modified during this Setup operation. The node that will be removed will be listed in the "Name of this node" field.

• On the Error and Usage Reporting page, specify the information you would like to send to Microsoft that will help to improve SQL Server. By default, options for error reporting and feature usage are enabled. For more information, see <u>Error and Usage Report Settings</u>.

• The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified.

- The Ready to Install page displays a tree view of options that were specified during Setup. To continue, click **Remove**.
- During the remove operation, the Progress page provides status.

• The Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation, click **Close**. For information about Setup log files, see <u>How to: View SQL Server 2008 Setup Log Files</u>.

# Concepts

How to: View SQL Server 2008 Setup Log Files

### SQL Server Setup Help How to: Repair a Failed SQL Server 2008 Installation

# 

#### - -

High Availability Solutions Overview > Getting Started with SQL Server 2008 Failover Clustering > Failover Clustering How-to Topics >

Use this procedure to repair a corrupt SQL Server 2008 installation.

Before you continue, for SQL Server failover clusters, review the following important information:

- Repair must be run on individual cluster nodes.
- To repair a failover cluster node after a failed Prepare operation, use **Remove node** and then perform the Prepare step again. See <u>How to: Add</u> or <u>Remove Nodes in a SQL Server Failover Cluster (Setup)</u>.
- •

# <sup>□</sup> Procedure Title

- 1. Launch the SQL Server Setup program (setup.exe) from SQL Server 2008 installation media.
- 2. After prerequisites and system verification, the Setup program will display the SQL Server Installation Center page.
- 3. Click **Maintenance** in the left-hand navigation area, and then click **Repair** to start the repair operation.
- 4. Setup support rule and file routines will run to ensure that your system has prerequisites installed and that the computer passes Setup validation rules. Click **OK** or **Install** to continue.
- 5. On the Select Instance page, select the instance to repair, and then click **Next** to continue.

- 6. The repair rules will run to validate the operation. To continue, click **Next**.
- 7. The Ready to Repair page indicates that the operation is ready to proceed. To continue, click **Repair**.
- 8. The Repair Progress page shows the status of the repair operation. The Complete page indicates that the operation is finished.

#### Concepts

Troubleshooting an Installation of SQL Server Failover Cluster Troubleshooting How to: View SQL Server 2008 Setup Log Files

# SQL Server Setup Help How to: Remove a SQL Server Failover Cluster Instance (Setup)

See Also

#### - -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Failover Clustering How-to Topics</u> >

Use this procedure to uninstall a SQL Server failover clustered instance.

#### <sup>Important:</sup>

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

#### Before you begin

Consider the following important points before you uninstall a SQL Server failover cluster:

- On a computer running SQL Server 2008 along with previous SQL Server versions, Enterprise Manager and other programs that depend on SQL-DMO may be disabled. This can occur in the following situations:
  - Side-by-side installations of SQL Server 2008, 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 2008, where SQL Server 2000 is installed after SQL Server 2008.

This issue is due to removal of the registration for the SQL Server 2008 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 2008 are installed side-by-side in a clustered environment, SQL Server Native Client should not be uninstalled if you uninstall SQL Server 2008. The SQL Server Resource DLL from SQL Server 2008 always uses SQL Server Native Client to

connect to SQL Server. When SQL Server 2008 is installed, the SQL Server Cluster Resource DLL is upgraded. When SQL Server 2008 is uninstalled, the SQL Server Resource DLL is not replaced with the version required by SQL Server 2000. Leaving SQL Server Native Client on cluster nodes allows the SQL Server Resource DLL to continue to connect to instances of SQL Server 2000.

- If SQL Server Native Client is uninstalled by accident, SQL Server resources will fail to start. To reinstall SQL Server Native Client, run the SQL Server Setup program to install SQL Server 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 <u>How to: Install SQL Server</u> <u>2008 from the Command Prompt</u>.

## <sup>□</sup> To uninstall a SQL Server failover cluster

1. To uninstall a SQL Server failover cluster, use the Remove Node functionality provided by SQL Server Setup to remove each node individually. For more information, see <u>How to: Add or Remove Nodes in a SQL Server Failover Cluster (Setup)</u>.

#### Tasks

How to: Read a SQL Server 2008 Setup Log File

#### Concepts

How to: View SQL Server 2008 Setup Log Files

SQL Server Setup Help How to: Recover from Failover Cluster Failure in Scenario 1

e Also

#### -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Failover Clustering How-to Topics</u> >

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.

## <sup>□</sup> To recover from a failover cluster hardware failure

- 1. After Node 1 fails, the 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 <u>Before Installing Failover Clustering</u>.
- 6. Ensure that the administrator accounts are the same on all cluster nodes. For more information, see <u>How to: Create a New SQL Server Failover</u> <u>Cluster (Setup)</u>.
- 7. Run SQL Server Setup to add Node 1 to the failover cluster. For more information, see <u>How to: Add or Remove Nodes in a SQL Server</u> <u>Failover Cluster (Setup)</u>.

#### Tasks

How to: Read a SQL Server 2008 Setup Log File

#### Concepts

How to: View SQL Server 2008 Setup Log Files

#### SQL Server Setup Help

#### How to: Recover from Failover Cluster Failure in Scenario 2

#### - -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Failover Clustering How-to Topics</u> >

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.

# <sup>D</sup> Procedure

## <sup>□</sup> To recover from a failover cluster failure

1.

After Node 1 fails, the 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 fail over 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 <u>Failover Cluster</u> <u>Troubleshooting</u>.

#### □ See Also

#### Tasks

How to: Read a SQL Server 2008 Setup Log File

#### Concepts

Failover Cluster Troubleshooting How to: View SQL Server 2008 Setup Log Files

#### **Other Resources**

How to: Rebuild the Master Database for SQL Server 2005 How to: Rebuild the Registry for SQL Server 2005

SQL Server Setup Help

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



#### - -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Failover Clustering How-to Topics</u> >

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.

# <sup>□</sup> Other Resources

You can use Kerberos authentication with SQL Server stand-alone instances or with SQL Server failover cluster instances. For additional information about enabling Kerberos, see the Knowledge Base article, "<u>How to use Kerberos</u> <u>authentication in SQL Server</u>."

# Concepts

Security Considerations for a SQL Server Installation

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

# See Also

#### - -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Failover Clustering How-to Topics</u> >

To maintain or update a SQL Server failover cluster, use **Add or Remove Programs** in Control Panel. For information about command prompt syntax, see <u>How to: Install SQL Server 2008 from the Command Prompt</u>.

Before you begin, review the following SQL Server Books Online topic: <u>Before</u> <u>Installing Failover Clustering</u>.

For additional information about changing the IP address of network adapters in the nodes of a cluster, see <u>this Microsoft Knowledge Base article</u>.

#### <sup>Important:</sup>

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.

# <sup>□</sup> To change the IP address of an existing SQL Server 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.

## <sup>ℤ</sup>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.

# Concepts

How to: View SQL Server 2008 Setup Log Files

## SQL Server Setup Help How to: Upgrade a SQL Server Failover Cluster Instance (Setup)

#### -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Failover Clustering How-to Topics</u> >

You can upgrade a SQL Server failvoer cluster to a SQL Server 2008 failover cluster using the SQL Server Installation Wizard or from the command prompt.

For upgrade information, see <u>Version and Edition Upgrades</u> and **Upgrading to SQL Server 2005**.

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

For sample syntax for command prompt installations, see <u>How to: Install SQL</u> <u>Server 2008 from the Command Prompt</u>.

# <sup>©</sup> Prerequisites

Before you begin, review the following important information:

- Before Installing Failover Clustering
- Review <u>Using Upgrade Advisor to Prepare for Upgrades</u>.
- Upgrading the Database Engine
- The <u>Recommended private heartbeat configuration</u> article in the Microsoft Knowledge Base.
- 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 an instance of SQL Server to a SQL Server 2008 failover cluster, the instance being upgraded must be a failover cluster. To upgrade a stand-alone instance of SQL Server to a SQL Server 2008 failover cluster, install a new SQL Server 2008 failover cluster, and then migrate user databases from the stand-alone instance using the Copy Database Wizard. For more information, see **Using the Copy Database Wizard**.
- You must upgrade one failover cluster node at a time. Failover during the upgrade operation has several options:
  - Move the resource group to a node that has already been upgraded.
  - Move the resource group to a node that has not yet been upgraded.
  - In the case of a single-node failover cluster, SQL Server Setup will take the SQL Server resource group offline.
- You cannot add features to an existing failover cluster.
- Changing the edition of failover cluster is limited to certain scenarios. See <u>Version and Edition Upgrades</u>.
- To control the failover behavior of cluster nodes during the upgrade process, run the upgrade operation from the command prompt and use the /FAILOVERCLUSTERROLLOWNERSHIP parameter. For more information, see <u>How to: Install SQL Server 2008 from the Command Prompt</u>.

## <sup>□</sup> To upgrade a SQL Server failover cluster

- 1. Insert the SQL Server installation media, and from the root folder, double-click setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click setup.exe.
- 2. If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.
- 3. Windows Installer 4.5 is also required, and may be installed by the Installation Wizard. If you are prompted to restart your computer, restart and then launch the SQL Server 2008 setup.exe again.
- 4. Once prerequisites are installed, the Installation Wizard will launch the SQL Server Installation Center. To upgrade an existing instance of SQL Server 2008, click "Upgrade from SQL Server 2000 or SQL Server 2005."
- 5. If Setup support files are required, SQL Server Setup will install them. If you are instructed to restrat your computer, restart before you continue.
- 6. The System Configuration Checker will run a discovery operation on your computer. To continue, click Ok. At this point, Setup log files have been created for your installation. For more information about log files, see <u>How to: View SQL Server 2008 Setup Log Files</u>.
- 7. On the Product key page, click a radio button to indicate whether you are upgrading to a free edition of SQL Server, or whether you have a PID key for a production version of the product. For more information, see Editions and Components of SQL Server 2008 and Version and Edition Upgrades. Enter your PID key in the space provided, then click Next to continue. Note that the PID key you use for a failover cluster upgrade must be consistent across all failover cluster nodes. All failover cluster nodes in the same SQL Server instance must be of the same SQL Server edition.
- 8. On the License Terms page, read the license agreement, and then select

the check box to accept the licensing terms and conditions. To continue, click **Next**. To end Setup, click **Cancel**.

- 9. On the Select Instance page, specify the SQL Server instance to upgrade to SQL Server 2008. To continue, click Next.
- 10. On the Feature Selection page, the features to upgrade will be preselected. A description for each component group appears in the righthand pane after you select the feature name. Note that you cannot change the features to be upgraded, and you cannot add features during the upgrade operation. To add features to an upgraded instance of SQL Server 2008 after the upgrade operation is complete, see <u>How to: Add</u> <u>Features to an Instance of SQL Server 2008 (Setup)</u>.
- 11. On the Instance Configuration page, specify whether to install a default or a named instance. For more information, see <u>Instance Configuration</u>. To continue, click Next.

**Instance ID suffix** - By default, the instance name is used as the Instance ID suffix. This is used to identify installation directories and registry keys for your instance of SQL Server. This is the case for default instances and named instances. For a default instance, the instance name and instance ID suffix would be MSSQLSERVER. To use a non-default instance ID suffix, select the **Instance ID suffix** check box and provide a value.

#### <sup>I</sup>∕ Note:

Typical stand-alone instances of SQL Server 2008, whether default or named instances, do not use a non-default value for the **Instance ID suffix** check box.

**Instance root directory** - By default, the instance root directory is C:\Program Files\Microsoft SQL Server\. To specify a non-default root directory, use the field provided, or click **Browse** to locate an installation folder.

All SQL Server service packs and upgrades will apply to every component of an instance of SQL Server.

**Detected instances and features** - The grid will show instances of SQL Server that are on the computer where Setup is running. If a default instance is already installed on the computer, you must install a named instance of SQL Server 2008. To continue, click **Next**.

- 12. The Disk Space Requirements page calculates the required disk space for the features you specify, and compares requirements to the available disk space on the computer where Setup is running. For more information, see <u>Disk Cost Summary</u>.
- 13. Work flow for the remainder of this topic depends on the features you have specified for your installation. You might not see all of the pages, depending on your selections.
- 14. On the Server Configuration Service Accounts page, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features you are upgrading.

Authentication and login inromation will be carried forward from the previous SQL Server instance. 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, are started manually, or are disabled. Microsoft recommends that you configure service accounts individually to provide least privileges for each service, where SQL Server services are granted the minimum permissions they need to complete their tasks. For more information, see <u>SQL Server Configuration - Service Accounts</u> and <u>Setting Up</u> <u>Windows Service Accounts</u>.

To specify the same logon account for all service accounts in this instance of SQL Server, provide credentials in the fields at the bottom of the page.

**Security Note** Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

15. Use the **Server Configuration - Collation** tab to specify non-default collations for the Database Engine and Analysis Services. For more

information, see <u>SQL Server Configuration - Collation</u>.

- 16. On the Full-Text Search Upgrade Options page, specify the upgrade options for the databases being upgraded. For more information, see <u>Full-Text Search Upgrade Options</u>.
- 17. On the **Error and Usage Reporting** page, specify the information you would like to send to Microsoft that will help to improve SQL Server. By default, options for error reporting and feature usage are enabled. For more information, see <u>Error and Usage Report Settings</u>.
- 18. The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified before the upgrade operation begins.
- 19. On the Cluster Upgrade Report page, specify the failover behavior while the node is offline during the upgrade operation. You can choose to failover to a node that has already been upgraded, or failover to a node that is at the previous version level. Use the Failover Behavior drop-down box to make your selection.
- 20. The Ready to Upgrade page displays a tree view of installation options that were specified during Setup. To continue, click **Install**.
- 21. During installation, the progress page provides status so you can monitor installation progress as Setup proceeds.
- 22. After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.
- 23. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For information about Setup log files, see <u>How to: View SQL</u> <u>Server 2008 Setup Log Files</u>.
# <sup>□</sup> Next Steps

After upgrading to SQL Server 2008, complete the following tasks:

- **Register your servers** Upgrade removes registry settings for the previous SQL Server instance. After upgrading, you must reregister your servers.
- **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 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 selectively installs and activates key services and features. For more information about surface area configuration, see the readme file for this release.

### $\Box$ See Also

#### Concepts

How to: Install SQL Server 2008 from the Command Prompt How to: View SQL Server 2008 Setup Log Files

#### **Other Resources**

How to: Verify a Successful Installation of SQL Server 2005 Services

### SQL Server Setup Help How to: Rename a SQL Server Failover Cluster Instance

# 

#### - -

<u>High Availability Solutions Overview</u> > <u>Getting Started with SQL Server 2008 Failover Clustering</u> > <u>Failover Clustering How-to Topics</u> >

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 more information, see <u>How to: Rename a Computer that Hosts a Stand-Alone Instance of SQL Server</u>.

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.

#### <sup>□</sup> 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.

# <sup>□</sup> 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:

#### <sup>©</sup> To minimize network propagation delay

1. Issue the following commands from a command prompt on the server node:

Copy Code

ipconfig /flushdns ipconfig /registerdns nbtstat -RR

### □ See Also

### Tasks

How to: Rename a Computer that Hosts a Stand-Alone Instance of SQL Server

### SQL Server Setup Help How to: Add Dependencies to a SQL Server Resource

# 

#### - -

High Availability Solutions Overview > Getting Started with SQL Server 2008 Failover Clustering > Failover Clustering How-to Topics >

If you install SQL Server 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 note 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 Setup, but SQL Server 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. Microsoft recommends that you put MS DTC in its own group with its own physical disk resource, if it is possible.

### <sup>□</sup> To add a dependency to a SQL Server 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 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.

### $\Box$ See Also

### Tasks

How to: Read a SQL Server 2008 Setup Log File

### Concepts

Before Installing Failover Clustering How to: View SQL Server 2008 Setup Log Files

SQL Server Setup Help SQL Server Preparation How-to Topics

- -

<u>Planning a SQL Server Installation</u> >

This section contains information on how to prepare SQL Server.

# <sup>□</sup> In This Section

• How to: Disable Server Message Block

### $\Box$ See Also

### Reference

Collation and Unicode Support Setting Up Windows Service Accounts

SQL Server Setup Help How to: Disable Server Message Block

# <mark>. See Also</mark>

#### - -

<u>Planning a SQL Server Installation</u> > <u>SQL Server Preparation How-to Topics</u> >

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.

### <sup>□</sup> 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.

#### <sup>□</sup> 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.

### $\Box$ See Also

#### Concepts

Security Considerations for a SQL Server Installation

#### **Other Resources**

SQL Server Preparation How-to Topics

SQL Server Setup Help

How to: Change Operating System Settings to Support Localized Versions



#### - -

<u>Planning a SQL Server Installation</u> > <u>SQL Server Preparation How-to Topics</u> >

Localized versions of SQL Server 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 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.

### <sup>□</sup> 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.

- 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, so it must match your localized version of SQL Server.

4. Click **Apply** to confirm the change, and **OK** to close the window.

#### <sup>□</sup> 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.
- 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.

#### <sup>©</sup> To change the system locale setting

- 1. If not already installed, install the operating system MUI that matches your localized version of SQL Server.
- 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 installation.

4. Click **Apply** to confirm the change, and **OK** to close the window.

### □ See Also

#### Reference

Collation and Unicode Support Hardware and Software Requirements for Installing SQL Server 2008

#### **Other Resources**

SQL Server Preparation How-to Topics

SQL Server Setup Help Installing SQL Server 2008



#### - -

The SQL Server Installation Wizard is Windows Installer-based. It provides a single feature tree to install all SQL Server components:

- SQL Server Database Engine
- Analysis Services
- Reporting Services
- Integration Services
- Replication
- Management tools
- Documentation

**Note** SQL Server is available in 32-bit and 64-bit editions. The 64-bit and 32bit editions of SQL Server are installed either through the Installation Wizard, or at a command prompt. For more information about SQL Server components, see <u>Editions and Components of SQL Server 2008</u> and <u>Features Supported by the</u> <u>Editions of SQL Server 2008</u>.

By default, sample databases and sample code are not installed as part of SQL Server Setup. To install sample databases and sample code for non-Express editions of SQL Server 2008, see the CodePlex Web site: <u>http://go.microsoft.com/fwlink/?LinkId=87843</u>. To read about support for SQL Server sample databases and sample code for SQL Server Express, see Databases and Samples Overview: <u>http://go.microsoft.com/fwlink/?</u> <u>LinkId=110391</u>.

Regardless of whether you use the SQL Server Installation Wizard or the command prompt to install SQL Server, the Setup process involves the following steps:

# <sup>□</sup> In This Section

#### Planning a SQL Server Installation

Desribes how to prepare your computer for SQL Server:

- Review hardware and software requirements.
- Review System Configuration Checker requirements and blocking issues.
- Review security considerations for a SQL Server installation.

#### Configuring SQL Server 2008

After Setup finishes, you can configure SQL Server by using graphical and command prompt utilities.

## <sup>□</sup> Related Sections

Considerations for Installing the SQL Server Database Engine

Describes how to install and configure the SQL Server Database Engine.

**Considerations for Installing Analysis Services** 

Describes how to install and configure Analysis Services.

Considerations for Installing Reporting Services

Describes how to install and configure Reporting Services.

**Considerations for Installing Integration Services** 

Describes how to install and configure Integration Services.

Considerations for Installing SQL Server Replication

This section describes how to install and configure SQL Server Replication.

Considerations for Installing SQL Server Management Tools

This section describes how to install and configure SQL Server management tools.

Considerations for Installing SQL Server Samples and Sample Databases

This section describes how to install SQL Server AdventureWorks samples.

### $\Box$ See Also

### Tasks

How to: Install SQL Server 2008 (Setup)

### **Other Resources**

High Availability Solutions Overview

### SQL Server Setup Help Considerations for Installing the SQL Server Database Engine

# 

#### - -

#### Installing SQL Server 2008 >

The Database Engine component of SQL Server 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.

SQL Server supports up to 50 instances of the Database Engine on a single computer. To create a typical SQL Server installation, see <u>How to: Install SQL Server 2008 (Setup)</u>.

**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.

The following features are installed when you select **SQL Server** Database Engine on the Components to Install page of the SQL Server Installation Wizard:

- Database Engine
- Replication is an optional component
- Full-Text Search is an optional component

The following additional features are options for many typical user scenarios:

- Integration Services
- Connectivity components
- Programming models
- Management tools
- Management Studio
- SQL Server Books Online

#### <sup> </</sup>Note:

By default, sample databases and sample code are not installed as part of

SQL Server Setup. To install sample databases and sample code, see the <u>CodePlex Web site</u>.

### □ See Also

#### Tasks

How to: Upgrade to SQL Server 2008 (Setup)

#### Reference

Editions and Components of SQL Server 2008

#### Concepts

Planning a SQL Server Installation

#### **Other Resources**

High Availability Solutions Overview

### SQL Server Setup Help Considerations for Installing Analysis Services



#### - -

Installing SQL Server 2008 >

SQL Server provides a single Setup program to install any or all of its components, including Analysis Services. Through Setup, you can install Analysis Services with or without other SQL Server components on a single computer.

This topic highlights important considerations that you should be aware of before you install Analysis Services. Information in this topic will help you evaluate the installation options so that you can make selections that result in a successful installation.

This topic does not include instructions for starting Setup, running Setup from the command line, or using the Installation Wizard. For step-by-step instructions on how to start Setup and select components to install, see <u>Quick-Start</u> <u>Installation of SQL Server 2008</u>. For information about command-line options for installing Analysis Services, see <u>How to: Install SQL Server 2008 from the Command Prompt</u>.

## <sup>©</sup> Preparing to Install SQL Server Analysis Services

Before you install Microsoft SQL Server Analysis Services, review the following requirements:

- Hardware and Software Requirements for Installing SQL Server 2008
- <u>Check Parameters for the System Configuration Checker</u>
- <u>Security Considerations for a SQL Server Installation</u>

# <sup>□</sup> Selecting Components to Install

While you can install Analysis Services alone, note that many data warehouse solutions require the installation of additional SQL Server components to enable the development, deployment, and administration of Analysis Services databases. The following table describes these additional SQL Server 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.	Х	X	
Management Tools Interactive tools for managing SQL Server, including SQL Server Configuration Manager, SQL Server Management Studio, SQL Server Profiler, and Replication Monitor.	Х		
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.		Х	
SQL Server Database Services The SQL Server Database Engine and tools for managing relational and XML data, replication, and full-text search.			X

Reporting Services			
A report server for managing, executing,			
rendering, and distributing reports, and the			Х
Report Manager, a Web-based user			
interface that provides access to SQL			
Server Reporting Services components,			
including administrative tools and reports.			
Integration Services			
SQL Server Integration Services, a set of			Х
graphical and programmable objects for			
moving, copying, and transforming data.			
Documentation and Samples			
SQL Server Books Online	v	v	$\mathbf{v}$
documentation, programming reference	λ	Х	Х
material, and samples for all SQL Server			
components.			

# <sup>©</sup> Options for Installing Analysis Services

You can install Analysis Services in the following configurations:

- Install Analysis Services on a computer with no previous instances of SQL Server.
- Install SQL Server 2008 Analysis Services side-by-side with an existing instance of SQL Server 2005 Analysis Services or of SQL Server 2000 Analysis Services.
- Upgrade an existing instance of SQL Server 2005 Analysis Services or of SQL Server 2000 Analysis Services to SQL Server 2008 Analysis Services.

# <sup>□</sup> Installing SQL Server Analysis Services

After you review the installation requirements for SQL Server and verify that your computer meets those requirements, you are ready to install SQL Server Analysis Services.

If you are using the Installation Wizard to install Analysis Services, you will use a series of pages to specify the components and options. The following table lists only those pages in the Installation Wizard where the options that you select affect your installation of Analysis Services:

Page	Recommendations
	Select <b>Analysis Services</b> to install the Analysis Services
Components	service.
to Install	Select Workstation components, Books Online and
	development tools to install tools and documentation.
Instance Name	You can specify only one instance name on the <b>Instance Name</b> page for all the SQL Server 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.
	When you do a side-by-side installation with SOL Server 2000
	Analysis Services, you must install SQL Server 2008 Analysis Services as a named instance.
	When you do a side-by-side installation with SQL Server 2005 Analysis Services, you must install SQL Server 2008 Analysis Services as a uniquely named instance
	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.

Service Account	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 <b>MSSQLServerOLAPService</b> . The service name for a named instance is <b>MSOLAP\$</b> <i>instancename</i> , where <i>instancename</i> is the name that you specified for the instance during setup.
Authentication Mode	This page sets the authentication provider for the Database Engine. SQL Server Analysis Services supports only Windows Authentication.
<b>Collation</b> <b>Settings</b>	<ul> <li>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.</li> <li>To use separate collation settings for Analysis Services and the SQL Server Database Engine</li> <li>1. On the Collation Settings page of the installation wizard, select the Customize for each service account check box.</li> <li>2. Next to Service, select Analysis Services, and then configure the collation options for Analysis Services.</li> <li><sup>I'</sup> Note: The SQL collations option does not apply for Analysis Services and is not available when you configure the Analysis Services collation options.</li> <li>3. Next to Service, select Database Engine and configure the collation options for the SQL Server Database Engine.</li> </ul>

Collation and Unicode Support.

### $\Box$ See Also

### Tasks

How to: Upgrade to SQL Server 2008 (Setup)

### **Other Resources**

High Availability Solutions Overview
# SQL Server Setup Help Analysis Services Installation How-to Topics



#### - -

Installing SQL Server 2008 > Considerations for Installing Analysis Services >

This section contains the following common procedures used for installing Microsoft SQL Server Analysis Services.

For common procedures used for upgrading from a previous version of Analysis Services and on migrating Analysis Services packages, see <u>Considerations for</u> <u>Installing Analysis Services</u>.

# <sup>Important:</sup>

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.

# <sup>□</sup> In This Section

- How to: Install Analysis Services by Using Setup
- 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

# Concepts

Considerations for Installing Analysis Services

# SQL Server Setup Help How to: Install Analysis Services by Using Setup

# 

#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Analysis Services</u> > <u>Analysis Services</u> <u>Installation How-to Topics</u> >

You can install Microsoft SQL Server Analysis Services 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 the initial setup of SQL Server. This includes feature selection, instance naming rules, service account configuration, strong password guidelines, and collation selection.

# <sup>I</sup>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 the installation of SQL Server.

# <sup>©</sup> To Install Analysis Services by Using Setup

- 1. Start the SQL Server Installation Wizard.
- 2. Optionally, review <u>How to: Install SQL Server 2008 (Setup)</u> for a 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 <u>Considerations for Installing Analysis</u> <u>Services</u> for more information about other SQL Server features, such as documentation and development tools that you might 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 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.

## Concepts

Considerations for Installing Analysis Services

### **Other Resources**

Analysis Services Installation How-to Topics

#### SQL Server Setup Help

How to: Install Multiple Instances of Analysis Services

# e Also

#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Analysis Services</u> > <u>Analysis Services</u> <u>Installation How-to Topics</u> >

You can run multiple instances of Microsoft SQL Server Analysis Services on the same computer. To install an additional instance, run Setup again and specify a unique name for the new instance.

#### <sup>I</sup>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.

## <sup>©</sup> To install an additional instance of Analysis Services

- 1. Run Setup either by using the SQL Server Installation Wizard or at the command prompt. For more information, see <u>Considerations for</u> <u>Installing Analysis Services</u>.
- 2. When you use the SQL Server Installation Wizard, specify a unique instance name on the **Instance Name** page. When you run Setup at the command prompt, specify a unique instance name by using the **INSTANCE** option.

# **Other Resources**

Analysis Services Installation How-to Topics

SQL Server Setup Help

How to: Install Analysis Services Side-by-Side with an Earlier Version



#### - -

<u>Installing SQL Server 2008 > Considerations for Installing Analysis Services > Analysis Services</u> <u>Installation How-to Topics</u> >

You can install SQL Server 2008 Analysis Services side-by-side with an existing instance of SQL Server 2005 Analysis Services or of SQL Server 2000 Analysis Services.

When you do a side-by-side installation with SQL Server 2000 Analysis Services, you must install SQL Server 2008 Analysis Services as 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 2008 Analysis Services the default instance by using the Analysis Services Instance Rename tool. For more information, see <u>How to: Rename an Instance of Analysis Services</u>.

When you do a side-by-side installation with SQL Server 2005 Analysis Services, you must install SQL Server 2008 Analysis Services as a uniquely named instance

## <sup>Important:</sup>

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.

## <sup>©</sup> To Install Analysis Services Side-by-Side with an Earlier Version

1. Run Setup either by using the SQL Server Installation Wizard or at the

command prompt. For more information, see <u>Considerations for</u> <u>Installing Analysis Services</u>.

2. When you use the SQL Server Installation Wizard, specify a unique instance name on the **Instance Name** page. When you run Setup at the command prompt, specify a unique instance name by using the **INSTANCE** option.

# **Other Resources**

Analysis Services Installation How-to Topics

#### SQL Server Setup Help

#### How to: Rename an Instance of Analysis Services



#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Analysis Services</u> > <u>Analysis Services</u> <u>Installation How-to Topics</u> >

You can rename an existing instance of Microsoft Analysis Services by using the **Rename Instance** dialog box.

# <sup>Important:</sup>

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.

## <sup>✓</sup>Note:

On a 64-bit computer, you must use the 64-bit version of the Instance Rename tool.

# <sup>☑</sup> Note:

The Analysis Services Instance Rename tool is not supported for use in a cluster environment.

# <sup>□</sup> To rename an instance of Analysis Services

1. Launch the **Instance Rename** tool, **asinstancerename.exe**, from C:\Program Files\Microsoft SQL Server\10\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**.

#### **Other Resources**

**Rename Instance Dialog Box (Analysis Services)** <u>Analysis Services Installation How-to Topics</u> **Managing Analysis Services Instances** 

SQL Server Setup Help

How to: Add or Remove Features for an Instance of Analysis Services



#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Analysis Services</u> > <u>Analysis Services</u> <u>Installation How-to Topics</u> >

In Microsoft SQL Server, feature maintenance is supported only through Add or Remove Programs in Control Panel in Microsoft Windows.

## <sup>Important:</sup>

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.

#### <sup>©</sup> To change components in an existing instance of SQL Server 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**, and then click **Change**. This will start the **Microsoft SQL Server Maintenance Wizard**.
- 3. On the **Component Selection** page, under SQL Server 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 Maintenance Wizard** then launches the

Microsoft SQL Server 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.

# **Other Resources**

Analysis Services Installation How-to Topics

# SQL Server Setup Help How to: Uninstall an Instance of Analysis Services



#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Analysis Services</u> > <u>Analysis Services</u> <u>Installation How-to Topics</u> >

Use the following steps to uninstall an instance of Microsoft SQL Server Analysis Services.

## <sup>I</sup> 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.

# <sup>□</sup> To remove an instance of SQL Server 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**, and then click **Change**. This will start the **Microsoft SQL Server Maintenance Wizard**.

#### <sup>I</sup>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 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 Maintenance Wizard** then launches the **Microsoft SQL Server 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.

# **Other Resources**

Analysis Services Installation How-to Topics

# SQL Server Setup Help Considerations for Installing Reporting Services



#### - ▼

Installing SQL Server 2008 >

SQL Server provides a single Setup program that installs all components, including Reporting Services. Using Setup, you can install Reporting Services with or without other SQL Server components on a single computer. You can also select specific features in Reporting Services if you want to install Reporting Services client and server applications separately.

This topic contains information and recommendations for:

- <u>Selecting client and server applications</u> to include in a Reporting Services installation.
- <u>Selecting additional services and features</u> to support report server operations.
- <u>Selecting a service account</u> for the Report Server service.
- <u>Selecting an installation mode</u> to have Setup minimally or completely configure the instance you are installing.
- <u>Installing multiple instances and versions</u> of Reporting Services on the same computer.

# <sup>©</sup> Preparing to Install Reporting Services

Before you install Reporting Services, review the following requirements and component descriptions:

- Hardware and Software Requirements for Installing SQL Server 2008
- <u>Security Considerations for a SQL Server Installation</u>
- <u>Setting Up Windows Service Accounts</u>

After you are familiar with the installation requirements, you can run Setup from the command line or the Installation Wizard:

- How to: Install SQL Server 2008 (Setup)
- How to: Install SQL Server 2008 from the Command Prompt

# <sup>□</sup> Selecting Client and Server Applications

In addition to the shared tools provided by SQL Server, Reporting Services includes a complete set of tools and applications for business reporting. The following list explains which tools and applications are installed when you select specific features in the Installation Wizard or command line arguments:

<b>Feature Selection</b>	Description
	Installs the following applications:
Reporting Services	Report Server service (which includes the Report Server
or	Web service, background processing application, and Report Manager) Report Builder Report Designer Model
/FEATURES=RS	Designer, Reporting Services Configuration tool, and the Reporting Services command line utilities (rsconfig.exe, rskeymgmt.exe and rs.exe).
Client Tools	
or	Installs SQL Server Management Studio used to configure the report server, manage jobs, manage shared schedules, and configure role definitions.
/FEATURES=Tools	
Business Intelligence Development	
Studio	Installs the Visual Studio design environment that provides
	Report Designer and Model Designer.
or	
/FEATURES=BIDS	

# <sup>□</sup> Selecting Additional Services and Features

A Reporting Services deployment can benefit from the addition of other SQL Server component technologies and services:

- Select **Database Engine Services** in the Feature Selection page to host the report server databases. Reporting Services requires a local or remote instance of SQL Server 2005 or SQL Server 2008 Database Engine to host the report server databases. If you install an instance of the Database Engine concurrently with Reporting Services, you can eliminate a post-Setup task by installing the default configuration, which creates and configures the report server databases automatically. In addition to hosting the report server databases, SQL Server relational databases can also be a source of data for reports and models.
- **SQL Server Agent** is required if you want to use the Reporting Services background processing application for subscriptions and scheduled reporting. SQL Server Agent should be running in the instance of the Database Engine that hosts the report server database. You can specify SQL Server Agent in the Server Configuration page when you install Database Engine.
- **SQL Server Browser** is required if you are installing an instance of the Database Engine that will be used to host a report server database for a report server instance that is on a different computer. You can specify SQL Server Browser in the Server Configuration page when you install Database Engine.
- Select **Analysis Services** in the Feature Selection page if you want to use SQL Server Analysis Services as a data source for reports and models.
- Select **SQL Server Books Online** in the Feature Selection page if you want to install the product documentation for Reporting Services.
- Select **Integration Services** in the Feature Selection page if you want to use the sample SSIS package, script, and reports to query report execution log data. You must have an installation of Integration Services to run the package.

# <sup>□</sup> Selecting a Service Account

Setup provides a Server Configuration page in the Installation Wizard so that you can configure the services that are part of the current installation. The installation does not select a default service account, so you must explicitly specify the service account that you want to use. It is recommended that you use a least-privilege domain user account with network connection permissions. If possible, specify an account that is used exclusively by the report server so that you can audit login activity for this account.

Later, if you want to modify the Report Server service account or update the password, use the Reporting Services Configuration tool to make your changes. For more information about account recommendations or updating service account settings, see <u>Setting Up Windows Service Accounts</u> and <u>Configuring the Report Server Service Account</u>.

# <sup>□</sup> Selecting an Installation Option

Setup requires that you select an installation option for Reporting Services:

- **Install the native mode default configuration** instructs Setup to install Reporting Services preconfigured to use default values. 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 or else you cannot select this installation option. This option creates a fully deployed report server in native mode that is ready to use when setup is finished, saving you time and steps. For more information, see <u>Default</u> <u>Configuration for a Native Mode Installation (Reporting Services)</u>.
- **Install the SharePoint mode default configuration** instructs Setup to install Reporting Services preconfigured to use a report server database that works with SharePoint data storage. This default installation option is available only if you are installing a local Database Engine instance and a Reporting Services instance at the same time. You should only select this option if you plan to configure report server integration with a SharePoint deployment. The report server will not be fully functional until you do so. For more information, see <u>Default Configuration for SharePoint Integrated Mode (Reporting Services</u>).
- Install but do not configure the server instructs Setup to copy the program files to disk, configure the Report Server service, and register the report server Windows Management Instrumentation (WMI) provider. This installation option is referred to as a "Files-only" installation and it is the default installation option. Further configuration using the Reporting Services Configuration tool is necessary before the report server can be used. Select this option if you want to configure the report server separately as a post-Setup task, use custom settings, use an existing or a remote Database Engine instance to host the report server database, or install Reporting Services as a named instance. For more information, see <u>Files-Only Installation (Reporting Services)</u>.

Setup does not install components on remote computers. Furthermore, deployment scenarios such as scale-out deployment, SharePoint integrated mode, or configuring a report server database on a failover cluster are beyond the scope of the Setup program. For more information about how these deployment scenarios are supported and configured, see **Planning and Architecture (Reporting Services)** in SQL Server Books Online.

## **Comparing Installation Options**

The following table compares the settings used in a default configuration with those used in a files-only installation.

Operation	Default Configuration	Files-Only Installation
Configures instance name.	Requires that you install Database Engine and Reporting Services at the same time as the same instance. Uses the SQL Server instance name that you specify for the Database Engine instance that hosts the report server databases.	Uses the SQL Server instance name. A files-only installation can be either the default instance or a named instance.
Creates program folders and copies files.	Yes	Yes
Registers the Report Server WMI provider.	Yes	Yes
Configures the service account and startup type.	Yes. This value must be specified during Setup.	Yes. This value must be specified during Setup.
Starts the Report Server service.	Yes	Yes
Grants folder permissions to the Report Server service account.	Yes	Yes
Creates the <b>reportserver</b> and <b>reportservertempdb</b> databases. Creates the <b>rsexecrole</b> that	Yes. If the default configuration is for SharePoint integrated mode, the report server system property <b>IsSharePointEnabled</b> is set to <b>True</b> .	No

specifies database permissions.		
Configures a database connection.	Yes. The Report Server service account is granted rights to the database through the <b>rsexecrole</b> .	No
Reserves URLs for report server and Report Manager (native mode only).	Yes	No
Initializes the report server.	Yes	No
Configures e-mail delivery.	No	No
Configures unattended execution account.	No	No
Backs up encryption keys.	No	No

# **Configuring Reporting Services**

After you install Reporting Services, you can modify the installation to reconfigure the server or select custom settings for your installation. 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.

To run the Reporting Services Configuration tool, click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2008**, point to **Configuration Tools**, and then click **Reporting Services Configuration**. For more information, see **Reporting Services Configuration Tool** and <u>Reporting Services Configuration</u> <u>How-to Topics</u>.

## **Verifying Your Reporting Services Installation**

Following installation and configuration, verify that Reporting Services components are running. For more information, see <u>How to: Verify a Reporting Services Installation</u>.

# Installing Multiple Instances and Versions of Reporting Services

The report server component is instance-aware. You can install multiple instances of the same version, or different versions of Reporting Services if you want to run the SQL Server 2008, SQL Server 2005, or SQL Server 2000 versions of Reporting Services side-by-side on the same computer.

SQL Server 2000 Reporting Services can continue to run as a default instance, side-by-side with named instances of SQL Server 2008 Reporting Services. If you run SQL Server 2000 Reporting Services alongside other versions, it must be a default instance because SQL Server 2000 Reporting Services does not support named instances. Therefore, when doing a side-by-side installation on a computer that is running SQL Server 2000 Reporting Services, SQL Server Setup allows only the installation of named instances of SQL Server 2008 Reporting Services.

Setup installs one instance at a time. A single instance of SQL Server can contain multiple components, including Reporting Services. You can install multiple instances of Reporting Services on the same computer by running Setup multiple times.

If you install multiple instances of Reporting Services that run as the default instance and as named instances, each instance will be identified through the instance name. SQL Server 2008 also includes version information. For example, if you installed a default and named instance of SQL Server 2005 and SQL Server 2008, you would see program files and services for all four instances:

- SQL Server 2005 Reporting Services (MSSQLSERVER)
- SQL Server 2005 Reporting Services (MyInstanceName)
- SQL Server 2008 Reporting Services (MSRS10.MSSQLSERVER)
- SQL Server 2008 Reporting Services (MSRS10.MyInstanceName)

By default, if you install Reporting Services as a named instance, the instance name will be used to define the virtual directory for the report server and Report Manager, and it will be included in the URL for a report server and for Report Manager so that each application is uniquely named relative to other instances of the application that run on the same computer. For more information, see <u>Configuring Report Server URLs</u>.

## **Shared Features**

You cannot install multiple instances of the shared features, but you can run them side-by-side earlier versions (for example, running SQL Server 2005 and SQL Server 2008 Business Intelligence Development Studio on the same computer).

You cannot use the report authoring, server configuration, or management tools of one version with another version. For example, you cannot use the Reporting Services Configuration tool from SQL Server 2008 to configure an earlier version of a report server. Be sure to install all the tools and applications you need for each version you install.

# <sup>□</sup> See Also

#### Concepts

Configuring Reporting Services for Scale-Out Deployment

#### **Other Resources**

Deployment (Reporting Services) Upgrade (Reporting Services) Reporting Services Components and Tools Planning and Architecture (Reporting Services)

SQL Server Setup Help

Default Configuration for a Native Mode Installation (Reporting Services)



#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Reporting Services</u> >

*Default configuration for native mode* refers to a Reporting Services installation where Setup both installs and configures a report server instance. After Setup is finished, the report server is running and ready to use. A native mode report server runs as a stand-alone application server. Native mode is the default mode.

Setup installs the following Reporting Services features when you select the default configuration for native mode option: Report Server service (which includes the Report Server Web service, background processing application, and Report Manager), Report Builder, Report Designer Preview, the Reporting Services Configuration tool, and the Reporting Services command line utilities (rsconfig.exe, rskeymgmt.exe and rs.exe). This option does not apply to shared features such as SQL Server Management Studio or Business Intelligence Development Studio, which must be specified as separate items if you want to install them.

Setup configures the following for a native mode report server installation:

- Service account for the Report Server service.
- Report Server Web service URL.
- Report Manager URL.
- Report Server database.
- Service account access to the report server databases.
- DSN connection for the report server databases.

Setup does not configure the unattended execution account, report server e-mail, back up the encryption keys, or a scale-out deployment. You can use the Reporting Services Configuration tool to configure these properties. For more information, see **Reporting Services Configuration Tool**.

# <sup>□</sup> When to Install the Default Configuration for Native Mode

A default configuration installs Reporting Services in an operational state so that you can use the report server immediately after Setup is finished. Specify this mode when you want to save steps by eliminating any required configuration tasks you would otherwise have to perform in the Reporting Services Configuration tool.

Installing the default configuration does not guarantee that the report server will work after Setup is finished. The default URLs might not register when the service starts. Always test your installation to verify that the service starts and runs as expected.

# <sup>©</sup> Requirements

This installation option uses default values to configure the core settings required to make a report server operational. It has the following requirements:

- Reporting Services and SQL Server Database Engine must be installed together in the same instance. The Database Engine instance hosts the report server database that Setup creates and configures.
- The user account used to run Setup must be a member of the local Administrators group and have permission to access and create databases on the Database Engine instance that hosts the report server databases.
- Setup must be able to use the default values to reserve the URLs that provide access to the report server and Report Manager. These values are port 80, a strong wildcard, and the virtual directory names in the format ReportServer\_<instance\_name> and Reports\_<instance\_name>.
- Setup must be able to use the default values to create the report server databases. These values are <u>ReportServer</u> and <u>ReportServerTempDB</u>. If you have existing databases from a previous installation, Setup will be blocked because it cannot configure the report server in the default configuration for native mode. You must rename, move, or delete the databases to unblock Setup.

If your computer does not meet all requirements for a default installation, you must install Reporting Services in files-only mode and then use the Reporting Services Configuration tool to configure it after Setup is finished.

Do not try to reconfigure your computer only to allow a default installation to continue. Doing so could require several hours of work, effectively eliminating the time-saving benefit the installation option provides. The best solution is to install Reporting Services in files-only mode and then configure the report server to use specific values.

# <sup>□</sup> Default URL Reservations

URL reservations are composed of a prefix, host name, port, and virtual directory:

Part	Description
Prefix	The default prefix is HTTP. If you previously installed a Secure
	Sockets Layer (SSL) certificate, Setup will try to create URL
	reservations that use the HTTPS prefix.
Host name	The default host name is a strong wildcard (+). It specifies that the
	report server will accept any HTTP request on the designated port for
	any host name that resolves to the computer, including
	http:// <computername>/reportserver, http://localhost/reportserver, or</computername>
	http:// <ipaddress>/reportserver.</ipaddress>
Port	The default port is 80. If you install Reporting Services on 32-bit
	Windows XP SP2, the default port is 8080. Note that if you use any
	port other than port 80, you will have to explicitly add it to the URL
	when you open a Reporting Services Web application in a browser
	window.
Virtual directory	By default, virtual directories are created in the format of
	ReportServer_< <i>instance_name</i> > for the Report Server Web service and
	Reports_ <i><instance_name></instance_name></i> for Report Manager. For the Report Server
	Web service, the default virtual directory is <u>reportserver</u> . For Report
	Manager, the default virtual directory is <u>reports</u> .

An example of the complete URL string might be as follows:

- http://+:80/reportserver, provides access to the report server.
- http://+:80/reports, provides access to Report Manager.

# How to Install a Report Server in the Default Configuration for Native Mode

You can specify a native mode default configuration through the command line or in the Installation wizard. The following topics provide step-by-step instructions:

- How to: Install SQL Server 2008 (Setup).
- How to: Install SQL Server 2008 from the Command Prompt.

#### **Example Command Line Script**

The following example includes the Database Engine Service because it is required for a default configuration.

setup /q /ACTION=install /FEATURES=SQL,RS,TOOLS /INSTANCENAME=MSSQLS

#### **Installation Wizard**

When you select **Reporting Services** in the Feature Selection page, Setup provides a Reporting Services Configuration page that allows you to specify the installation mode. To specify a default native mode installation, select **Install the Native mode default configuration** on the Reporting Services Configuration page.
# □ See Also

#### Tasks

How to: Verify a Reporting Services Installation

#### Reference

Setting Up Windows Service Accounts

## Concepts

Configuring the Report Server Service Account Configuring Report Server URLs Configuring a Report Server Database Connection Files-Only Installation (Reporting Services) Default Configuration for SharePoint Integrated Mode (Reporting Services) Considerations for Installing Reporting Services Initializing a Report Server Configuring Report Server URLs Considerations for Installing Reporting Services Quick-Start Installation of SQL Server 2008

#### **Other Resources**

**Configuring a Report Server for Secure Sockets Layer (SSL) Connections Planning a Deployment Mode** 

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

Default Configuration for SharePoint Integrated Mode (Reporting Services)



#### -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Reporting Services</u> >

*Default configuration for SharePoint integrated mode* refers to a Reporting Services installation where Setup both installs and configures a report server instance, and creates the report server database in a format that supports content storage and addressing from a SharePoint site.

Setup installs the following Reporting Services features when you select the default configuration for SharePoint integrated mode option: Report Server service, Report Builder, Report Designer Preview, the Reporting Services Configuration tool, and the Reporting Services command line utilities (rsconfig.exe, rskeymgmt.exe and rs.exe). This option does not apply to shared features such as SQL Server Management Studio or Business Intelligence Development Studio, which must be specified as separate items if you want to install them.

When you select a default configuration option in Setup, Setup will configure the report server in either native mode or SharePoint integrated mode. For a SharePoint integrated mode report server, Setup configures the following:

- Service account for the Report Server service.
- Report Server Web service URL.
- Report Manager URL.
- Report Server database.
- Service account access to the report server databases.
- DSN connection to the report server databases.

Setup does not configure the unattended execution account, report server e-mail, back up the encryption keys, or a scale-out deployment. You can use the Reporting Services Configuration tool to configure these properties. For more information, see **Reporting Services Configuration Tool**.

After Setup is finished, the report server might not be fully operational, because it is not yet fully integrated with a SharePoint farm. You must still install a Reporting Services Add-in, configure integration settings, and possibly install additional prerequisite software if you are integrating the report server with a SharePoint farm that spans multiple servers. For more information, see <u>Configuring Reporting Services for SharePoint 3.0 Integration</u>.

# When to Install the Default Configuration for SharePoint Integrated Mode

This installation option configures the report server's core settings that enable the report server to be integrated with Windows SharePoint Services or Microsoft Office SharePoint Server . Specify this mode when:

- You want to save steps by eliminating any required configuration tasks you would otherwise have to perform in the Reporting Services Configuration tool. The default values for the Reporting Services URLs, database name, and virtual directories must be available to use. These values must be unique. If they are already used by another installation on the computer, Setup cannot use them to configure a default installation.
- You plan to immediately install the Reporting Services Add-in and configure report server integration settings on the SharePoint site. If the report server provides report processing capability to a SharePoint farm, you must also install a SharePoint Web front end on the report server computer so that you can join it to the farm. For more information, see <u>Configuring Reporting Services for SharePoint 3.0 Integration</u>.

# <sup>□</sup> Requirements

This installation option has the following requirements:

- Reporting Services and SQL Server Database Engine must be installed together in the same instance. The Database Engine instance hosts the report server database that Setup creates and configures.
- The user account used to run Setup must be a member of the Administrators group on the local computer and must have permission to access and create databases that use the Database Engine instance that hosts the report server databases.
- Setup must be able to use the default values to reserve the URLs that provide access to the report server and Report Manager. These values are port 80, a strong wildcard, and the virtual directory names in the format **ReportServer\_**<*instance\_name*> and **Reports\_**<*instance\_name*>.
- Setup must be able to use the default values to create the report server databases. For a default instance, these values are <u>ReportServer</u> and <u>ReportServerTempDB</u>. If you have existing databases from a previous installation, Setup will be blocked because it cannot configure the report server. You must rename, move, or delete the databases to unblock Setup.

Setup does not check for a SharePoint product or technology installation on the local machine, nor does it install the Reporting Services Add-in and configure integration settings in SharePoint Central Administration. You must install and configure these applications separately. For more information, see <u>Configuring Reporting Services for SharePoint 3.0 Integration</u>.

If your computer does not meet these requirements for a default installation, you must install Reporting Services in files-only mode and configure it manually after Setup is finished. You can use the Reporting Services Configuration tool to configure Reporting Services.

Do not try to reconfigure your computer only to allow a default installation to continue. Doing so could require several hours of work, effectively eliminating the time-saving benefit the installation option provides. The best solution is to install Reporting Services in files-only mode and then configure the report server to use specific values.

# <sup>□</sup> Default URL Reservations

URL reservations are composed of a prefix, host name, port, and virtual directory:

Part	Description
Prefix	The default prefix is HTTP. If you previously installed a Secure
	Sockets Layer (SSL) certificate, Setup will try to create URL
	reservations that use the HTTPS prefix.
Host name	The default host name is a strong wildcard (+). It specifies that the
	report server will accept any HTTP request on the designated port for
	any host name that resolves to the computer, including
	http:// <computername>/reportserver, http://localhost/reportserver, or</computername>
	http:// <ipaddress>/reportserver.</ipaddress>
	The default port is 80. If you install Reporting Services on 32-bit
Port	Windows XP SP2, the default port is 8080. Note that if you use any
	port other than port 80, you will have to explicitly add it to the URL
	when you open a Reporting Services Web application in a browser
	window.
Virtual directory	By default, virtual directories are created in the format of
	ReportServer_< <i>instance_name</i> > for the Report Server Web service and
	Reports_< <i>instance_name</i> > for Report Manager. For the Report Server
	Web service, the default virtual directory is <u>reportserver</u> . For Report
	Manager, the default virtual directory is <u>reports</u> .

An example of the complete URL string might be as follows:

Example URL	Description			
http://+:80/reportserverProvides access to the report server.				
	Provides access to Report Manager.			
http://+:80/reports	✓ <b>Note:</b> In SharePoint integrated mode, this URL will produce the rsOperationNotSupportedSharePointMode error message because Report Manager is not supported in this mode. In this mode, it is expected that all content			

management and report access is handled through SharePoint instead of Report Manager.

# How to Install a Report Server in the Default Configuration for SharePoint Integrated Mode

You can specify a SharePoint mode default configuration through the command line or in the Installation wizard. The following topics provide step-by-step instructions:

- How to: Install SQL Server 2008 (Setup).
- How to: Install SQL Server 2008 from the Command Prompt.

## **Example Command Line Script**

This example includes the Database Engine Service because it is required for a default configuration.

setup /q /ACTION=install /FEATURES=SQL,RS,TOOLS /INSTANCENAME=MSSQLS

## **Installation Wizard**

When you select Reporting Services in the Feature Selection page, Setup provides a Reporting Services Configuration page that allows you to specify the installation mode. To specify a SharePoint integrated mode installation, select **Install the SharePoint integrated mode default configuration** on the Reporting Services Configuration page.

# □ See Also

#### Tasks

How to: Verify a Reporting Services Installation

## Concepts

Configuring the Report Server Service Account Configuring Report Server URLs Configuring a Report Server Database Connection Files-Only Installation (Reporting Services) Default Configuration for a Native Mode Installation (Reporting Services) Considerations for Installing Reporting Services

#### **Other Resources**

Deployment (Reporting Services) Planning a Deployment Mode Planning for SharePoint Integration Requirements for Running Reporting Services in SharePoint Integrated Mode

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Files-Only Installation (Reporting Services)



#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Reporting Services</u> >

*Files-only installation* refers to a Reporting Services installation where Setup creates the folder structure for Reporting Services program files, copies the files to disk, registers the Report Server service on the local computer, configures the service account, grants files permissions to the service account, and registers the Reporting Services WMI provider.

A files-only installation includes the following Reporting Services features: Report Server service (which hosts the Report Server Web service, background processing application, and Report Manager), Report Builder, Report Designer Preview, the Reporting Services Configuration tool, and the Reporting Services command line utilities (rsconfig.exe, rskeymgmt.exe and rs.exe). It does not apply to shared features such as SQL Server Management Studio or Business Intelligence Development Studio, which must be specified as separate items if you want to install them.

In contrast with other installation modes, a report server that is installed in filesonly mode is not operational when Setup is finished. Additional configuration will be required to bring the report server online by using the **Reporting Services Configuration Tool**.

# <sup>B</sup> When to Select Files-Only Installation Mode

A files-only installation must be performed when:

- You want to connect the report server to a remote report server database.
- You want to install the report server as a named instance.
- You have deployment requirements that include using custom settings or functionality, and you want full control over when and how the server is configured.
- Installing a SQL Server failover cluster that includes Reporting Services.

# <sup>B</sup> How to Perform a Files-Only Installation

Files-only installation is the default for Reporting Services.

You can specify a files-only installation through the command line or in the Installation wizard. The following topics provide step-by-step instructions:

- How to: Install SQL Server 2008 (Setup).
- How to: Install SQL Server 2008 from the Command Prompt.

## **Example Command Line Script**

For clarity, the example includes /RSINSTALLMODE="FilesOnlyMode". However, because files-only mode is the default, you can omit this and still get a files-only mode installation.

setup /q /ACTION=install /FEATURES=RS /InstanceName=MSSQLSERVER /RSS

## **Installation Wizard**

When you select Reporting Services in the Feature Selection page, Setup provides a Reporting Services Configuration page that enables you to specify the installation mode. To specify a files-only installation, select **Install but do not configure the report server** on the Reporting Services Configuration page.

# □ See Also

#### Tasks

How to: Verify a Reporting Services Installation

## Concepts

Configuring the Report Server Service Account Configuring Report Server URLs Configuring a Report Server Database Connection Default Configuration for SharePoint Integrated Mode (Reporting Services) Default Configuration for a Native Mode Installation (Reporting Services) Considerations for Installing Reporting Services

#### **Other Resources**

Deployment (Reporting Services) Upgrade (Reporting Services) Reporting Services Components and Tools

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help How to: Verify a Reporting Services Installation



#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Reporting Services</u> >

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 about how to configure user access, see **Tutorial: Setting Permissions in Reporting Services** in SQL Server Books Online.

## <sup>☑</sup>Note:

The Reporting Services tutorials can help you learn important skills 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 the tools installed correctly. For more information, see **Tutorial: How to Locate and Start Reporting Services Tools** in SQL Server Books Online.

## <sup>©</sup> To verify that the report server is installed and running

- Run the Reporting Services Configuration tool and connect to the report server instance you just installed. The Web Service URL page includes a link to the Report Server Web service. Click the link to verify you can access the server. If the report server database is not configured, do that first before clicking the link.
- Open the Services console applications and verify that the Report Server service is running. To view the status of the Report Server 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**.

- 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 <u>ReportServer</u>. 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 <u>Configuring Report Server URLs</u>. If you are a local administrator on Windows Vista or Windows Server 2008, see <u>How to: Configure a Report Server for Local Administration on Windows Vista and Windows Server 2008</u>.
- 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 instructions on how to install sample databases, applications, and data files that are used in Reporting Services tutorials, see Initial Installation (Reporting Services) in SQL Server Books Online.

## <sup>□</sup> To verify that Report Manager is installed and running

- 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 or in the Report Manager URL page in the Reporting Services Configuration tool. By default, the Report Manager virtual directory is <u>Reports</u>. You can use the following URL to verify Report Manager installation: http://<*computer name*>/Reports<\_*instance name*>.
- 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.

# <sup>□</sup> To verify that Report Designer is installed and running

- 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 (Reporting Services)** in SQL Server Books Online.
- If you installed report samples, open the sample report project files and publish the reports to a report server. For more information, see **Publishing Models** in SQL Server Books Online.

# □ See Also

**Other Resources** 

Troubleshooting HTTP Errors Troubleshooting Configuration Errors Troubleshooting Server and Database Connection Problems Cause and Resolution of Reporting Services Errors

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help How to: Troubleshoot a Reporting Services Installation Problem



#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Reporting Services</u> >

If you cannot install Reporting Services because of errors that occur during Setup, use the instructions in this topic to address the conditions that are most likely to cause installation errors. For instructions on how to troubleshoot post-Setup errors that occur while configuring a report server for operation, see **Troubleshooting Configuration Errors**.

# <sup>□</sup> Check Setup Logs

Setup errors are recorded in log files in the Microsoft SQL Server\100\Setup Bootstrap\Log folder. A subfolder is created each time you run Setup. The subfolder name is the time and date you ran Setup. The log files include a collection of files. Open the \*\_summary.txt file to view product, component, and instance information. Open the \*\_errorlog.txt file to view error information generated during Setup. Open the \*\_RS\_\*\_ComponentUpdateSetup.log to view Reporting Services setup information. For instructions on how to view the Setup log files, see <u>How to: View SQL Server 2008 Setup Log Files</u>.

# <sup>□</sup> Check Prerequisites

Setup checks prerequisites automatically. However, if you are troubleshooting setup problems, it is helpful to know which requirements Setup is checking for.

- Account requirements for running Setup include membership in the local Administrators group. Setup must have permission to add files, registry settings, create local security groups, and set permissions. If you are installing a default configuration, Setup must have permission to create a report server database on the SQL Server instance on which you are installing.
- Operating System must support HTTP.SYS 1.1. This means you must be using Windows XP Service Pack 2 (SP2), Windows Server 2003, Windows Vista, or Windows Server 2008.
- HTTP service must be enabled and running.
- Distributed Transaction Coordinator (DTC) must be running if you are also installing SQL Server Agent service.
- Authz.dll must be present in the System32 folder.

Setup no longer checks for Internet Information Services (IIS) or ASP.NET. Reporting Services requires MDAC 2.0 and the Microsoft .NET Framework version 2.0; Setup will install these if they are not already installed.

# Troubleshoot Problems with the Default Configuration

Reporting Services supports three modes of installation: default configuration for native mode, default configuration for SharePoint integrated mode, and filesonly mode. Installing Reporting Services in the default configuration presents a larger set of requirements that must be met in order for Setup to continue. If you cannot select this installation option, verify that the system meets all the requirements described in Default Configuration for a Native Mode Installation (Reporting Services).

If you cannot resolve the requirements issue, perform a files-only install and use the Reporting Services Configuration tool to configure the server. An important advantage to using the tool is that you can configure each setting independently and isolate any errors that occur during configuration. For more information, see **Reporting Services Configuration Tool**.

# $\Box$ See Also

**Concepts** <u>Considerations for Installing Reporting Services</u>

**Other Resources Troubleshooting Concepts (Reporting Services)** 

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Considerations for Installing Integration Services



- -

Installing SQL Server 2008 >

SQL Server provides a single Setup program to install any or all of its components, including Integration Services. Through Setup, you can install Integration Services with or without other SQL Server components on a single computer.

This topic highlights important considerations that you should know before you install Integration Services. Information in this topic will help you evaluate the installation options so that you can make selections that result in a successful installation.

This topic does not include instructions for starting Setup, using the Setup Wizard, or running Setup from the command line. For step-by-step instructions on how to start Setup and select components to install, see <u>Quick-Start</u> <u>Installation of SQL Server 2008</u>. For information about command-line options for installing Integration Services, see <u>How to: Install SQL Server 2008 from the Command Prompt</u>.

# <sup>©</sup> Preparing to Install Integration Services

Before you install Microsoft SQL Server Integration Services, review the following requirements:

- <u>Hardware and Software Requirements for Installing SQL Server 2008</u>
- <u>Check Parameters for the System Configuration Checker</u>
- <u>Security Considerations for a SQL Server Installation</u>

# <sup>©</sup> Selecting an Integration Services Configuration

You can install SQL Server Integration Services in the following configurations:

- You can install SQL Server Integration Services on a computer that has no previous instances of SQL Server.
- You can install SQL Server 2008 Integration Services side-by-side with an existing instance of SQL Server 2005 Integration Services or of SQL Server 2000 Data Transformation Services (DTS) or of both.
- You can upgrade an existing instance of SQL Server 2005 Integration Services to SQL Server 2008 Integration Services. For more information, see **Considerations for Upgrading Integration Services**. For information about backward compatibility with earlier versions of Integration Services, see <u>Integration Services Backward Compatibility</u>
- You can upgrade an existing instance of SQL Server 2000 Data Transformation Services (DTS) to SQL Server 2008 Integration Services. For more information, see <u>Considerations for Upgrading Data</u> <u>Transformation Services</u>. For information about backward compatibility with DTS, see <u>Support for Data Transformation Services (DTS) in SQL</u> <u>Server 2008</u>.

# □ Installing Integration Services

After you review the installation requirements for SQL Server and ensure that your computer meets those requirements, you are ready to install Integration Services.

If you are using the Setup Wizard to install Integration Services, you will use a series of pages to specify components and options. The following table lists only those pages in the Setup Wizard where the options that you select affect your installation of Integration Services:

Page	Recommendations
	Select <b>Integration Services</b> to install the Integration Services
	service and to run packages outside the design environment.
	For a complete installation of Integration Services, together with
	the tools and documentation for developing and managing packages, select both <b>Integration Services</b> and the following
	Shared Features:
	• <b>Business Intelligence Development Studio</b> to install the tools for designing packages.
	<ul> <li>Management Tools - Complete to install SQL Server</li> </ul>
	Management Studio for managing packages.
	<ul> <li>SQL Server Books Online to Install documentation.</li> <li>Client Tools SDK to install managed assemblies for</li> </ul>
	Integration Services programming.
	To use a dedicated server for extraction, transformation, and loading (ETL) processes, we recommend that you install a local instance of the SQL Server Database Engine when you install Integration Services. Integration Services typically stores packages in an instance of the Database Engine and relies on
	SQL Server Agent for scheduling those packages. If the ETL server does not have an instance of the Database Engine, you will have to schedule or run packages from a server that does have an instance of the Database Engine. This means that the

Feature Selection	packages will not be running on the ETL server, but instead on the server from which they were started. As a result, the resources of the dedicated ETL server are not being used as intended. Furthermore, the resources of other servers may be strained by the running ETL processes. Many data warehousing solutions also require the installation of additional SQL Server components, such as the SQL Server Database Engine, SQL Server Analysis Services, and SQL Server Reporting Services.
	✓ Note: Some SQL Server components that you can select for installation on the Feature Selection page of the Setup 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 Services option installs the Integration Services components required for the SQL Server Import and Export Wizard. The Business Intelligence Development Tools (BIDS) option installs the Integration Services components required to design a package, but the Integration Services service is not installed 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 Feature Selection page.
	Any selection that you make on the <b>Instance Configuration</b> page does not affect Integration Services or the Integration Services service. You can only install one instance of the Integration Services service on a computer. You connect to the service by using the computer name.
Instance Configuration	By default, the Integration Services service is configured to manage packages that are stored in the <b>msdb</b> database in the instance of the Database Engine that is installed at the same time

	as Integration Services. If an instance of the Database Engine is
	not installed at the same time as Integration Services, the
	Integration Services service is configured to manage packages
	that are stored in the <b>msdb</b> database of the local, default instance
	of the Database Engine. To manage packages that are stored in a
	named instance or a remote instance of the Database Engine, or
	in multiple instances of the Database Engine, you have to
	modify the configuration file. For more information about how
	to modify this configuration file, see <b>Configuring the</b>
	Integration Services Service.
	Review the settings for the Integration Services service on the
	Service Accounts tab of the Server Configuration page.
Server Configuration	By default, the Integration Services service is registered to run under the built-in Network Service account, and the <b>Startup</b> <b>Type</b> is <b>Automatic</b> . You do not have to enter a password for the built-in Network Service account.

# <sup>D</sup> Understanding the Integration Services Service

Integration Services installs the Integration Services service, which enables you to do the following tasks in SQL Server Management Studio:

- Manage Integration Services packages and folders in a hierarchical view. Packages can be stored in an instance of the Database Engine or in the file system.
- Monitor the execution of Integration Services packages that are running on a computer.

## <sup>I</sup>Important:

The Integration Services service is not required to create, save, and run Integration Services packages.

The Integration Services service is installed when you select the **Integration Services** option on the **Feature Selection** page. When you accept the default settings on the **Server Configuration** page, the Integration Services service is enabled and its **Startup Type** is **Automatic**.

You can only install a single instance of the Integration Services service on a computer. The service is not specific to a particular instance of the Database Engine. You connect to the service by using the name of the computer on which the service is running.

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 the Integration Services Service**.

By default, the Integration Services service is configured to manage packages that are stored in the **msdb** database in the instance of the Database Engine that is installed at the same time as Integration Services. If an instance of the Database Engine is not installed at the same time as Integration Services, the Integration Services service is configured to manage packages that are stored in the **msdb** database of the local, default instance of the Database Engine. To manage packages that are stored in a named instance or a remote instance of the

Database Engine, or in multiple instances of the Database Engine, you have to modify the configuration file. For more information about how to modify this configuration file, see **Configuring the Integration Services Service**.

By default, in a new installation, Integration Services is configured not to log events that are related to the running of packages to the Application event log. This setting prevents too many event log entries when you use the Data Collector feature of SQL Server 2008. The events that are not logged are EventID 12288, "Package started," and EventID 12289, "Package finished successfully." To log these events to the Application event log, open the registry for editing. Then in the registry, locate the

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\100\SSIS node, and change the DWORD value of the LogPackageExecutionToEventLog setting from 0 to 1.

# □ Installing Integration Services on 64-bit Computers

## **Integration Services Features Installed on 64-bit Computers**

Setup installs various Integration Services features based on the setup options that you select:

- When you install SQL Server and select Integration Services for installation, Setup installs all available 64-bit Integration Services features and tools.
- If you require Integration Services design-time features, you must also install Business Intelligence Development Studio.
- If you require 32-bit versions of the Integration Services runtime and tools to run certain packages in 32-bit mode, you must also install Business Intelligence Development Studio.

64-bit features are installed under the **Program Files** directory, and 32-bit features are installed separately under the **Program Files (x86)** directory. (This behavior is not specific to Integration Services or to SQL Server.)

## <sup>I</sup>Important:

Business Intelligence Development Studio, the 32-bit development environment for Integration Services packages, is not supported on the Itanium 64-bit operating system and is not installed on Itanium servers.

For more information, see **64-bit Considerations for Integration Services**.

## **Support for Data Transformation Services on 64-bit Computers**

There is no 64-bit design-time or run-time support for DTS packages. On a 64bit computer, DTS packages, and Integration Services packages that run DTS packages, can run only in 32-bit mode. To install the 32-bit version of the **dtexec** utility (dtexec.exe), make sure to select Client Tools or Business Intelligence Development Studio during setup.

There is also no 32-bit design-time or run-time support for DTS packages on

Itanium-based operating systems. Therefore, you cannot create, view, modify, or run DTS packages on Itanium-based operating systems.

# Installing Support for SQL Server 2000 Data Transformation Services Packages

SQL Server 2008 does not install run-time support for DTS packages. You have to install this run-time support before you can run and manage DTS packages. For information about how to install support for DTS, see <u>How to: Install</u> <u>Support for Data Transformation Services Packages</u>.

For more detailed information about support for DTS in SQL Server 2008, see <u>Support for Data Transformation Services (DTS) in SQL Server 2008</u>.

# $\Box$ See Also

## **Other Resources**

**Configuring Integration Services in a Clustered Environment** 

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Integration Services Installation How-to Topics



#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Integration Services</u> >

This section contains the following common procedures used for installing Microsoft SQL Server Integration Services.

For common procedures used for upgrading from Data Transformation Services (DTS), see <u>Upgrading Data Transformation Services How-to Topics</u>. For common procedures used for migrating DTS packages, see <u>Migrating Data Transformation Services Packages How-to Topics</u>.

# <sup>I</sup> 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.

# <sup>□</sup> In This Section

• How to: Install Support for Data Transformation Services Packages
## $\Box$ See Also

## Concepts

Considerations for Installing Integration Services How to: Install SQL Server 2008 from the Command Prompt

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help

#### How to: Install Support for Data Transformation Services Packages

#### - -

<u>Installing SQL Server 2008</u> > <u>Considerations for Installing Integration Services</u> > <u>Integration Services</u> <u>Installation How-to Topics</u> >

Data Transformation Services (DTS) packages can continue to run alongside Integration Services packages on the same computer. For more information, see <u>Support for Data Transformation Services (DTS) in SQL Server 2008</u>.

To use DTS packages alongside Integration Services packages, you can install run-time and design-time support as described in the following sections.

## <sup>ℤ</sup>Note:

There is no design-time or run-time support for DTS packages on Itaniumbased operating systems. Therefore, you cannot create, view, modify, or run DTS packages on Itanium-based operating systems.

## Installing Support for DTS Packages

DTS support in SQL Server 2008 consists of multiple components, as described in the topic, <u>Support for Data Transformation Services (DTS) in SQL Server</u> <u>2008</u>. Although Setup can install some of these components, those components are not fully functional until you install DTS run-time support. To install Integration Services together with all the features that support for DTS requires, you have to install multiple items as described in the following procedure.

## $^{\Box}$ To install all the features that support DTS

- 1. During Setup, on the **Feature Selection** page, make the following selections:
  - 1. Select **Integration Services**. This option installs the ActiveX Script task and the DTS Package Migration Wizard.
  - 2. Select **Client Tools Backward Compatibility.** This option installs the Execute DTS 2000 Package task.
- 2. After Setup, install the DTS runtime as described in the sections, "Installing Run-time Support for DTS Packages on a 32-bit Computer" and "Installing Run-time Support for DTS Packages on a 64-bit Computer," later in this topic.
- 3. (Optional) After Setup, install the DTS designer as described in the section, "Installing Design-time Support for DTS Packages," later in this topic.

# Installing Run-time Support for DTS Packages on a 32-bit Computer

SQL Server 2008 does not install run-time support for DTS packages. Before you can run and manage DTS packages, you have to install this run-time support as described in the following procedure.

## <sup>□</sup> To install run-time support for Data Transformation Services packages on a 32-bit computer

• On the installation media that you used to install SQL Server 2008, locate and run the version of the SQLServer2005\_BC\*.msi Windows Installer package that is appropriate for the operating system.

—or—

• In an Internet browser, open the <u>Microsoft SQL Server 2008 Feature Pack</u> page, and then download and install the **Microsoft SQL Server 2005** Backward Compatibility Components.

# Installing Run-time Support for DTS Packages on a 64-bit Computer

There is no 64-bit run-time support for DTS packages. On a 64-bit computer, DTS packages, and Integration Services packages that run DTS packages, can run only in 32-bit mode. To run packages in 32-bit mode outside BI Development Studio on a 64-bit computer, you have to install the 32-bit version of the Integration Services runtime and command prompt utilities.

## <sup>□</sup> To install the 32-bit version of the Integration Services runtime and command prompt utilities on a 64-bit computer

• During Setup, select **Business Intelligence Development Studio** to install the 32-bit version of the Integration Services runtime, the **dtexec** utility (dtexec.exe), and other Integration Services tools.

## □ Installing Design-time Support for DTS Packages

SQL Server 2008 does not install design-time support for DTS packages. Before you can use the SQL Server 2008 tools to open and view DTS packages, you have to download and install this design-time support as described in the following procedure.

## <sup>I</sup>Note:

The DTS package designer that you can download works with the SQL Server 2008 tools, but the designer's features have not been updated for SQL Server 2008.

## <sup>□</sup> To install design-time support for Data Transformation Services packages

• In an Internet browser, open the Feature Pack for Microsoft SQL Server 2005 Web page, and then download and install the Microsoft SQL Server 2000 DTS Designer Components.

By default, the binary files for SQL Server 2000 are in the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn folder, and the binary files for SQL Server 2008 are in the %ProgramFiles%\Microsoft SQL Server\100\Tools\Binn folder. These default locations might cause the path of the binary files for SQL Server 2000 to occur in the PATH environment variable before the path of the binary files for SQL Server 2008. If this happens, an error message can appear when you try to use the DTS Designer. To resolve this error, copy the files that the designer requires to a new location as described in the following procedures.

#### <sup>□</sup> To ensure that the DTS designer can be used in SQL Server Management Studio

- Copy the files, SEMSFC.DLL, SQLGUI.DLL, and SQLSVC.DLL, from the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn folder to the %ProgramFiles%\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE folder.
- Copy the files, SEMSFC.RLL, SQLGUI.RLL, and SQLSVC.RLL, from the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn\Resources folder to the %ProgramFiles%\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\Resources folder.

## <sup>□</sup> To ensure that the DTS designer can be used in Business Intelligence Development Studio

- 1. Copy the files, SEMSFC.DLL, SQLGUI.DLL, and SQLSVC.DLL, from the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn folder to the %ProgramFiles%\Microsoft Visual Studio 9.0\Common7\IDE folder.
- Copy the files, SEMSFC.RLL, SQLGUI.RLL, and SQLSVC.RLL, from the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn\Resources folder to the %ProgramFiles%\Microsoft Visual Studio 9.0\Common7\IDE\Resources folder.

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Considerations for Installing SQL Server Replication

## 

#### -

Installing SQL Server 2008 >

Replication components can be installed by using the SQL Server Installation Wizard or at a command prompt. Install replication when you install SQL Server, or when you modify an existing instance.

After replication components are installed, you must configure the server before you can use replication. For more information, see **Designing and Implementing (Replication)** in SQL Server Books Online.

#### <sup>Important:</sup>

If you install replication components when you modify an existing instance of SQL Server, you must stop and restart SQL Server Agent after the installation is completed. This action helps ensure 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 Replication** on the **Feature Selection** page of the Installation Wizard.

## □ Installing Replication from the Command Prompt

To install replication when installing a new instance of SQL Server

• See <u>How to: Install SQL Server 2008 from the Command Prompt</u>.

## $\Box$ See Also

#### Concepts

Installing SQL Server 2008 How to: Install SQL Server 2008 from the Command Prompt Features Supported by the Editions of SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

Considerations for Installing SQL Server Management Tools

## See Also

#### - ▼

#### Installing SQL Server 2008 >

SQL Server Management tools include the following components:

- SQL Server Management Studio
- SQL Server Profiler
- SQL Server Database Tuning Advisor
- Command prompt tools, such as sqlcmd.exe and osql.exe.
- Business Intelligence Development Studio add-ins to Microsoft Visual Studio

Note that Business Intelligence Development Studio is a separate option during SQL Server installation.

Regardless of how many instances of SQL Server, Analysis Services, or Reporting Services are installed on a computer, only one copy of SQL Server 2008 Management Tools will be installed.

SQL Server 2008 Management Tools can run side-by-side on the same computer with earlier versions of SQL Server Management Tools.

## $\Box$ See Also

## Tasks

How to: Install SQL Server 2008 (Setup) How to: Upgrade to SQL Server 2008 (Setup)

## Reference

Editions and Components of SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Considerations for Installing SQL Server Samples and Sample Databases

See Also

- -

Installing SQL Server 2008 >

The code samples and sample databases for SQL Server are no longer included with the product. Instead, you can locate and download the samples and sample databases from the **Microsoft SQL Server Samples and Community Projects** Web site at <u>http://www.codeplex.com/sqlserversamples</u>. In the tables organized by technology, there are hyperlinks to samples for the SQL Server components and to the official SQL Server sample databases. These tables also provide links to samples and projects that are available in the SQL Server community.

## <sup>□</sup> Installing Samples and Sample Databases

To install the complete set of product samples, on the **Microsoft SQL Server Samples and Community Projects** home page, click the **Releases** tab. Note that the default release for each project is the most current release. This might be a community technology preview release. To select a prior release, click the release name in the **Release** pane.

On CodePlex, you can locate specific projects for the product samples for a technology area of SQL Server (Analysis Services, Database Engine, Data Programmability, and so on) by clicking the link for the technology you want in the technology area table of the Microsoft SQL Server Samples and Community Projects home page at <a href="http://www.codeplex.com/sqlserversamples">http://www.codeplex.com/sqlserversamples</a>. In each specific project, click the **Releases** tab, and then click the Windows Installer package to download the set of samples in that technology area. To view the code for samples in that technology area, click the **Source Code** tab, and then click the folder icon in the **Browse** column.

## <sup>I</sup>Note:

If you choose to install the samples, by default, the sample files are installed in the following location: C:\Program Files\Microsoft SQL Server\100\Samples\ <*technology\_name*>.

To install the official sample databases for SQL Server, go to <u>http://www.CodePlex.com/MSFTDBProdSamples</u>.Or, on the Microsoft SQL Server Samples and Community Projects home page, in the Sample Databases section, click the Download link and select the databases that you want.

## Providing Feedback

Http://connect.microsoft.com and CodePlex provide a direct way to report issues and ask questions about the samples for each technology area. To report problems and review the status of issues that you have reported, file a bug at <u>Http://connect.microsoft.com</u>. To discuss various sample ideas, provide feedback about existing samples, or ask questions about the samples, click the **Discussions** tab at Http://Connect.Microsoft.com/SqlServerSamples or one of the related projects. Please do not use the **Issues** tab on CodePlex to report samples defects.

## <sup>D</sup> Updating the Location for the .NET Framework CLR Samples

Before you compile Microsoft .NET Framework common language runtime (CLR) samples for SQL Server, verify that the path of the version of the .NET Framework compiler is the first Framework directory in the PATH environment variable. The location of the compiler that is shipped with this release of SQL Server is C:\<*Windows root directory*>\Microsoft.NET\Framework\v2.0.*x*. Drive C is the installation drive, <*Windows root directory*> is either Windows or WINNT, and *x* is five digits.

## □ See Also

## Concepts

System and Sample Databases

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Installation How-to Topics

Ξ 🖌

<u>Installing SQL Server 2008</u> >

Use the following topics to install SQL Server 2008.



## <sup>□</sup> In This Section

Торіс	Description
How to: Install SQL Server	Procedural topic for a typical SQL Server
<u>2008 (Setup)</u>	installation by using the Installation Wizard.
How to: Install SQL Server	Procedural topic that provides sample syntax and
2008 from the Command	installation parameters for running unattended
<u>Prompt</u>	Setup.
How to: Add Features to an	Procedural topic for updating an existing
Instance of SQL Server 2008	instance of SOL Server 2008
( <u>Setup</u> )	

## $\Box$ See Also

## Concepts

Planning a SQL Server Installation Upgrading to SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help How to: Install SQL Server 2008 (Setup)



#### - -

<u>Installing SQL Server 2008</u> > <u>Installation How-to Topics</u> >

The SQL Server Installation Wizard is Windows Installer-based. It provides a single feature tree for installation of all SQL Server components, so you do not have to install the following components individually:

- Database Engine
- Analysis Services
- Reporting Services
- Integration Services
- Replication
- Management tools
- Connectivity components
- Sample databases, samples, and SQL Server Books Online

These additional topics document other ways to install SQL Server 2008:

- How to: Install SQL Server 2008 from the Command Prompt.
- <u>How to: Create a New SQL Server Failover Cluster (Setup)</u>.
- <u>How to: Upgrade to SQL Server 2008 (Setup)</u>.

## <sup>©</sup> Prerequisites

Before you install SQL Server, review topics in <u>Planning a SQL Server</u> <u>Installation</u>.

#### <sup> </</sup> 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.

## <sup>□</sup> To install SQL Server 2008

- 1. Insert the SQL Server installation media. From the root folder, doubleclick setup.exe. To install from a network share, locate the root folder on the share, and then double-click setup.exe.
- 2. If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.
- 3. Windows Installer 4.5 is also required, and might be installed by the Installation Wizard. If you are prompted to restart your computer, restart it, and then restart SQL Server 2008 setup.exe.
- 4. When the prerequisites are installed, the Installation Wizard will run the SQL Server Installation Center. To create a new installation of SQL Server 2008, click **New Installation or Add Features to an Existing Installation**.
- 5. The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information, see <u>How to: View SQL</u>

Server 2008 Setup Log Files.

- 6. On the Product Key page, select a radio button to indicate whether you are installing a free edition of **SQL Server**, or a production version of the product that has a PID key. For more information, see <u>Editions and Components of SQL Server 2008</u>.
- 7. On the License Terms page, read the license agreement, and then select the check box to accept the licensing terms and conditions.
- 8. The Installation Wizard will install SQL Server prerequisites if they are not already on the computer. These include the following:
  - .NET Framework 2.0
  - SQL Server Native Client
  - SQL Server Setup Support Files

To install prerequisites, click Install.

- 9. The System Configuration Checker will verify the system state of your computer before Setup continues.
- 10. On the Feature Selection page, select the components for your installation. A description for each component group appears in the right-hand pane after you select the feature name. You can select any combination of check boxes. For more information, see Editions and Components of SQL Server 2008.

You can also specify a custom directory for shared components by using the field at the bottom of the Feature Selection page. To change the installation path for shared components, either update the path name in the field at the bottom of the dialog box, or click **Browse** to navigate to an installation directory. The default installation path is C:\Program Files\Microsoft SQL Server\100\.

11. On the Instance Configuration page, specify whether to install a default instance or a named instance. For more information, see <u>Instance</u> <u>Configuration</u>. To continue, click Next.

**Instance ID suffix** — By default, the instance name is used as the

Instance ID suffix. This is used to identify installation directories and registry keys for your instance of SQL Server. This is the case for default instances and named instances. For a default instance, the instance name and instance ID suffix would be MSSQLSERVER. To use a non-default instance ID suffix, select the **Instance ID suffix** check box and provide a value.

## <sup>I</sup>Note:

Typical stand-alone instances of SQL Server 2008, whether default or named instances, do not use a non-default value for the **Instance ID suffix** check box.

**Instance root directory** — By default, the instance root directory is C:\Program Files\Microsoft SQL Server\100\. To specify a non-default root directory, use the field provided, or click **Browse** to locate an installation folder.

All SQL Server service packs and upgrades will apply to every component of an instance of SQL Server.

**Detected instances and features** — The grid shows instances of SQL Server that are on the computer where Setup is running. If a default instance is already installed on the computer, you must install a named instance of SQL Server 2008.

- 12. The Disk Space Requirements page calculates the required disk space for the features you specify. It then compares the required space to the available disk space. For more information, see <u>Disk Cost Summary</u>.
- 13. Work flow for the remainder of this topic depends on the features you have specified for your installation. You might not see all of the pages, depending on your selections.
- 14. On the Server Configuration Service Accounts page, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features you selected to install.

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, are started manually, or are disabled. Microsoft recommends that you configure service accounts individually to provide least privileges for each service, where SQL Server services are granted the minimum permissions they need to complete their tasks. For more information, see <u>SQL Server</u> <u>Configuration - Service Accounts</u> and <u>Setting Up Windows Service Accounts</u>.

To specify the same logon account for all service accounts in this instance of SQL Server, provide credentials in the fields at the bottom of the page.

**Security Note** Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

- 15. Use the **Server Configuration Collation** tab to specify non-default collations for the Database Engine and Analysis Services. For more information, see <u>SQL Server Configuration Collation</u>.
- 16. Use the Database Engine Configuration Account Provisioning page to specify the following:
  - Security Mode select Windows Authentication or Mixed Mode Authentication for your instance of SQL Server. If you select Mixed Mode Authentication, you must provide a strong password for the built-in SQL Server system administrator account. After a device establishes a successful connection to SQL Server, the security mechanism is the same for both Windows Authentication and Mixed Mode. For more information, see <u>Database Engine Configuration - Account Provisioning</u>.
  - SQL Server Administrators You must specify at least one system administrator for the instance of SQL Server. To add the account under which SQL Server Setup is running, click Add Current User. To add or remove accounts from the list of system administrators, click Add or Remove, and then edit the list of users, groups, or computers that will have administrator privileges for the instance of SQL Server. For more information, see Database

#### Engine Configuration - Account Provisioning.

When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

17. Use the Database Engine Configuration - Data Directories page to specify non-default installation directories. To install to default directories, click **Next**.

## <sup>I</sup> Important:

If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.

For more information, see <u>Database Engine Configuration - Data</u> <u>Directories</u>.

- 18. Use the Database Engine Configuration FILESTREAM page to enable FILESTREAM for your instance of SQL Server. For more information, see <u>Database Engine Configuration Filestream</u>.
- 19. Use the Analysis Services Configuration Account Provisioning page to specify users or accounts that will have administrator permissions for Analysis Services. You must specify at least one system administrator for Analysis Services. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for Analysis Services. For more information, see <u>Analysis Services Configuration Account Provisioning</u>.

When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

20. Use the Analysis Services Configuration — Data Directories page to specify non-default installation directories. To install to default

directories, click Next.

## <sup>I</sup> Important:

If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.

For more information, see <u>Analysis Services Configuration - Data</u> <u>Directories</u>.

- 21. Use the Reporting Services Configuration page to specify the type of Reporting Services installation to create. Options include the following:
  - Native mode default configuration
  - SharePoint mode default configuration
  - Unconfigured Reporting Services installation

For more information about Reporting Services configuration modes, see <u>Report Server Installation Options</u>.

- 22. On the **Error and Usage Reporting** page, specify the information you would like to send to Microsoft that will help to improve SQL Server. By default, options for error reporting and feature usage are enabled. For more information, see <u>Error and Usage Report Settings</u>.
- 23. The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified.
- 24. The Ready to Install page shows a tree view of installation options that were specified during Setup. To continue, click **Install**.
- 25. During installation, the Installation Progress page provides status so you can monitor installation progress as Setup proceeds.
- 26. After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.

27. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For more information, see <u>How to: View SQL Server 2008 Setup Log Files</u>.

## <sup>□</sup> Next Steps

Configure your new SQL Server installation

- To reduce the attackable surface area of a system, SQL Server selectively installs and activates key services and features. For more information, see <u>Understanding Surface Area Configuration</u> on MSDN.
- For more information about using graphical tools and command prompt utilities, see <u>Configuring SQL Server 2008</u>.

## $\Box$ See Also

#### Tasks

How to: Upgrade to SQL Server 2008 (Setup)

#### Concepts

How to: View SQL Server 2008 Setup Log Files How to: Install SQL Server 2008 from the Command Prompt

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help How to: Install SQL Server 2008 from the Command Prompt

## 

#### - -

<u>Installing SQL Server 2008</u> > <u>Installation How-to Topics</u> >

Before you run SQL Server Setup, review <u>Planning a SQL Server Installation</u>.

Installing a new instance of SQL Server from the command prompt enables you to specify the features to install and how they should be configured. You can also specify silent, basic, or full interaction with the Setup user interface.

Command prompt installation is supported in the following scenarios:

- Installing, upgrading, or removing an instance and shared components of SQL Server 2008 on a local computer with syntax and parameters specified at the command prompt.
- Installing, upgrading, or removing a failover cluster instance.
- Upgrading from one SQL Server 2008 edition to another edition of SQL Server 2008.
- Installing an instance of SQL Server 2008 on a local computer with syntax and parameters specified in a configuration file. You can use this method to copy an installation configuration to multiple computers, or to install multiple nodes of a failover cluster installation.

When you install SQL Server from the command prompt, specify Setup parameters for your installation at the command prompt as part of your installation syntax.

## <sup>ℤ</sup>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 login as a service, and to act as part of the operating system on all failover cluster nodes.

## <sup>B</sup> Proper Use of Setup Parameters

Use the following guidelines to develop installation commands with correct syntax:

Copy Code

/PARAMETER
/PARAMETER=true/false

One exception is /FEATURES, which is a multi-value parameter, but its format is

• /FEATURES=AS,RS,IS with no space, comma-delimited

Note the use of escaping characters:

- /INSTANCEDIR=c:\PathName is supported.
- /INSTANCEDIR=c:\PathName\ is supported
- /INSTANCEDIR="c:\PathName\\" is supported.
- /INSTANCEDIR="c:\PathName\" is not supported.

## <sup>□</sup> Sample Syntax

Use the following sample scripts to install SQL Server components at the command prompt. The scripts provide command prompt syntax to install SQL Server in typical user scenarios. Test and modify samples to meet the needs of your organization.

Supported use of Setup parameters

Parameter Use

Basic installation scenarios, including Database Engine Services and Management Tools

Installation Scripts for Basic Installation Scenarios

Failover cluster installation scenarios

**Installation Scripts for Failover Cluster Scenarios** 

Analysis Services installation scenarios

#### **Installation Scripts for Analysis Services Installation Scenarios**

Reporting Services installation scenarios

Installation Options and Scripts for Reporting Services

Integration Services installation scenarios

Installation Scripts for Analysis Services Installation Scenarios

Reference documentation for installation parameters

#### **Installation parameters**

Compare Setup Parameters between SQL Server Versions

Parameters from previous SQL Server versions

## <sup>©</sup> Installation Scripts for Basic Installation Scenarios

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 components.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

Setup.exe /q /ACTION=Install /FEATURES=SQL,AS,RS,IS,Tools,BIDS,E **Important:** 

Do not use a blank password. Use a strong password.

#### <sup>I</sup>Note:

For a default instance, specify "MSSQLSERVER" for the */INSTANCENAME parameter*. For a named instance, specify the instance name.

For more information about service account configuration, see <u>Setting Up</u> <u>Windows Service Accounts</u>. For more information about log files, see <u>How</u> <u>to: View SQL Server 2008 Setup Log Files</u>.

To install a new, stand-alone instance with the SQL Server Database Engine, Replication, and Full-Text Search components.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

Setup.exe /q /ACTION=Install /FEATURES=SQL /INSTANCENAME=MSSQLSE

**Important** Do not use a blank password. Use a strong password.

## <sup>I</sup>Note:

For a default instance, specify "MSSQLSERVER" for the */INSTANCENAME parameter*. For a named instance, specify the instance name.

To install SQL Server Management Tools only.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

Copy Code

```
Setup.exe /Action=Install /FEATURES=Tools
```

To install SQL Server Express.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

Setup.exe /q /FEATURES=SQL\_Tools /INSTANCE=Express /SQLSVCACCOUN Use the /ENABLERANU parameter to enable RANU Use the /ADDCURRENTUSERASSQLADMIN parameter to provision the use

To upgrade an existing instance or failover cluster node from SQL Server 2000 or SQL Server 2005.

- 1. Before you begin, review the upgrade documentation in <u>Using Upgrade</u> <u>Advisor to Prepare for Upgrades</u> and <u>Backward Compatibility</u>. It is not necessary to specify the features to upgrade.
- 2. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 3. Run the following command at the command prompt:
Setup.exe /q /ACTION=upgrade /INSTANCENAME=MSSQLSERVER /SQLSYSAL

If upgrading a node of a failover cluster instance, use the /FAILOVERCLUSTERROLLOWNERSHIP parameter to control the failover behavior before the upgrade operation takes to node offline. Use of this parameter is as follows:

- /FAILOVERCLUSTERROLLOWNERSHIP=0 will not roll cluster ownership to upgraded nodes, and do not add this node to the list of possible owners of the SQL Server cluster at the end of upgrade.
- /FAILOVERCLUSTERROLLOWNERSHIP=1 will roll cluster ownership to upgraded nodes, and will add this node to the list of possible owners of the SQL Server cluster at the end of upgrade.
- /FAILOVERCLUSTERROLLOWNERSHIP=2 is the default setting. It will be used if this parameter is not specified. This setting indicates that SQL Server Setup will manage cluster ownership as needed.
- 4. To upgrade shared components only, run the following command at a command prompt:

Copy Code

Setup.exe /q /ACTION=upgrade

**Important** Do not use a blank password. Use a strong password.

#### <sup>I</sup>Note:

For a default instance, specify "MSSQLSERVER" for the */INSTANCENAME* parameter. For a named instance, specify the instance name.

To upgrade an existing instance of SQL Server 2008 to a different edition of SQL Server 2008.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

Setup.exe /q /ACTION=editionupgrade /INSTANCENAME=MSSQLSERVER /F

**Important** Do not use a blank password. Use a strong password.

#### <sup>I</sup>Note:

For a default instance, specify "MSSQLSERVER" for the */INSTANCENAME* parameter. For a named instance, specify the instance name. If the instance name is not specified, only the shared components will be upgraded.

To install SQL Server 2008 by using a configuration file.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

Setup.exe /q /ACTION=Install /Configurationfile=<path to Configu</pre>

After running SQL Server 2008 Setup, a configuration file is generated and saved by default to *<drive>*:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\Log\

<YYYYMMDD\_HHMMSS\ConfigurationFile.ini. Open the file to see descriptions of Setup parameters and settings for the installation that was completed. You can use the configuration file to reproduce additional instances or to install additional nodes of a failover cluster installation.

For more information about log files, see <u>How to: View SQL Server 2008</u> <u>Setup Log Files</u>.

To add components to an existing SQL Server instance.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

Setup.exe /q /ACTION=Install /INSTANCENAME=MSSQLSERVER /FEATURES

**Note** To add features to a named instance, use the instance name instead of

"MSSQLSERVER" in the example above.

To uninstall an existing SQL Server instance.

- 1. Insert the SQL Server installation media into the disk drive.
- 2. At a command prompt, navigate to the following directory: %Program Files%\Microsoft SQL Server\100\Setup Bootstrap and type the following command at the command prompt:

Setup100.exe /Action=Uninstall /FEATURES=SQL,AS,RS,IS,Tools /INS

**Note** To remove a named instance, use the instance name instead of "MSSQLSERVER" in the example above.

# □ Installation Scripts for Failover Cluster Installation Scenarios

There are two options for SQL Server Failover Cluster installation:

- Integrated installation installs and configures a single-node failover cluster instance in a single step.
- Advanced installation, also known as Enterprise installation, consists of 2 steps: preparing a failover cluster instance, and then completing the installation to create a single-node failover cluster instance.

Before you install a SQL Server failover cluster instance, review the following topics:

- <u>Hardware and Software Requirements for Installing SQL Server 2008</u>
- <u>Security Considerations for a SQL Server Installation</u>
- <u>Before Installing Failover Clustering</u>

#### <sup>I</sup> Important:

All failover cluster installation commands require an underlying Windows cluster. All the nodes that will be part of a SQL Server failover cluster must be part of the same Windows cluster.

Test and modify the following failover cluster installation scripts to meet the needs of your organization.

To install an integrated, single-node SQL Server failover cluster with the Database Engine.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at a command prompt:

setup.exe /q /ACTION=InstallFailoverCluster /FEATURES=SQL /INSTA

To install a single-node SQL Server failover cluster with Analysis Services.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

```
setup.exe /q /ACTION=InstallFailoverCluster /FEATURES=AS /INSTAM
```

To install a single-node SQL Server failover cluster instance with the Database Engine and Analysis Services, default instance.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=InstallFailoverCluster /InstanceName=MSSQLS

To install a single-node SQL Server failover cluster instance with the Database Engine and Analysis Services, named instance.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=InstallFailoverCluster /InstanceName=INSTAN

#### **Advanced Scenarios for Failover Cluster Installation:**

The following two examples use two steps to create a Database Engine failover cluster instance: one step to prepare the failover cluster instance, and another step to complete the process.

Test and modify the installation scripts to meet the needs of your organization.

To perform the "Preparation" step of a failover cluster advanced installation scenario for the Database Engine. This is step 1 of 2.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

```
Setup.exe /q /ACTION=PrepareFailoverCluster /FEATURES=SQL /INSTA
```

To perform the "Completion" step of a failover cluster advanced installation scenario for the Database Engine. This is step 2 of 2. Run the following command on the computer that will be the active node in the failover cluster to make it useable. You must run the "CompleteFailoverCluster" action on the node that owns the shared disk in the SQL Server failover cluster.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=CompleteFailoverCluster /INSTANCENAME="<Ins</pre>

The following two examples use two steps to create an Analysis Services failover cluster instance. Test and modify the installation scripts to meet the needs of your organization.

To perform the "Preparation" step of a failover cluster advanced installation scenario for Analysis Services. This is step 1 of 2.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=PrepareFailoverCluster /FEATURES=AS /Instar

To perform the "Completion" step of a failover cluster advanced installation scenario for Analysis Services. This is step 2 of 2. Run the following command on the computer that will be the active node in the failover cluster to make it useable. You must run the "CompleteFailoverCluster" action on the node that owns the shared disk in the Analysis Services failover cluster.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=CompleteFailoverCluster /INSTANCENAME="<Ins</pre>

The following two examples use two steps to create a SQL Server failover cluster instance with the Database Engine and Analysis Services. Test and modify the installation scripts to meet the needs of your organization.

To perform the "Preparation" step of a failover cluster advanced installation scenario for the Database Engine and Analysis Services. This is step 1 of 2.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt to prepare a default instance:

setup.exe /q /ACTION=PrepareFailoverCluster /InstanceName=MSSQLS

Run the following command at the command prompt to prepare a named instance:

To perform the "Completion" step of a failover cluster advanced installation scenario for the Database Engine and Analysis Services. This is step 2 of 2. Run the following command on the computer that will be the active node in the failover cluster to make it useable. You must run the "CompleteFailoverCluster" action on the node that owns the shared disk in the Analysis Services failover cluster.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt to complete failover cluster installation for a default instance:

setup.exe /q /ACTION=CompleteFailoverCluster /InstanceName=MSSQL

Run the following command at the command prompt to complete failover cluster installation for a named instance:

#### Add Node and Remove Node Scenarios:

Use the following syntax to add or remove nodes from an existing SQL Server failover cluster instance. The AddNode action gathers feature and setting information from the active node, installs the necessary components, and joins the new node to the existing SQL Server failover cluster. AddNode can be used for both integrated and advanced scenarios.

Test and modify the installation scripts to meet the needs of your organization.

To add a node to an existing failover cluster instance with the Database Engine.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=AddNode /INSTANCENAME="<Insert Instance Nam</pre>

To add a node to an existing failover cluster instance with Analysis Services.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=AddNode /INSTANCENAME="<Insert Instance Nam</pre>

To add a node to an existing failover cluster instance with the Database Engine and Analysis Services.

1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.

2. Run the following command at the command prompt:

setup.exe /q /ACTION=AddNode /INSTANCENAME="<Insert Instance Nam</pre>

To completely remove a SQL Server failover cluster instance, use the REMOVENODE action on each failover cluster node. Test and modify the scripts to meet the needs of your organization.

To remove a node from an existing failover cluster instance with the Database Engine.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=RemoveNode /INSTANCENAME="<Insert Instance</pre>

To remove a node from an existing failover cluster instance with Analysis Services.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=RemoveNode /INSTANCENAME="<Insert Instance</pre>

To remove a node from an existing failover cluster instance with the Database Engine and Analysis Services.

- 1. Insert the SQL Server 2008 installation media into the disk drive and open the root folder.
- 2. Run the following command at the command prompt:

setup.exe /q /ACTION=RemoveNode /INSTANCENAME="<Insert Instance</pre>

## □ Installation Scripts for Analysis Services

To install or upgrade Analysis Services at a command prompt, specify "**AS**" as one of the values for the **FEATURES** command-line option. For example, to install Analysis Services, you would specify **/FEATURES=AS** on the command line.

The following table shows the parameters available for installing an instance of Analysis Services.

Parameter	Default Value	Description
ASBACKUPDIR	C:\Program Files\Microsoft SQL Server\ <instance ID&gt;\OLAP\Backup For example, the Instance ID might look something like this: MSAS10.MSSQLSERVER.</instance 	Specifies the folder in wl the Analysis Services backup files are stored.
ASCOLLATION	Latin1_General_CI_AS	Specifies the collation setting for Analysis Services.
ASCONFIGDIR	C:\Program Files\Microsoft SQL Server\ <instance ID&gt;\OLAP\Config</instance 	Specifies the folder in wl the Analysis Services configuration files are installed.
ASDATADIR	C:\Program Files\Microsoft SQL Server\ <instance ID&gt;\OLAP\Data</instance 	Specifies the folder in w the Analysis Services cu are built.
ASLANGUAGE	1033	Specifies the locale identifier (LCID). The va 1033 represents US-Engl
ASLOGDIR	C:\Program Files\Microsoft SQL Server\ <instance ID&gt;\OLAP\Log</instance 	Specifies the folder in wl the Analysis Services log files are stored. Specifies whether the

ASPROVIDERMSOLAP	1	MSOLAP provider can r
		in process.
		Specifies the
		MSSQLServerOLAPSer
		service account domain a
		logon name. Use the
		argument
ASSVCACCOUNT	local system account	ASSVCPASSWORD to
		specify the password tha
		corresponds to the logon
		name. If you do not spec
		this argument, the defaul
		will be used.
		Specifies the
		MSSQLServerOLAPSer
		service account password
		that corresponds to the
		logon name that is specif
ASSVCPASSWORD	blank	by the argument
		ASSVCACCOUNT.
		Specifying a blank passw
		will cause an error. To us
		blank password do not
		specify this parameter.
		Specifies whether the
		MSSQLServerOLAPSer
		service is started
ASSVCSTARTUPTYPE	Automatic	automatically when the
		Windows operating syste
		is restarted. The options
		Automatic, Disabled, or
		Manual.
		Specifies the
ASSYSADMINACCOUNTS	blank	MSSQLServerOLAPSer
		administrator account
		password.
	C:\Program Files\Microsoft	Specifies the folder in w
ASIEMPDIR	SQL Server\ <instance< td=""><td>the Analysis Services</td></instance<>	the Analysis Services

	ID>\OLAP\Temp	temporary files are stored
		Specifies whether error
		reporting is enabled or
ERRORREPORTING	0	disabled. To enable error
		reporting, specify 1.
		Otherwise, specify 0.
		Specifies whether Softwa
		Quality Monitoring (SQI
SQMREPORTING	0	is enabled. To enable SQ
		reporting, specify 1.
		Otherwise, specify 0.

Test and modify the following Analysis Services installation scripts to meet the needs of your organization.

To install a new default instance of Analysis Services and Management Tools.

- 1. Insert the installation media into the disk drive and open the folder at <CD or DVD drive>:\SQLDev\Servers\, or navigate to the network build folder \\..\SQLDev\Servers\.
- 2. Run the following command at a command prompt:

setup.exe /ACTION=install /FEATURES=AS,Tools /IN

**Important** Do not use a blank password. Use a strong password.

To upgrade an existing instance of SQL Server 2000 Analysis Services or SQL Server 2005 Analysis Services

- 1. Insert the installation media into the disk drive and open the folder at <CD or DVD drive>:\SQLDev\Servers\, or navigate to the network build folder \\..\SQLDev\Servers\..
- 2. Run the following command at a command prompt:

setup.exe /ACTION=upgrade /FEATURES=AS /INSTANCE

To repair a broken SQL Server 2008 Analysis Services installation

- 1. Insert the installation media into the disk drive and open the folder at <CD or DVD drive>:\SQLDev\Servers\, or navigate to the network build folder \\..\SQLDev\Servers\..
- 2. Run the following command at a command prompt:

setup.exe /ACTION=repair /FEATURES=AS /INSTANCEM

To uninstall an instance of Analysis Services

- 1. Insert the installation media into the disk drive and open the folder at <CD or DVD drive>:\SQLDev\Servers\, or navigate to the network build folder \\..\SQLDev\Servers\..
- 2. Run the following command at a command prompt:

setup.exe /ACTION=uninstall /FEATURES=AS /INSTAM

# Installation Options and Scripts for Reporting Services

To install or upgrade Reporting Services from the command prompt, specify "**RS**" as one of the values for the /**FEATURES** command line option. Reporting Services is instance-aware so you must specify the /**INSTANCENAME** parameter. Use the following options to configure a Reporting Services installation:

Option	Valid Values	Required
	FilesOnlyMode	
RSInstallMode	DefaultNativeMode	Optional
	DefaultSharePointMode	

	<domain>\<user></user></domain>	
	"NT AUTHORITY\NETWORK SERVICE"	
RSSVCACCOUNT	"NT AUTHORITY\LOCAL SYSTEM"	Yes
	"NT AUTHORITY\LOCAL SERVICE"	
RSSVCPASSWORD	If you specify a built-in account, you can omit this value. Otherwise, specify the password of the domain user account.	Yes
	Manual	
RSSVCSTARTUPTYPE	Automatic	No
	Disabled	
		Required if the credentials you use to run Setup do not have
RSUPGRADEDATABASEACCOUNT	Database login for a remote report server database.	sysadmin or database owner

		permissions	ι
		on the	r
		report	ā
		server	5
		database.	5
			5
	The password for the login		-
	or an empty string if the		1
IN OT GIVE DELEASS WORD	login does not have a		1 :
	password.		1

Test and modify the following installation scripts to meet the needs of your organization.

To install Reporting Services server applications in files-only mode as the default instance using default values

- 1. Insert the SQL Server installation media into the disk drive.
- 2. Run the following command at the command prompt:

Setup /q /ACTION=install /FEATURES=RS /INSTANCENAME=MSSQLSERVER

If you intend to use default values, the command line can include minimal arguments. This example installs an un-configured report server in files-only mode without client tools, where the service account runs as Network Service. Notice that /RSINSTALLMODE is not specified. This is because files-only mode is the default installation mode and will be used if no other mode is specified.

Reporting Services requires an instance of the Database Engine to store application data. If you want to use an existing instance of the Database Engine for the report server databases, you must install Reporting Services in files-only mode so that you can specify a database server to use after installation is finished. For more information about files-only mode, see <u>Files-Only Installation</u> (Reporting Services).

To install Reporting Services, Database Engine Services, and client tools on a single computer in native mode in the default configuration

- 1. Insert the SQL Server installation media into the disk drive.
- 2. Run the following command at the command prompt:

setup /q /ACTION=install /FEATURES=SQL,RS,TOOLS /INSTANCENAME=MS

This script also installs Report Designer in Business Intelligence Development Studio and SQL Server Management Studio. To satisfy the requirements for default configuration, the Database Engine is also installed in the same instance as the report server. This script includes the full set of command line arguments for a report server installation, set to non-default values and fictional credentials. For more information about the default configuration modes, see <u>Default</u> <u>Configuration for a Native Mode Installation (Reporting Services)</u> and <u>Default</u> <u>Configuration for SharePoint Integrated Mode (Reporting Services)</u>.

To upgrade a SQL Server 2005 report server that uses a remote report server database

- 1. Insert the SQL Server installation media into the disk drive.
- 2. Run the following command at the command prompt:

setup /q /ACTION=upgrade /INSTANCENAME=MSSQLSERVER /RSUPGRADEDA1

Setup components and features within the instance are upgraded to the new version; you cannot upgrade specific features within the same instance.

This script requires that you specify credentials for one-time connection to the remote Database Engine instance that hosts the report server database. You can omit **RSUPGRADEDATABASEACCOUNT** if you have sysadmin or database update permissions on the remote server. For more information about upgrading a report server, see <u>Considerations for Upgrading Reporting Services</u>.

To uninstall Reporting Services, Database Engine, and client tools

- 1. Insert the SQL Server installation media into the disk drive.
- 2. Run the following command at the command prompt:

setup /q /ACTION=uninstall /FEATURES=SQL,RS,Tools /INSTANCENAME=

## <sup>B</sup> Installation Scripts for Integration Services

Integration Services is usually installed or upgraded at the same time as an instance of the SQL Server Database Engine. Although you can design and run Integration Services packages without an instance of the Database Engine, you cannot store SQL Server 2008 Integration Services packages in SQL Server unless an instance of the SQL Server 2008 Database Engine is available.

To install or upgrade Integration Services from the command prompt, specify "IS" as one of the values for the FEATURES command line option. Use the following options to configure the Integration Services service account:

Option	<b>Required?</b>	Description
		The account the will run the Integration Services
ISSVCAccount	Required	service. The recommended account is NT
		AUTHORITY\NETWORK SERVICE.
		The password for the account specified as
ISSVCPassword	Required	ISSVCAccount. When using the Network
		Service account, specify a blank password.
		The startup type for the service. Valid values are
ISSVCStartupType	Optional	Automatic, Manual, or Disabled. The default
		value is Automatic.

<sup>I</sup>Note:

Integration Services is not instance-specific. You can only install a single instance of Integration Services on a computer. You connect to the service from management tools and client applications by using the name of the server on which the service is running.

Test and modify the following installation scripts to meet the needs of your organization.

To install Integration Services with the default settings along with an instance of the SQL Server Database Engine and the client tools.

1. Insert the SQL Server installation media into the disk drive.

2. Run the following command at the command prompt:

setup.exe /q /ACTION=install /FEATURES=SQL,IS,Tools /ISSVCACCOUN

**Important** Do not use a blank password. Use a strong password.

### <sup>ℤ</sup>Note:

For the default instance of the Database Engine, specify "MSSQLSERVER" for *<InstanceName>*. For a named instance, specify the instance name. In this example, the default instance is specified.

## <sup>©</sup> Parameters for SQL Server Installation

Use the parameters listed in the table below to develop command prompt installation scripts.

SQL Server Component	Parameter Name	Description	Suppor
Analysis Services	/ASBACKUPDIR	Specifies the directory for Analysis Services backup files.	
Analysis Services	/ASCOLLATION	Specifies the collation setting for Analysis Services.	
Analysis Services	/ASCONFIGDIR	Specifies the directory for Analysis Services configuration files.	
Analysis Services	/ASDATADIR	Specifies the directory for Analysis Services data files.	
Analysis Services	/ASLANGUAGE	Specifies the locale for Analysis Services.	
Analysis Services	/ASLOGDIR	Specifies the directory for Analysis	

	Services log	
	files.	
	Specifies the	
	account for	
ASSVCACCOUNT	the Analysis	
	Services	
	service.	
	Specifies the	
	password for	
ASSVCPASSWORD	the Analysis	
	Services	
	service.	
	Specifies the	
	startup mode	2–Automatic,
ASSVCSTARTIDTVPF	for the	1=Disabled
	Analysis	
	Services	3=Manual
	service.	
	Specifies the	
	administrator	
ASSYSADMINACCOUNTS	credentials	
	for Analysis	
	Services.	
	Directory for	
	Analysis	
/ASTEMPDIR	Services	
	temporary	
	files.	
	Specifies	
	whether the	
A SPROVIDERMSOLAP	MSOLAP	1=enabled_0=
	provider can	i chubicu, o
	run in-	
	process.	
	Enables run-	
	as	
	credentials	
YENABLERANU	for SQL	
	ASSVCACCOUNT ASSVCPASSWORD ASSVCPASSWORD ASSVCSTARTUPTYPE ASSYSADMINACCOUNTS ASTEMPDIR ASTEMPDIR ASPROVIDERMSOLAP	Services log files. Specifies the account for the Analysis Services service. Specifies the password for the Analysis Services service. Specifies the startup mode for the Analysis Services service. Specifies the administrator credentials for Analysis Services. Specifies the administrator credentials for Analysis Services. Specifies the administrator credentials for Analysis Services. Specifies the administrator credentials for Analysis Services. Specifies the administrator credentials for Analysis Services. Enables run- as credentials for SQL

		Server Expresss installations.	
Database Engine	/INSTALLSQLDATADIR	Specifies the data directory for SQL Server data files.	1=enabled, 0=
Database Engine	/REBUILDDATABASES		
Database Engine	/SAPWD		
Database Engine	/SECURITYMODE		/SECURITYN
Database Engine	/SQLBACKUPDIR		
Database Engine	/SQLCOLLATION		
Database Engine	/SQLSVCACCOUNT		
Database Engine	/SQLSVCPASSWORD		
Database Engine	/SQLSVCStartuptype	Specifies the startup mode for the SQL Server service.	2=Automatic, 4=Disabled, 3=Manual
Database Engine	/SQLSYSADMINACCOUNTS	Use this parameter to provision users within SQL Server.	
Database Engine	/SQLTEMPDBDIR		
Database Engine	/SQLTEMPDBLOGDIR		
Database Engine	/SQLUSERDBDIR		

Database Engine	/SQLUSERDBLOGDIR		
Integration Services	/ISSVCAccount	Specifies the account for Integration Services.	DomainName System accou
Integration Services	/ISSVCPassword	Specifies the Integration Services password.	Required only account.
Integration Services	/ISSVCStartupType	Specifies the startup type for the Integration Services service.	
NLRegS extension of the DP	/NPENABLED		
NLRegS extension of the DP	/TCPENABLED		
Reporting Services	/RSInstallMode		
Reporting Services	/RSSVCACCOUNT		
Reporting Services	/RSSVCPASSWORD		
Reporting Services	/RSSVCStartupType		
Reporting Services	/RSUPGRADEDATABASEACCOUNT		
Reporting Services	/RSUPGRADEPASSWORD		
SQL Server Browser	/BROWSERSVCStartupType	Specifies the startup mode for SQL Server	2=Automatic, 4=Disabled,

		Browser	3=Manual
		service.	
SQL Server Agent	/AGTSVCACCOUNT	Specifies the account for the SQL Server service.	DomainName System accou
SQL Server Agent	/AGTSVCPASSWORD	Specifies the SQL Server Agent password	Required only account.
SQL Server Agent	/AGTSVCSTARTUPTYPE	Specifies the startup mode for the SQL Server Agent service.	2=Automatic, 4=Disabled, 3=Manual
SQL Server Setup	/ACTION	Specifies an installation work flow.	Install, Uninst
SQL Server Setup	/ERRORREPORTING	Error reporting for SQL Server.	1=enabled; 0=
SQL Server Setup	/FEATURES	Specifies components to install, uninstall, or upgrade.	SQL, SQLEnį FullText, AS, BIDS, BOL
SQL Server Setup	/HELP, H, ?	Displays the usage options for installation parameters.	
SQL Server Setup	/INDICATEPROGRESS	Specifies that the verbose Setup log file will be piped to the	

		console.	
SQL Server Setup	/INSTALLSHAREDDIR	Specifies a non-default installation dorectory for 64-bit	
SQL Server Setup	/INSTALLSHAREDWOWDIR	components. Specifies a non-default installation dorectory for 32-bit components.	
SQL Server Setup	/INSTANCEDIR	Specifies a non-default installation directory for shared components.	
SQL Server Setup	/INSTANCEIDSUFFIX	Specifies a non-default value for an InstanceID.	
SQL Server Setup	/INSTANCENAME	Specifies a non-default SQL Server instance name.	/InstanceNam or /InstanceNa Named Instan
SQL Server Setup	/Q	Specifies that Setup runs without any user interface.	
SQL Server Setup	/SQMREPORTING	Specifies feature usage reporting for SQL Server.	1=enabled; 0=

### $\Box$ See Also

#### Tasks

How to: Install SQL Server 2008 (Setup) How to: Read a SQL Server 2008 Setup Log File

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Add Features to an Instance of SQL Server 2008 (Setup)

e <u>See Also</u>

#### - -

<u>Installing SQL Server 2008</u> > <u>Installation How-to Topics</u> >

The SQL Server Installation Wizard provides a single feature tree for installation of all SQL Server components. You do not have to install each component individually.

- Database Engine
- Analysis Services
- Reporting Services
- Integration Services
- Replication
- Full-text search
- Management tools
- Connectivity components
- Sample databases, samples, and SQL Server Books Online

To add features to an instance of SQL Server 2008 from the command prompt, see <u>How to: Install SQL Server 2008 from the Command Prompt</u>.

## <sup>D</sup> Prerequisites

Before you continue, review topics in <u>Planning a SQL Server Installation</u>.

#### <sup>I</sup>∕ 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.

## - Procedures

To install SQL Server 2008

1.

Insert the SQL Server installation media. From the root folder, double-click setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click setup.exe.

• If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.

• Windows Installaer 4.5 is also required, and may be installed by the Installation Wizard. If you are prompted to restart your computer, restart and then restart SQL Server 2008 setup.exe.

• When prerequisites are installed, the Installation Wizard will launch the SQL Server Installation Center. To create a new installation of SQL Server 2008, click **New Installation or Add Features to an Existing Installation**.

• The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information about log files, see <u>How to: View SQL Server</u> <u>2008 Setup Log Files</u>.

• The Installation Wizard will install SQL Server prerequisites if they are not already on the computer. They include the following:

- .NET Framework 2.0
- SQL Server Native Client
- SQL Server Setup Support Files

To install prerequisites, click Install.

• The System Configuration Checker will verify the system state of your computer before Setup continues.

• On the Installation Type page, specify the instance you would like to update.

• On the Feature Selection page, select the components for your installation. A description for each component group appears in the right-hand pane after you select the feature name. You can select any combination of check boxes. For more information, see <u>Editions and Components of SQL Server 2008</u>. Each component can be installed only once on a given instance of SQL Server. To install multiple components, you must install an additional instance of SQL Server.

• The Disk Space Requirements page calculates the required disk space for the features you specify, and compares requirements to the available disk space on the computer where Setup is running. For more information, see <u>Disk Cost</u> <u>Summary</u>.

• Work flow for the remainder of this topic depends on the features you have specified for your installation. You might not see all of the pages, depending on your selections.

• On the Server Configuration — Service Accounts page, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features you selected to install.

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, are started manually, or are disabled. Microsoft recommends that you configure service accounts individually to provide least privileges for each service, where SQL Server services are granted the minimum permissions they need to complete their tasks. For more information, see <u>SQL Server Configuration - Service Accounts</u> and <u>Setting Up Windows Service Accounts</u>.

To specify the same login account for all service accounts in this instance of SQL Server, provide credentials in the fields at the bottom of the page.

**Security Note** Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

• Use the **Server Configuration - Collation** tab to specify non-default collations for the Database Engine and Analysis Services. For more information, see <u>SQL Server Configuration - Collation</u>.

• Use the Database Engine Configuration — Account Provisioning page to specify the following:

• Security Mode - select Windows Authentication or Mixed Mode Authentication for your instance of SQL Server. If you select Mixed Mode Authentication, you must provide a strong password for the built-in SQL Server system administrator account.

After a device establishes a successful connection to SQL Server, the security mechanism is the same for both Windows Authentication and Mixed Mode. For more information, see <u>Database Engine Configuration - Account Provisioning</u>.

• SQL Server Administrators — You must specify at least one system administrator for the instance of SQL Server. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for the instance of SQL Server. For more information, see <u>Database Engine Configuration - Account Provisioning</u>.

When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

• Use the Database Engine Configuration — Data Directories page to specify non-default installation directories. To install to default directories, click **Next**.

#### <sup>I</sup> Important:

If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.

For more information, see <u>Database Engine Configuration - Data Directories</u>.

• Use the Database Engine Configuration — FILESTREAM page to enable

FILESTREAM for your instance of SQL Server. For more information about FILESTREAM, see <u>Database Engine Configuration - Filestream</u>. To continue, click Next.

• Use the Analysis Services Configuration — Account Provisioning page to specify users or accounts that will have administrator permissions for Analysis Services. You must specify at least one system administrator for Analysis Services. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for Analysis Services. For more information, see <u>Analysis Services Configuration - Account Provisioning</u>.

When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

• Use the Analysis Services Configuration — Data Directories page to specify non-default installation directories. To install to default directories, click **Next**.

#### <sup>I</sup> Important:

If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.

For more information, see <u>Analysis Services Configuration - Data Directories</u>.

• Use the Reporting Services Configuration page to specify the type of Reporting Services installation to create. Options include the following:

- Native mode default configuration
- SharePoint mode default configuration
- Unconfigured Reporting Services installation

For more information about Reporting Services configuration modes, see <u>Report Server Installation Options</u>.

• On the Error and Usage Reporting page, specify the information you would

like to send to Microsoft that will help to improve SQL Server. By default, options for error reporting and feature usage are enabled. For more information, see <u>Error and Usage Report Settings</u>.

• The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified.

• The Ready to Install page displays a tree view of installation options that were specified during Setup.

• During installation, the Installation Progress page provides status so you can monitor installation progress as Setup proceeds.

• After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.

• If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For information about Setup log files, see <u>How to: View SQL Server 2008 Setup Log Files</u>.

## <sup>□</sup> Next Steps

Configure your SQL Server installation

- To reduce the attackable surface area of a system, SQL Server selectively installs and activates key services and features. For more information, see <u>Understanding Surface Area Configuration</u> on MSDN.
- For more information about using graphical tools and command prompt utilities, see <u>Configuring SQL Server 2008</u>.

### $\Box$ See Also

#### Tasks

How to: Upgrade to SQL Server 2008 (Setup)

#### Concepts

How to: View SQL Server 2008 Setup Log Files How to: Install SQL Server 2008 from the Command Prompt

© 2008 Microsoft Corporation. All rights reserved.
SQL Server Setup Help

How to: Rename a Computer that Hosts a Stand-Alone Instance of SQL Server



#### - -

<u>Installing SQL Server 2008</u> > <u>Installation How-to Topics</u> >

When you change the name of the computer that is running SQL Server, the new name is recognized during SQL Server startup. You do not have to run Setup again to reset the computer name. Instead, use thethe following steps to update system metadata that is stored in sys.servers and reported by the system function @@SERVERNAME. Update system metadata to reflect computer name changes for remote connections and applications that use @@SERVERNAME, or that query the server name from sys.servers.

The following steps cannot be used to rename an instance of SQL Server. They 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 part 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 computer renaming process differs from a computer that hosts a stand-alone instance.
- SQL Server does not support renaming computers that are involved in replication, except when you use 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 you rename a computer that is configured to use Reporting

Services, Reporting Services might not be available after the computer name change. For more information, see **Renaming a Report Server Computer**.

- When you rename a computer that is configured to use database mirroring, you must turn off database mirroring before the renaming operation. 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. Use the following steps to update system metadata.
- Users who connect to SQL Server through a Windows group that uses a hard-coded reference to the computer name might not be able to connect to SQL Server. This can occur after the rename 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 by using the new computer name after you have restarted SQL Server. To ensure that @@SERVERNAME returns the updated name of the local server instance, you should manually run the following procedure that applies to your scenario. The procedure you use depends on whether you are updating a computer that hosts a default or named instance of SQL Server.

# <sup>□</sup> To rename a computer that hosts a stand-alone instance of SQL Server

• For a renamed computer that hosts a default instance of SQL Server, run the following procedures:

Copy Code

```
sp_dropserver <old_name>
G0
sp_addserver <new_name>, local
G0
```

Restart the instance of SQL Server.

• For a renamed computer that hosts a named instance of SQL Server, run

the following procedures:

```
sp_dropserver <old_name\instancename>
G0
sp_addserver <new_name\instancename>, local
G0
```

Copy Code

Restart the instance of SQL Server.

# <sup>□</sup> After the Renaming Operation

After a computer has been renamed, any connections that used the old computer name must connect by using the new name.

## <sup>□</sup> 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. The following example shows the use of @@SERVERNAME.

SELECT @@SERVERNAME AS 'Server Name'

Copy Code

## <sup>□</sup> Additional Considerations

**Remote Logins -** If the computer has any remote logins, running **sp\_dropserver** might generate an error similar to the following:

Copy Co 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.

## <sup>□</sup> To drop remote logins

• For a default instance, run the following procedure:

```
Copy Code
```

sp\_dropremotelogin old\_name
G0

• For a named instance, run the following procedure:

Copy Code

sp\_dropremotelogin old\_name\instancename
G0

**Linked Server Configurations** - Linked server configurations will be affected by the computer renaming operation. Use **sp\_addlinkedserver** or **sp\_setnetname** to update computer name references. For more information, see the <u>sp\_addlinkedserver (Transact-SQL</u>) or <u>sp\_setnetname (Transact-SQL</u>) SQL Server Books Online topic on MSDN.

**Client Alias Names** - Client aliases that use named pipes will be affected by the computer renaming operation. For example, if an alias "PROD\_SRVR" was created to point to SRVR1 and uses the named pipes protocol, the pipe name will look like <u>\\SRVR1\pipe\sql\query</u>. After the computer is renamed, the path of the named pipe will no longer be valid and. For more information about named

pipes, see the <u>Creating a Valid Connection String Using Named Pipes</u> SQL Server Books Online topic on MSDN.

## □ See Also

## Concepts

Installing SQL Server 2008

SQL Server Setup Help How to: View SQL Server 2008 Setup Log Files



#### - -

Installing SQL Server 2008 > Installation How-to Topics >

Logging for SQL Server Setup has changed in this release. The main log you should look at is l<*drive*>:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\LOG\Summary.txt

Each execution of Setup will generate a new time-stamped log folder. For example, if you launch the SQL Server Installation Center page, it gets its own time-stamped log folder, and each Setup action invoked from that page gets its own as well, so you will probably see several time-stamped log folders in this directory. The time-stamped log folder name format is YYYMMDD\_hhmmss.

Detailed Setup logs are at the following location:

*<drive>*:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\LOG\.

When looking for errors in the detail log, search for the following phrases:

- Watson bucket
- Error:
- Exception has been

A typical Setup request goes through three execution phases:

- 1. Global rules check
- 2. Component update
- 3. User-requested action

Each of these phases will generate detail and summary logs, with additional log files being generated as appropriate. Setup is called at least three times per user-requested Setup action.

Typical log files generated will be:

- Detail\_GlobalRules.txt
- Detail\_ComponentUpdate.txt
- Detail.txt

The summary log file name format is Summary\_[machine name]\_timestamp\_[execution phase]

The final summary log is copied to %Program Files%\Microsoft SQL Server\100\setup bootstrap\log folder and named Summary.txt for quick reference.

Note: Setup will not CAB the log files unless there is an error.

MSI actions performed during Setup generate their own log files in the following format: [product feature]\_[cpu]\_[LCID (optional)]\_[attempt #].log. If an MSI execution fails, look in the associated MSI log for "return value 3" only for ENU versions.

Datastore files contain a snapshot of the state of all configuration objects being tracked by the Setup process, and are useful for troubleshooting configuration errors. XML file dumps are created for datastore objects for each execution phase. They are saved in their own log sub folder under the time-stamped log folder, as follows:

- Datastore\_GlobalRules
- Datastore\_ComponentUpdate
- Datastore

## $\Box$ See Also

## Concepts

Troubleshooting an Installation of SQL Server

SQL Server Setup Help How to: Read a SQL Server 2008 Setup Log File

# 

#### - -

<u>Installing SQL Server 2008</u> > <u>Installation How-to Topics</u> >

SQL Server Setup creates log files for each SQL Server component during installation; logging starts as soon as Setup.exe is launched.

## <sup>□</sup> To read a SQL Server Setup log file

1. Open Windows Explorer and navigate to the following location:

<drive>:\Program Files\Microsoft SQL Server\100\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:

<drive>:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\LOG\Files

Look at the most recently modified core log: SQLSetup[*xxxx*]\_[ComputerName]\_Core.log.

- 5. If an error was shown 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 installation, open the log file SQLSetup[*xxxx*]\_[*ComputerName*]\_SQL.log, and perform the 'UE 3' search to scan for errors.

## $\Box$ See Also

## Concepts

How to: View SQL Server 2008 Setup Log Files

SQL Server Setup Help How to: Uninstall an Existing Instance of SQL Server 2008 (Setup)



#### - -

<u>Installing SQL Server 2008</u> > <u>Installation How-to Topics</u> >

This article describes how to manually uninstall a stand-alone instance of SQL Server. By following the steps in this topic, you also prepare the system so that you can reinstall SQL Server.

<sup>II</sup> Important:
To maintain or update an instance of SQL Server, you must be a local
administrator with permission to log on as a service.

Consider the following important information before you use this procedure to uninstall SQL Server:

- We recommend that you use **Add or Remove Programs** in Control Panel to uninstall SQL Server.
- On a computer that is running SQL Server together with earlier 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 2008, 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 2008, where SQL Server 2000 is installed after SQL Server 2008.

This issue is caused by removal of the registration for the SQL Server SQL-DMO COM library. To re-enable Enterprise Manager and other programs that have SQL-DMO dependencies, register SQL-DMO by running **regsvr32.exe sqldmo.dll** at the command prompt. For more information, see <u>Troubleshooting an Installation of SQL Server</u>.

• Before you remove SQL Server components from a computer that has

the minimum required amount of physical memory, make sure that the page file size is sufficient. The page file size must be equal to two times the amount of physical memory. Insufficient virtual memory can cause an incomplete removal of SQL Server.

 In SQL Server 2005, the SQL Server Browser was an integrated installation together with the Database Engine or Analysis Services. In SQL Server 2008, the SQL Server Browser is a separate installation, performed automatically with the Database Engine or Analysis Services. If you have multiple instances of SQL Server 2008, the SQL Server Browser will uninstall automatically when the last instance of SQL Server 2008 is uninstalled.

If SQL Server 2005 exists on the system with one or more instances of SQL Server 2008, the SQL Server 2008 Browser will not be removed automatically when the last instance of SQL Server 2008 is uninstalled. The SQL Server Browser that is installed with SQL Server 2008 will remain on the system to ease connections to the instance of SQL Server 2005. You can leave the SQL Server 2008 Browser installed with SQL Server 2005 and it will continue to function correctly.

To uninstall all components of SQL Server 2008, you must uninstall the SQL Server Browser component manually from **Programs and Features** in **Control Panel**. If you remove the SQL Server 2008 Browser when a SQL Server 2005 named instance is present, connectivity to SQL Server 2005 might be disrupted. In this situation, you can reinstall SQL Server Browser in one of the following ways:

- Repair the instance of SQL Server 2005 by using **Programs and Features** in **Control Panel**.
- Install an instance of the SQL Server 2005 Database Engine or Analysis Services.

Before you remove SQL Server, 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. 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:

- 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, delete the local security groups for SQL Server components.
- **Save or rename** 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\100\Tools\Report Designer

## <sup>ℤ</sup>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 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 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.

## <sup>©</sup> To uninstall an instance of SQL Server 2008

- 1. To begin the uninstall process, from the **Start** menu, click **Control Panel**, and then double-click **Add or Remove Programs**.
- Select the SQL Server component to uninstall, and then click Change/Remove. This will start the SQL Server Installation Wizard.
- 3. Setup Support Rules will run to verify your computer configuration. To continue, click Ok.
- 4. On the Select Instance page, use the drop-down box to specify an instance of SQL Server to remove, or specify the option to remove only the SQL Server shared features and management tools. To continue, click Next.
- 5. On the Select Features page, specify the features to remove from the specified instance of SQL Server. To continue.
- 6. Removal rules will run to verify that the operation can complete successfully.
- 7. On the Ready to Uninstall page, review the list of components and features that will be uninstalled.

- 8. The Setup Progress page will display Setup status.
- 9. On the Completing Setup page, click **Close** to exit the Installation Wizard.
- 10. Repeat steps 2-9 until all SQL Server 2008 components have been removed.

## $\Box$ See Also

#### Tasks

How to: Read a SQL Server 2008 Setup Log File

#### Concepts

How to: View SQL Server 2008 Setup Log Files

SQL Server Setup Help Troubleshooting an Installation of SQL Server

- -

<u>Installing SQL Server 2008</u> >

To view troubleshooting information about installation and upgrade issues, see the <u>SQL Server 2008 Readme</u>.

## $\Box$ See Also

#### Tasks

How to: Read a SQL Server 2008 Setup Log File

#### Concepts

How to: View SQL Server 2008 Setup Log Files

SQL Server Setup Help Upgrading to SQL Server 2008

# 

#### -

You can upgrade instances of SQL Server 2000 or SQL Server 2005 to SQL Server 2008. Before running SQL Server Setup to upgrade to SQL Server 2008, review the following topics about the upgrade process.

# <sup>□</sup> In This Section

Торіс	Description	
Dlanning a SOI	Documents installation requirements, system configuration	
<u>Pidilillig a SQL</u>	checks run by SQL Server Setup, and security	
Server mstanation	considerations for a SQL Server installation.	
Version and	Documents supported upgrade paths to SQL Server 2008.	
Edition Upgrades		
Using Upgrade	Describes Upgrade Advisor, a tool that analyzes instances of	
Advisor to Prepare	SQL Server 2000 and SQL Server 2005 to identify known	
for Upgrades	upgrade issues.	
<u>Upgrade How-to</u>	Provides links to procedural topics for upgrading to SQL	
<u>Topics</u>	Server 2008.	
Backward	Lists the significant changes that might affect your	
<u>Compatibility</u>	applications after you upgrade to SQL Server 2008.	

## □ See Also

#### Concepts

Considerations for Upgrading the Database Engine Considerations for Upgrading Analysis Services Considerations for Upgrading Reporting Services Considerations for Upgrading Data Transformation Services Considerations for Upgrading Replicated Databases

#### **Other Resources**

**Considerations for Upgrading Integration Services** 

SQL Server Setup Help Version and Edition Upgrades

Ξ 🖌

<u>Upgrading to SQL Server 2008</u> >

The following upgrade scenarios are supported in this release of SQL Server:

Upgrade from	Supported Upgrade Paths
SQL Server 2000 (32-Bit) MSDE SP4 <sup>1</sup>	SQL Server 2008 Express
SQL Server 2000 (32-Bit) Standard SP4 <sup>1,4</sup>	SQL Server 2008 Standard SQL Server 2008 Enterprise
SQL Server 2000 (32-Bit) Developer SP4 <sup>1,4</sup>	No upgrade support.
SQL Server 2000 (32-Bit) Enterprise SP4 <sup>1,4</sup>	SQL Server 2008 Enterprise
SQL Server 2000 Enterprise Evaluation (32-bit, IA64, X64) <sup>4,5</sup>	No upgrade support.
SQL Server 2000 (64-Bit) IA64 Developer SP4 <sup>3,4,5</sup>	SQL Server 2008 (64-Bit) IA64 Developer
SQL Server 2000 (64-Bit) IA64 Enterprise SP4 <sup>3,4,5</sup>	SQL Server 2008 (64-Bit) IA64 Enterprise

SQL Server 2000 (32-Bit) Personal SP4 <sup>2</sup>	No upgrade support.
SQL Server 2005 (32-Bit) Express	SQL Server 2008 Express SQL Server 2008 Express Advanced
SQL Server 2005 (32-Bit) Express Advanced <sup>1</sup>	SQL Server 2008 Express Advanced
SQL Server 2005 (32-Bit) Workgroup <sup>1</sup>	No upgrade support.
SQL Server 2005 (32-Bit) Standard <sup>1</sup>	SQL Server 2008 Standard SQL Server 2008 Enterprise
SQL Server 2005 (32-Bit) Developer <sup>1</sup>	No upgrade support.
SQL Server 2005 (32-Bit) Enterprise <sup>1</sup>	SQL Server 2008 Enterprise
SQL Server 2005 Enterprise Evaluation (32-bit, IA64, X64)	No upgrade support.
SQL Server 2005 IA64 (64-bit) Developer	No upgrade support.
SQL Server 2005 IA64 (64-bit) Standard	SQL Server 2008 IA64 (64-bit) Enterprise
SQL Server 2005 IA64 (64-bit) Enterprise	SQL Server 2008 IA64 (64-bit) Enterprise

SQL Server 2005 X64 (64-bit) Developer	SQL Server 2008 X64 (64-bit) Developer
SQL Server 2005 X64 (64-bit) Standard	SQL Server 2008 X64 (64-bit) Standard SQL Server 2008 X64 (64-bit) Enterprise
SQL Server 2005 X64 (64-bit) Enterprise	SQL Server 2008 X64 (64-bit) Enterprise
SQL Server 2008 Express <sup>1</sup>	SQL Server 2008 Express SQL Server 2008 Express Advanced SQL Server 2008 Standard SQL Server 2008 Enterprise
SQL Server 2008 Express Advanced <sup>1</sup>	SQL Server 2008 Express Advanced SQL Server 2008 Standard SQL Server 2008 Enterprise
SQL Server 2008 Express x64 (64-bit)	SQL Server 2008 Express x64 (64- bit) SQL Server 2008 Express Advanced x64 (64-bit)
	SQL Server 2008 Express Advanced x64 (64-bit)

SQL Server 2008 Express Advanced x64 (64-bit)	SQL Server 2008 Standard x64 (64- bit) SQL Server 2008 Enterprise x64 (64-bit)
SQL Server 2008 Workgroup <sup>1</sup>	SQL Server 2008 Workgroup SQL Server 2008 Standard SQL Server 2008 Enterprise
SQL Server 2008 Web <sup>1</sup>	SQL Server 2008 Web
SQL Server 2008 Standard <sup>1,2</sup>	SQL Server 2008 Standard SQL Server 2008 Enterprise
SQL Server 2008 Developer <sup>1,2</sup>	SQL Server 2008 Workgroup SQL Server 2008 Standard SQL Server 2008 Developer SQL Server 2008 Enterprise
SQL Server 2008 Enterprise <sup>1,2</sup>	SQL Server 2008 Enterprise
SQL Server 2008 Enterprise Evaluation <sup>2</sup>	SQL Server 2008 Standard SQL Server 2008 Enterprise SQL Server 2008 Enterprise Evaluation

SQL Server 2008 IA64 (64-bit) Enterprise Evaluation <sup>2</sup>	SQL Server 2008 IA64 (64-bit) Enterprise SQL Server 2008 IA64 (64-bit) Enterprise Evaluation
SQL Server 2008 x64 (64-bit) Enterprise Evaluation <sup>2</sup>	SQL Server 2008 x64 (64-bit) Standard SQL Server 2008 x64 (64-bit) Enterprise SQL Server 2008 x64 (64-bit) Enterprise Evaluation
SQL Server 2008 IA64 (64-bit) Developer <sup>2</sup>	SQL Server 2008 IA64 (64-bit) Standard SQL Server 2008 IA64 (64-bit) Developer SQL Server 2008 IA64 (64-bit) Enterprise
SQL Server 2008 IA64 (64-bit) Standard <sup>2</sup>	SQL Server 2008 IA64 (64-bit) Enterprise
SQL Server 2008 x64 (64-bit) Standard <sup>2</sup>	SQL Server 2008 IA64 (64-bit) Standard SQL Server 2008 IA64 (64-bit) Enterprise
SQL Server 2008 IA64 (64-bit) Enterprise <sup>2</sup>	SQL Server 2008 IA64 (64-bit) Enterprise

SQL Server 2008 x64 (64-bit) Enterprise <sup>2</sup>	SQL Server 2008 x64 (64-bit)
SQL Server 2008 x64 (64-bit) Enterprise <sup>2</sup>	Enterprise

<sup>1</sup>This SQL Server edition can be upgraded to SQL Server 2008 on the 32-bit subsystem (WOW64) of a 64-bit server.

<sup>2</sup>Changing the edition of a SQL Server 2008 failover cluster is limited. The following edition downgrade scenarios are not supported for SQL Server 2008 failover clusters:

- SQL Server 2008 Enterprise to SQL Server 2008 Developer, Standard, or Enterprise Evaluation.
- SQL Server 2008 Developer to SQL Server 2008 Standard or Enterprise Evaluation.
- SQL Server 2008 Standard to SQL Server 2008 Enterprise Evaluation.
- SQL Server 2008 Enterprise Evaluation to SQL Server 2008 Standard.

<sup>3</sup>Upgrade of SQL Server 2000 (64-Bit) IA64 failover clusters are not supported.

<sup>4</sup>Upgrade of SQL Server 2000 Analysis Services to SQL Server 2008 is not supported on failover clusters.

<sup>5</sup>Upgrade of SQL Server 2000 (64-Bit) will not install SQL Server 2008 Management Tools. To install Management Tools, you musr re-run Setup after the upgrade is complete.

Upgrade Notes

- Cross-version instances of SQL Server 2008 are not supported. Version numbers of the Database Engine, Analysis Services, and Reporting Services components must be the same in an instance of SQL Server 2008.
- Before upgrading SQL Server, enable Windows Authentication for SQL Server Agent and verify the default configuration: that the SQL Server Agent service account is a member of the SQL Server sysadmin group.
- Before upgrading from one edition of SQL Server 2008 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 <u>Features Supported by the Editions of SQL Server</u> 2008.

- Cross-platform upgrade is not supported. You cannot upgrade a 32-bit instance of SQL Server to native 64-bit. However, you can upgrade a 32-bit instance of SQL Server to the WOW64: the 32-bit subsystem on a 64-bit server as noted in the table above. You can also back up or detach databases from a 32-bit instance of SQL Server, and then restore or attach them to an instance of SQL Server (64-bit) if the databases are not published in replication. In this case, you must also re-create any logins and other user objects in **master**, **msdb**, and **model** system databases.
- To upgrade to SQL Server 2008, you must be running a supported operating system. For more information, see <u>Hardware and Software</u> <u>Requirements for Installing SQL Server 2008</u>.
- Upgrade will be blocked if there is a pending restart.
- Upgrade will be blocked if the Windows Installer service is not running.
- Upgrade will be blocked if performance counters are corrupt.
- To upgrade an instance of SQL Server to a SQL Server failover cluster, the instance being upgraded must be a failover cluster. To upgrade a stand-alone instance of SQL Server to a SQL Server failover cluster, install a new SQL Server failover cluster and then migrate user databases from the stand-alone instance by using the Copy Database Wizard. For more information about upgrade, see <u>How to: Upgrade a SQL Server</u> <u>Failover Cluster Instance (Setup)</u>. For more information about database migration, see **Using the Copy Database Wizard**.
- To upgrade SQL Server 2005 to SQL Server 2008 on a computer that is running Windows Server 2008, you must be running SQL Server 2005 SP2. SQL Server 2005 SP1 is not a supported upgrade scenario.

# <sup>□</sup> Adding Components to an Instance of SQL Server

You cannot add components to an existing installation of SQL Server at the time you upgrade to SQL Server 2008. After you upgrade an instance of SQL Server to SQL Server 2008, you can add features by using the SQL Server 2008 Installation Wizard: Setup.exe.

# <sup>□</sup> Cross-Language Support

- The English-language version of SQL Server is supported on all localized versions of supported operating systems.
- Localized versions of SQL Server are supported on localized operating systems that are the same language as the localized SQL Server version.
- Localized versions of SQL Server can be upgraded to localized versions of SQL Server 2008 of the same language.
- Localized versions of SQL Server cannot be upgraded to the Englishlanguage version of SQL Server 2008.
- Localized versions of SQL Server cannot be upgraded to localized SQL Server 2008 versions of a different localized language.
- Localized versions of SQL Server are also supported on English-language versions of supported operating systems by using the Windows Multilingual User Interface Pack (MUI) settings. However, you must verify certain operating system settings before you install a localized version of SQL Server 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 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, you must correctly set these operating system settings before you install SQL Server 2008.

## □ See Also

#### Reference

Hardware and Software Requirements for Installing SQL Server 2008

#### Concepts

<u>Planning a SQL Server Installation</u> <u>Upgrading to SQL Server 2008</u> <u>Using Upgrade Advisor to Prepare for Upgrades</u>

### SQL Server Setup Help Using Upgrade Advisor to Prepare for Upgrades



- -

<u>Upgrading to SQL Server 2008</u> >

SQL Server Upgrade Advisor helps you prepare for upgrades to SQL Server 2008. Upgrade Advisor analyzes installed components from earlier versions of SQL Server, and then generates a report that identifies issues to fix either before or after you upgrade.

# <sup>□</sup> 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, 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.
# <sup>D</sup> Upgrade Advisor Analysis

Upgrade Advisor analyzes the following SQL Server components:

- Database Engine
- Analysis Services
- Reporting Services
- Integration Services
- Data Transformation Services
   Note: Notification Services is not included in Upgrade Advisor scans because it has been removed from SQL Server 2008.

The analysis examines objects that can be accessed, such as scripts, stored procedures, triggers, and trace files. Upgrade Advisor cannot analyze desktop applications or encrypted stored procedures.

Output is in the form of an XML report. View the XML report by using the Upgrade Advisor report viewer.

## <sup>I</sup>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 change your server or applications because of the undetectable issues.

# <sup>□</sup> How to Install and Run Upgrade Advisor

The location where you install SQL Server 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. For more information, see <u>Version and Edition</u> <u>Upgrades</u>. 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 installation media, and from the <u>Microsoft Download Center</u>.

Prerequisites for installing and running Upgrade Advisor are as follows:

- Windows XP SP2 or a later version, Windows Vista, Windows Server 2003 SP1 or a later version, or Windows Server 2008.
- Windows Installer 4.5 or a later version. The .NET Framework 2.0 requires Windows Installer 4.5. You can install Windows Installer from the <u>Windows Installer Web site</u>.
- The .NET Framework 2.0 or a later version. The .NET Framework 2.0 is available on the SQL Server 2005 product media, and from the <u>SDK</u>, <u>redistributable</u>, and <u>service pack download Web site</u>.
  - To install the .NET Framework 2.0 from the SQL Server 2005 media, locate the root of the disc drive. Then, 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 your operating system.
- SQL Server 2000 decision support objects (DSO) are required to scan upgrade issues in Analysis Services. To install DSO, insert the SQL Server 2000 media into the disc 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 SQL Server 2000 DTS packages. Use the SQL Server 2000 installation disc to install client components.

• SQL Server 2005 Backward Compatibility Components are required to scan SQL Server 2005 DTS packages that were migrated from SQL Server 2000. Use the SQL Server 2005 installation disc to install backward compatibility components.

To install Upgrade Advisor from the Web, click the download button on the download page. You can then run installation immediately, or save the SQLUA.msi file to run later. If you are installing from the product disc, run SQLUA.msi directly from the product disc.

After you install Upgrade Advisor, you can open it from the **Start** menu:

• Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2008**, and then click **SQL Server 2008 Upgrade Advisor**.

For more information, see the Upgrade Advisor documentation and the Upgrade Advisor Readme file installed with Upgrade Advisor.

## $\Box$ See Also

## Reference

Version and Edition Upgrades

## Concepts

Working with Multiple Versions and Instances of SQL Server Backward Compatibility

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Considerations for Upgrading the Database Engine

# 

#### - -

#### <u>Upgrading to SQL Server 2008</u> >

You can upgrade the SQL Server Database Engine to SQL Server 2008. For information about supported upgrade paths, see <u>Version and Edition Upgrades</u>.

The SQL Server Setup program can upgrade an earlier version of SQL Server with minimal user intervention. However, the correct preparation and a familiarity with the upgrade process can make it easier to prevent or solve any issues that occur.

This topic provides the information that 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.
- Links to procedural topics for migrating databases to SQL Server.
- Considerations for failover clusters.
- Post-upgrade tasks and considerations.

## <sup>□</sup> Known Upgrade issues

Before upgrading the Database Engine, review <u>SQL Server Database Engine</u> <u>Backward Compatibility</u>. For information about supported upgrade scenarios and upgrade known issues, see <u>Version and Edition Upgrades</u>. For backward compatibility content for other SQL Server components, see <u>Backward</u> <u>Compatibility</u>.

## <sup>I</sup> Important:

Before you upgrade from one edition of SQL Server to another, verify that the functionality that you are currently using is supported in the edition to which you are upgrading.

# <sup>D</sup> Pre-Upgrade Checklist

Upgrading SQL Server from an earlier version is supported by the SQL Server Setup program. You can also migrate databases from previous SQL Server versions. Migration can be from one SQL Server instance to another on the same computer, or from a SQL Server instance on another computer. Migration options include use of the Copy Database Wizard, Backup and restore functionality, use of the SQL Server Integration Services Import and Export Wizard, and bulk export/bulk import methods.

Before upgrading the Database Engine, make sure to do the following:

- Review <u>Hardware and Software Requirements for Installing SQL Server</u> 2008.
- Review <u>Check Parameters for the System Configuration Checker</u>.
- Review <u>Security Considerations for a SQL Server Installation</u>.
- Review <u>Version and Edition Upgrades</u>.
- Review <u>Using Upgrade Advisor to Prepare for Upgrades</u>.
- Review <u>SQL Server Database Engine Backward Compatibility</u>.
- Review <u>Migrating Query Plans</u>.

Review the following issues and make changes before you upgrade SQL Server:

- When upgrading from a 64-bit edition of SQL Server to a 64-bit edition of SQL Server 2008, you must upgrade Analysis Services before you upgrade the Database Engine.
- Back up all SQL Server database files from the instance to be upgraded, so that you can restore them, if it is required.
- Run the appropriate Database Console Commands (DBCC) on databases to be upgraded to ensure that they are in a consistent state.
- Estimate the disk space that is required to upgrade SQL Server components, in addition to user databases. For disk space that is required by SQL Server components, see <u>Hardware and Software Requirements for Installing SQL Server 2008</u>.
- Ensure that existing SQL Server system databases **master**, **model**, **msdb**, and **tempdb** are configured to autogrow, and ensure that they have sufficient 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 might 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 retains the configuration value for **max worker threads**. However, we recommend that you change the **max worker threads** value to 0 before upgrading, to let the Database Engine calculate the optimal number of threads. For more information, see **max worker threads Option** in SQL Server Books Online.
- Quit all applications, including all services that have SQL Server dependencies. Upgrade might fail if local applications are connected to the instance being upgraded.
- If you utilize Database Mirroring, see <u>How to: Minimize Downtime for</u> <u>Mirrored Databases When Upgrading Server Instances</u> in SQL Server Books Online.

# <sup>D</sup> Upgrading the Database Engine

You can overwrite an installation of SQL Server 2000 or SQL Server 2005 with a version upgrade. If an earlier version of SQL Server is detected when you run SQL Server Setup, all previous SQL Server program files are upgraded, and all data stored in the previous SQL Server instance is preserved. In addition, earlier versions of SQL Server Books Online will remain intact on the computer.

#### <sup>A</sup>Caution:

When you upgrade SQL Server, 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 by using the SQL Server Installation Wizard.

## <sup>A</sup>Caution:

If your computer has SQL Server 2000 Management Tools and a default instance of SQL Server 2008 installed, SQL Server Setup will let you install a default instance of SQL Server 2000. However, doing this will disable the installed instance of SQL Server 2008. Therefore, do not install a default instance of SQL Server when SQL Server 2000 Management Tools and a default instance of SQL Server 2008 already exist on the computer.

## **Database Compatibility Level After Upgrade**

The compatibility levels of the **tempdb**, **model**, **msdb** and **Resource** databases are set to 100 after upgrade. The **master** system database retains the compatibility level it had before upgrade, unless that level was less than 80. If the compatibility level of **master** was less than 80 before upgrade, it is set to 80 after upgrade.

If the compatibility level of a user database was 80 or 90 before upgrade, it remains the same after upgrade. If the compatibility level was 70 or less before

upgrade, in the upgraded database, the compatibility level is set to 80, which is the lowest supported compatibility level in SQL Server 2008.

## <sup>I</sup>∕ Note:

New user databases will inherit the compatibility level of the **model** database.

# <sup>□</sup> Migrating Databases

You can move user databases to an instance of SQL Server using backup and restore or detach and attach functionalities in SQL Server.

## <sup>I</sup> Important:

A database that has the identical name on both source and destination servers cannot be moved or copied. In this case, it will be noted as "Already exists."

For more information, see **Using the Copy Database Wizard** or **How to: Upgrade SQL Server with the Copy Database Wizard**.

# <sup>□</sup> After Upgrading the Database Engine

After upgrading the Database Engine, complete the following tasks:

- Re-register your servers. For more information about registering servers, see the **Registering Servers** topic in SQL Server Books Online.
- Re-populate full-text catalogs. This operation enhances the performance of your SQL Server installation. For more information, see the **sp\_fulltext\_catalog (Transact-SQL)** topic in SQL Server Books Online.
- Update statistics on all databases following upgrade. Use the **sp\_updatestats** stored procedure to update statistics in user-defined tables in SQL Server databases.
- Run **DBCC UPDATEUSAGE** on all databases to correct any incorrect row or page counts.
- Configure the SQL Server installation. To reduce the attackable surface area of a system, SQL Server selectively installs and enables key services and features.
- Validate or remove USE PLAN hints that are generated by SQL Server 2005 and applied to queries on partitioned tables and indexes. SQL Server 2008 changes the way queries on partitioned tables and indexes are processed. Queries on partitioned objects that use the USE PLAN hint for a plan that is generated by SQL Server 2005 might contain a plan that is not usable in SQL Server 2008. We recommend the following procedures after you upgrade to SQL Server 2008.

#### When the USE PLAN hint is specified directly in a query:

- 1. Remove the USE PLAN hint from the query.
- 2. Test the query.
- 3. If the optimizer does not select an appropriate plan, tune the query, and then consider specifying the USE PLAN hint with the desired query plan.

#### When the USE PLAN hint is specified in a plan guide:

- 1. Use the sys.fn\_validate\_plan\_guide function to check the validity of the plan guide. Alternatively, you can check for invalid plans by using the Plan Guide Unsuccessful event in SQL Server Profiler.
- 2. If the plan guide is not valid, drop the plan guide. If the optimizer does not select an appropriate plan, tune the query, and then consider specifying the USE PLAN hint with the query plan that you want.

A plan that is not valid will not cause the query to fail when the USE PLAN hint is specified in a plan guide. Instead, the query is compiled without using the USE PLAN hint. For more information about query processing on partitioned objects, see **Query Processing Enhancements on Partitioned Tables and Indexes**.

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:

```
EXEC sp_fulltext_service 'pause_indexing', 1
```

```
EXEC sp_fulltext_service 'pause_indexing', 0
```

```
Copy Code
```

## $\Box$ See Also

## Concepts

Working with Multiple Versions and Instances of SQL Server Backward Compatibility Considerations for Upgrading Replicated Databases

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Upgrade a Database Using Detach and Attach (Transact-SQL)



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading the Database Engine</u> >

## <sup>a</sup>Security Note:

We recommend that you do not attach or restore databases from unknown or untrusted sources. Such databases could contain malicious code that might execute unintended Transact-SQL code or cause errors by modifying the schema or the physical database structure. Before you use a database from an unknown or untrusted source, run **DBCC CHECKDB** on the database on a nonproduction server and also examine the code, such as stored procedures or other user-defined code, in the database.

In SQL Server 2008, you can use detach and attach operations to upgrade a user database from SQL Server 2000 or SQL Server 2005. After you attach a SQL Server 2005 or SQL Server 2000 database to SQL Server 2008, the database becomes available immediately and is then automatically upgraded.

However, the following restrictions apply:

- Copies of the **master**, **model** or **msdb** database created using SQL Server 2000 or SQL Server 2005 cannot be attached.
- When attaching a replicated database that was copied instead of detached:
  - If you attach the database to an upgraded version of the same server instance, you must execute **sp\_vupgrade\_replication** to upgrade replication after the attach operation finishes. For more information, see **sp\_vupgrade\_replication (Transact-SQL)**.
  - If you attach the database to a different server instance (regardless of version), you must execute **sp\_removedbreplication** to remove replication after the attach

operation finishes. For more information, see **sp\_removedbreplication (Transact-SQL)**.

• When the **APPLY**, **PIVOT**, **TABLESAMPLE**, or **UNPIVOT** keywords are used against databases that are upgraded from SQL Server 2000 to SQL Server 2008, the compatibility level of the database must be set to 100. To set the database compatibility level, see **sp\_dbcmptlevel** (**Transact-SQL**).

## <sup>Important:</sup>

In SQL Server 2000 Service Pack 3 (SP3) and later versions of SQL Server, attach and detach disable cross-database ownership chaining for the database by setting its **cross db ownership chaining** option to 0. For information about enabling chaining, see **cross db ownership chaining Option**.

# <sup>□</sup> Full-Text Index Upgrade Options

## <sup>I</sup> Note:

After you attach a SQL Server 2005 or SQL Server 2000 database to SQL Server 2008, the database becomes available immediately and is then automatically upgraded. If the database has full-text indexes, the upgrade process either imports, resets, or rebuilds them, depending on the setting of the **upgrade\_option** server property. If the upgrade option is set to import (**upgrade\_option** = 2) or rebuild (**upgrade\_option** = 0), the full-text indexes will be unavailable during the upgrade. Depending the amount of data being indexed, importing can take several hours, and rebuilding can take up to ten times longer. Note also that when the upgrade option is set to import, the associated full-text indexes are rebuilt if a full-text catalog is not available. To change the setting of the **upgrade\_option** server property, use **sp\_fulltext\_service**.

# - Procedures

To upgrade a database using detach and attach

1.

Detach the database from the instance of SQL Server 7.0 or SQL Server 2000 by using the **sp\_detach\_db** stored procedure.

For more information, see SQL Server Books Online for that version of SQL Server.

## <sup>I</sup>Note:

In SQL Server 2005, this stored procedure has new options. For more information, see **sp\_detach\_db (Transact-SQL)**.

• Optionally, move the detached database file or files and the log file or files.

You should move the log files along with the data files, even if you intend to create new log files. In some cases, reattaching a database requires its existing log files. Therefore, always keep all the detached log files until the database has been successfully attached without them.

## <sup>☑</sup> Note:

If you try to attach the database without specifying the log file, the attach operation will look for the log file in its original location. If the original copy of the log still exists in that location, that copy is attached. To avoid using the original log file, either specify the path of the new log file or remove the original copy of the log file (after copying it to the new location).

• Attach the copied files to the instance of SQL Server 2005 by using the **CREATE DATABASE** statement with the FOR ATTACH or FOR ATTACH\_REBUILD\_LOG option.

## <sup>I</sup>Note:

For information about how to attach a SQL Server 2005 database by using Object Explorer, see **How to: Attach a Database (SQL Server Management** 

#### Studio).

• We recommend that you run DBCC UPDATEUSAGE on the upgraded database.

In earlier versions of SQL Server, the values for the table and index row counts and page counts can become incorrect. Therefore, databases that were created on versions prior to SQL Server 2005 may contain incorrect counts. After you upgrade a database to SQL Server 2005, we recommend that you run DBCC UPDATEUSAGE to correct any invalid counts. This DBCC statement corrects the rows, used pages, reserved pages, leaf pages and data page counts for each partition in a table or index. For more information, see **DBCC UPDATEUSAGE (Transact-SQL)**.

• Optionally, if you are making a copy of the database (instead of moving it), you can reattach the original database on the instance of SQL Server 7.0 or SQL Server 2000 by using the **sp\_attach\_db** or **sp\_attach\_single\_file\_db** stored procedure.

For more information, see SQL Server Books Online for that version of SQL Server.

# <sup>□</sup> Managing Metadata on the Upgraded Server Instance

When you attach a database onto another server instance, to provide a consistent experience to users and applications, you might have to re-create some or all of the metadata for the database, such as logins, jobs, and permissions, on the other server instance. For more information, see **Managing Metadata When Making a Database Available on Another Server Instance**.

# <sup>E</sup> Examples

The following example upgrades the SQL Server 2000 pubs database to a SQL Server 2005 database by using Transact-SQL statements to detach and attach the database.

1. Connect SQL Server 2000 Query Analyzer to a server instance on which pubs is attached, and detach the database by using the sp\_detach\_db stored procedure.

Copy Code

```
USE master;
GO
EXEC sp_detach_db @dbname = N'pubs';
GO
```

2. For the purposes of this example, using the method of your choice, copy the pubs files (pubs.mdf and pubs\_log.ldf) from C:\Program Files\Microsoft SQL Server\MSSQL\Data\ (this is the default locations of **pubs** in SQL Server 2000) to C:\Program Files\Microsoft SQL

Server\MSSQL10.MSSQLSERVER\MSSQL\Data\ (this is the SQL Server 2005 data directory).

#### <sup>I</sup>Important:

For a production database, place the database and transaction log on separate disks.

## <sup>I</sup>Note:

To copy files over the network to a disk on a remote computer using the universal naming convention (UNC) name of the remote location. A UNC name takes the form \\Servername\Sharename\Path\Filename. As for writing files to the local hard disk, the user account used by SQL Server must have been granted the permissions required for reading or writing to a file on the remote disk.

3. Attach the copied pubs database and, optionally, log files to an instance of SQL Server 2005 (this example uses the same database name). In SQL Server Management Studio, open a new Query Editor query and connect to the server instance on which you want to attach the database.

#### <sup>☑</sup> Note:

For more information, see **SQL Server Management Studio Database Engine Query Editor Window**.

Execute the following CREATE DATABASE statement.

immediately visible in Object Explorer. To view the database, click the Object Explorer window, and select **View** > **Refresh**. When the **Databases** node is expanded, the newly attached database now appears in the list of databases.

4. Optionally, reattach the original pubs database to the instance of SQL Server 2000 by using the sp\_attach\_db stored procedure. In Query Analyzer, enter the following:

```
USE master;
Go
EXEC sp_attach_db @dbname = N'pubs',
  @filename1 =
    N'c:\Program Files\Microsoft SQL Server\MSSQL\Data\pubs.mc
  @filename2 =
    N'c:\Program Files\Microsoft SQL Server\MSSQL\Data\pubs_lc
GO
```

## □ See Also

## Concepts

Using Upgrade Advisor to Prepare for Upgrades

#### **Other Resources**

Detaching and Attaching Databases CREATE DATABASE (Transact-SQL) How to: Attach a Database (SQL Server Management Studio) How to: Upgrade SQL Server with the Copy Database Wizard Securing Data and Log Files

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Full-Text Search Upgrade



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading the Database Engine</u> >

Upgrading full-text search to SQL Server 2008 is done during setup and when database files and full-text catalogs from a previous version of SQL Server are attached, restored, or copied using the Copy Database Wizard.

This topic discusses the following aspects of full-text upgrade:

- <u>Upgrading a Server Instance</u>
- <u>Full-Text Upgrade Options</u>
- <u>Considerations for Choosing a Full-Text Upgrade Option</u>
- <u>Migrating Full-Text Indexes When Upgrading a Database to SQL Server</u> 2008
- <u>Considerations for Restoring a SQL Server 2005 Full-Text Catalog to</u> <u>SQL Server 2008</u>
- Attaching a SQL Server 2005 database to SQL Server 2008

## <sup>D</sup> Upgrading a Server Instance

For an in-place upgrade, an instance of SQL Server 2008 is set up side-by-side with the old version of SQL Server, and data is migrated. If the old version of SQL Server had full-text search installed, a new version of full-text search is automatically installed. Side-by-side install means that each of the following components exists at the instance-level of SQL Server.

#### Word breakers, stemmers, and filters

Each instance now uses its own set of word breakers, stemmers, and filters, rather than relying on the operating system version of these components. These components are also easier to register and configure at a per-instance level. For more information, see **Word Breakers and Stemmers** and **Full-Text Search Filters**.

Filter daemon host

The full-text filter daemon hosts are processes that safely load and drive third-party extensible components used for index and query, such as word breakers, stemmers, and filters, without compromising the integrity of the Full-Text Engine. A server instance uses a multithreaded process for all multithreaded filters and a single-threaded process for all single-threaded filters.

#### <sup>I</sup>Note:

SQL Server 2008 introduces a service account for the FDHOST Launcher service (MSSQLFDLauncher). This service propagates the service account information to the filter daemon host processes of a specific instance of SQL Server. For information about setting the service account, see **How to: Set the FDHOST Launcher (MSSQLFDLauncher) Service Account for Full-Text Search (SQL Server Configuration Manager)**.

In SQL Server 2005 and earlier versions, each full-text index resides in a full-text catalog that belongs to a filegroup, has a physical path, and is treated as a database file. In SQL Server 2008, a full-text catalog is a logical concept—a

virtual object—that refers to a group of full-text indexes. Therefore, a new fulltext catalog is not treated as a database file with a physical path. However, during upgrade of any full-text catalog that contains data files, a new filegroup is created on same disk. This maintains the old disk I/O behavior after upgrade. Any full-text index from that catalog is placed in the new filegroup if the root path exists. If the old full-text catalog path is invalid, the upgrade keeps the fulltext index in the same filegroup as base table or, for a partitioned table, in the primary filegroup.

For more information about the architecture of full-text search in SQL Server 2008, see **Full-Text Search Architecture**.

## <sup>I</sup>Note:

SQL Server 2005 Transact-SQL DDL statements that specify full-text catalogs work correctly.

## <sup>□</sup> Full-Text Upgrade Options

When upgrading a server instance to SQL Server 2008, the user interface allows you to choose one of the following full-text upgrade options.

#### Import

Full-text catalogs are imported. Typically, import is significantly faster than rebuild. For example, when using only one CPU, import runs about 10 times faster than rebuild. However, an imported full-text catalog does not use the new and enhanced word breakers introduced in SQL Server 2008, so you might want to rebuild your full-text catalogs eventually.

#### <sup>I</sup>Note:

Rebuild can run in multi-threaded mode, and if more than 10 CPUs are available, rebuild might run faster than import if you allow rebuild to use all of the CPUs.

If a full-text catalog is not available, the associated full-text indexes are rebuilt. This option is available for only SQL Server 2005 databases.

For information about the impact of importing full-text index, see "Considerations for Choosing a Full-Text Upgrade Option," later in this topic.

#### Rebuild

Full-text catalogs are rebuilt using the new and enhanced word breakers. Rebuilding indexes can take awhile, and a significant amount of CPU and memory might be required after the upgrade.

#### Reset

Full-text catalogs are reset. SQL Server 2005 full-text catalog files are removed, but the metadata for full-text catalogs and full-text indexes is retained. After being upgraded, all full-text indexes are disabled for change tracking and crawls are not started automatically. The catalog will remain empty until you manually issue a full population, after the upgrade completes.

# Considerations for Choosing a Full-Text Upgrade Option

When choosing the upgrade option for your upgrade, consider the following:

- How do you use word breakers? The SQL Server 2008 full-text search service includes new **word breakers and stemmers**. These might change the results of full-text queries from previous releases for a specific text pattern or scenario. Therefore, how you use word breakers is important when choosing a suitable upgrade option:
  - If the word breakers of the full-text language you use did not change in SQL Server 2008, or if recall accuracy is not critical to you, importing is suitable. Later, if you experience any recall issues, you can upgrade to the new word breakers simply by rebuilding your full-text catalogs. For more information, see "Word Breakers and Imported Full-text indexes," later in this section.
  - If you care about recall accuracy and you use one of the word breakers that have been improved in SQL Server 2008, rebuilding is suitable.
- Were any full-text indexes built on integer full-text key columns? Rebuilding performs internal optimizations that improve the query performance of the upgraded full-text index in some cases. Specifically, if you have full-text catalogs that contain full-text indexes for which the **fulltext key column** of the base table is an integer data type, rebuilding achieves ideal performance of full-text queries after upgrade. In this case, we highly recommend using the **Rebuild** option.

## <sup>I</sup>Note:

For full-text indexes in SQL Server 2008, we recommend that the column serving as the full-text key be an integer data type. For more information, see **Performance Tuning and Optimization (Full-Text Search)**.

• What is the priority for getting your server instance online? Importing or rebuilding during upgrade takes a lot of CPU resources, which delays getting the rest of the server instance upgraded and online. If getting the server instance online as soon as possible is important, and if you are willing to run a manual population after the upgrade, **Reset** is suitable.

# Using new Word Breakers After Importing a SQL Server 2005 Full-Text Index

When importing an SQL Server 2005 full-text index, it is important to realize that SQL Server 2008 includes new word breakers for many of the languages that exist in SQL Server 2005. Only the word breakers for English, Korean, Thai, and Chinese (all forms) remain the same. For other languages, SQL Server 2008 introduces a new generation of word breakers, which potentially might behave slightly differently from SQL Server 2005 word breakers in imported SQL Server 2005 full-text indexes.

If a full-text catalog was imported when a SQL Server 2005 database was upgraded to SQL Server 2008, one or more languages used by the full-text indexes in full-text catalog might now be associated with new word breakers. For those languages, when a query uses the new word breakers, mismatches between the query and the full-text index content might occur occasionally because of slight differences in the behavior of the old and new word breakers. In this case, to guarantee a total match between queries and the full-text index content, either:

- Rebuild the full-text catalog that contains the full-text index (ALTER FULLTEXT CATALOG catalog\_name REBUILD).
- Issue a FULL POPULATION on the full-text index (ALTER FULLTEXT INDEX ON *table\_name* START FULL POPULATION).

## <sup>I</sup>Important:

If your full-text catalogs and indexes are using unchanged word breakers, such as just the English word breakers, for instance, no mismatch can occur. The word breakers used at query times will be the ones used when the index was created at indexing and future query times.

For information about word breakers, see **Word Breakers and Stemmers**.

## <sup>D</sup> Upgrading Noise-Word Files to Stoplists

SQL Server 2005 noise words have been replaced by SQL Server 2008 stopwords. When a database is upgraded to SQL Server 2008 from a previous release, the noise-word files are no longer used in SQL Server 2008. However, the old noise-word files are stored in the FTDATA\ FTNoiseThresaurusBak folder, and you can use them later when updating or building the corresponding SQL Server 2008 stoplists.

After the upgrade:

- If you never added, modified, or deleted any noise-word files in your installation of SQL Server 2005, the system stoplist should meet your needs.
- If your noise-word files were modified in SQL Server 2005, those modifications are lost during upgrade. To re-create those updates, you must manually recreate those modifications in the corresponding SQL Server 2008 stoplist. For more information, see **ALTER FULLTEXT STOPLIST** (Transact-SQL).
- If you do not want to apply any stopwords to your full-text indexes (for example, if you deleted or erased your noise-word files in your SQL Server 2005 installation), you must turn off the stoplist for each upgraded full-text index. Run the following Transact-SQL statement (replacing *database* with the name of the upgraded database and *table* with the name of the *table*):

Copy Code

```
Use database;
ALTER FULLTEXT INDEX ON table
SET STOPLIST OFF;
GO
```

The STOPLIST OFF clause removes stop-word filtering, and it will trigger a population of the table, without filtering any words considered to be noise.

# <sup>□</sup> Back Up and Imported Full-Text Catalogs

For full-text catalogs that are rebuilt or reset during upgrade (and for new full-text catalogs), the fulltext catalog is a logical concept and does not reside in a filegroup. Therefore, to back up a full-text catalog in SQL Server 2008, you must identify every filegroup that contains a full-text index of the catalog and back them up, one by one. For more information, see **Backing Up and Restoring a SQL Server 2008 Full-Text Catalog**.

For full-text catalogs that have been imported from SQL Server 2005, the full-text catalog is still a database file in its own filegroup. The SQL Server 2005 backup process for full-text catalogs still applies except that the MSFTESQL service does not exist in SQL Server 2008. For information about the SQL Server 2005 process, see <u>"Backing Up and Restoring Full-Text Catalogs" in SQL Server 2005 Books Online</u>.

## <sup>I</sup>Note:

For information about backup and restore of a SQL Server 2008 database, see **Backing Up and Restoring a SQL Server 2008 Full-Text Catalog**.

# <sup>B</sup> Migrating Full-Text Indexes When Upgrading a Database to SQL Server 2008

Database files and full-text catalogs from a previous version of SQL Server can be upgraded to an existing SQL Server 2008 server instance by using attach, restore, or the Copy Database Wizard. SQL Server 2005 full-text indexes, if any, are either imported, reset, or rebuilt. SQL Server 2000 full-text indexes, if any, are rebuilt or reset. The **upgrade\_option** server property controls which full-text upgrade option the server instance uses during these database upgrades.

After you attach, restore, or copy any SQL Server 2005 or SQL Server 2000 database to SQL Server 2008, the database becomes available immediately and is then automatically upgraded. Depending on the amount of data being indexed, importing can take several hours, and rebuilding can take up to ten times longer. Note also that when the upgrade option is set to import, if a full-text catalog is not available, the associated full-text indexes are rebuilt.

#### To change full-text upgrade behavior on a server instance

- Transact-SQL: Use the **upgrade\_option** action of **sp\_fulltext\_service**
- SQL Server Management Studio : Use the **Full-Text Upgrade Option** of the **Server Properties** dialog box. For more information, see **How to: View or Change Server Properties for Full-Text Search (SQL Server Management Studio)**.

# Considerations for Restoring a SQL Server 2005 Full-Text Catalog to SQL Server 2008

One method of upgrading fulltext data from a SQL Server 2005 database to SQL Server 2008 is to restore a full database backup to SQL Server 2008.

While importing a catalog, you can back up and restore the database and the catalog file. The behavior is the same as in SQL Server 2005:

- The full database backup will include the full-text catalog. To refer to the full-text catalog, use its SQL Server 2005 file name, sysft\_+*catalog-name*.
- If the full-text catalog is offline, the backup will fail.

For more information about backing up and restoring SQL Server 2005 full-text catalogs, see <u>Backing Up and Restoring Full-Text Catalogs</u> and <u>File Backup and</u> <u>Restore and Full-Text Catalogs</u> in SQL Server 2005 Books Online.

When the database is restored on SQL Server 2008, a new database file will be created for the full-text catalog. The default name of this file is ftrow\_catalog-name.ndf. For example, if your catalog-name is cat1, the default name of the SQL Server 2008 database file would be ftrow\_cat1.ndf. But if the default name is already being used in the target directory, the new database file would be named ftrow\_catalog-name{GUID}.ndf, where GUID is the Globally Unique Identifier of the new file.

After the catalogs have been imported, the **sys.database\_files** and **sys.master\_files** are updated to remove the catalog entries and the **path** column in **sys.fulltext\_catalogs** is set to NULL.

To back up a database

- Full Database Backups
- Creating Transaction Log Backups (full recovery model only)

To restore a database backup

- Performing a Complete Database Restore (Simple Recovery Model)
- Performing a Complete Database Restore (Full Recovery Model)

<sup>I</sup>Note:

For information about restoring SQL Server 2008 full-text indexes, see **Piecemeal Restore and Full-Text Indexes**.

## Example

The following example uses the MOVE clause in the **RESTORE** statement, to restore a SQL Server 2005 database named ftdb1. The SQL Server 2005 database, log, and catalog files are moved to new locations on the SQL Server 2008 server instance, as follows:

- The database file, ftdb1.mdf, is moved to C:\Program Files\Microsoft SQL Server\MSSQL.1MSSQL10.MSSQLSERVER\MSSQL\DATA\ftdb1.mdf.
- The log file, ftdb1\_log.ldf, is moved to a log directory on your log disk drive, *log\_drive*:\*log\_directory*\ftdb1\_log.ldf.
- The catalog files that correspond to the sysft\_cat90 catalog are moved to C:\temp. After the full-text indexes are imported, they will automatically be placed in a database file, C:\ftrow\_sysft\_cat90.ndf, and the C:\temp will be deleted.

RESTORE DATABASE [ftdb1] FROM DISK = N'C:\temp\ftdb1.bak' WITH FIL MOVE N'ftdb1' TO N'C:\Program Files\Microsoft SQL Server\MSSQL10. MOVE N'ftdb1\_log' TO N'log\_drive:\log\_directory\ftdb1\_log.ldf', MOVE N'sysft\_cat90' TO N'C:\temp';

# <sup>□</sup> Attaching a SQL Server 2005 database to SQL Server 2008

In SQL Server 2008, a full-text catalog is a logical concept that refers to a group of full-text indexes. The full-text catalog is a virtual object that does not belong to any filegroup. However, when you attach a SQL Server 2005 database that contains full-text catalog files onto a SQL Server 2008 server instance, the catalog files are attached from their previous location along with the other database files, the same as in SQL Server 2005.

The state of each attached full-text catalog on SQL Server 2008 is the same as when the database was detached from SQL Server 2005. If any full-text index population was suspended by the detach operation, the population is resumed on SQL Server 2008, and the full-text index becomes available for full-text search.

If SQL Server 2008 cannot find a full-text catalog file or if the full-text file was moved during the attach operation without specifying a new location, the behavior depends on the selected full-text upgrade option. If the full-text upgrade option is **Import** or **Rebuild**, the attached full-text catalog is rebuilt. If the full-text upgrade option is **Reset**, the attached full-text catalog is reset.

For more information about detaching and attaching a database, see **Detaching and Attaching Databases**, **CREATE DATABASE (Transact-SQL)**, **sp\_attach\_db**, and **sp\_detach\_db (Transact-SQL)**.
# $\Box$ See Also

**Other Resources** 

Getting Started with Full-Text Search Full-Text Engine Word Breakers and Stemmers Full-Text Search Filters

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Migrating Query Plans



#### -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading the Database Engine</u> >

In most cases, upgrading a database to SQL Server 2008 will result in improved query performance. However, if you have mission-critical queries that have been carefully tuned for performance, you may want to preserve the query plans for these queries before upgrading by creating a plan guide for each query. If, after upgrading, the query optimizer chooses a less efficient plan for one or more of the queries, you can enable the plan guides and force the query optimizer to use the pre-upgrade plans.

To create plan guides before upgrading follow these steps:

- Record the current plan for each mission critical query by using the sp\_create\_plan\_guide stored procedure and specifying the query plan in the USE PLAN query hint.
- 2. Verify that the plan guide is applied to the query.
- 3. Upgrade the database to SQL Server 2008.
- The plans are persisted in the upgraded database in the plan guides and serve as a fallback in case of plan regressions after the upgrade.We recommend that you not enable the plan guides after the upgrade because you might miss opportunities for better plans in the new release or beneficial recompiles due to updated statistics.
- 4. If less efficient plans are chosen after the upgrade, activate all or a subset of the plan guides to override the new plans.

# <sup>□</sup> Example

The following example shows how to record a pre-upgrade plan for a query by creating a plan guide.

#### Step 1: Collect the Plan

The query plan recorded in the plan guide must be in XML format. XMLformatted query plans can be produced through the following ways:

- SET SHOWPLAN\_XML
- SET STATISTICS XML
- Querying the **query\_plan** column of the **sys.dm\_exec\_query\_plan** dynamic management function.
- The SQL Server Profiler Showplan XML, Showplan XML Statistics Profile, and Showplan XML For Query Compile event classes.

For more information about producing and analyzing query plans, see **Analyzing a Query**.

The following example collects the query plan for the statement SELECT City, StateProvinceID, PostalCode FROM Person.Address ORDER BY PostalCode DESC; by querying dynamic management views.

```
USE AdventureWorks;
G0
SELECT query_plan
    FROM sys.dm_exec_query_stats AS qs
    CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS st
    CROSS APPLY sys.dm_exec_text_query_plan(qs.plan_handle, DEFAULT,
    WHERE st.text LIKE N'SELECT City, StateProvinceID, PostalCode FR
G0
```

#### Step 2: Create the Plan Guide to Force the Plan

Using the XML-formatted query plan (obtained by any of the methods previously described) in the plan guide, copy and paste the query plan as a string literal inside the USE PLAN query hint specified in the OPTION clause of

#### sp\_create\_plan\_guide.

Within the XML plan itself, escape quotation marks (') that appear in the plan with a second quotation mark before creating the plan guide. For example, a plan that contains WHERE A.varchar = 'This is a string' must be escaped by modifying the code to WHERE A.varchar = ''This is a string''.

The following example creates a plan guide for the query plan collected in step 1 and inserts the XML Showplan for the query in the @hints parameter. For brevity, only partial Showplan output is included in the example.

```
EXECUTE sp_create_plan_guide
@name = N'Guide1',
@stmt = N'SELECT City, StateProvinceID, PostalCode FROM Person.Addre
@type = N'SQL',
@module_or_batch = NULL,
@params = NULL,
@hints = N'OPTION(USE PLAN N''<ShowPlanXML xmlns=''''http://schemas.
Version='''0.5'''' Build=''''9.00.1116''''>
<BatchSequence><Batch><Statements><StmtSimple>
...
</StmtSimple></Statements></Batch>
</BatchSequence></ShowPlanXML>'')';
G0
```

#### Step 3: Verify That the Plan Guide Is Applied to the Query

Run the query again and examine the query plan that is produced. You should see that the plan matches the one that you specified in the plan guide.

# $\Box$ See Also

**Other Resources** 

sp\_create\_plan\_guide (Transact-SQL) Using the USE PLAN Query Hint Plan Forcing Scenarios and Examples

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help How to: Minimize Downtime for Mirrored Databases When Upgrading Server Instances



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading the Database Engine</u> >

This topic describes how to minimize downtime for mirrored databases when you update your server instances from Microsoft SQL Server 2005 to Microsoft SQL Server 2008. This process involves sequentially upgrading the instances of SQL Server 2005 that are participating in database mirroring. This form of upgrade, which is known as a *rolling upgrade*, reduces downtime to only a single failover. Note that for high-performance mode sessions in which the mirror server is geographically distant from the principal server, a rolling upgrade might be inappropriate.

# <sup>ℤ</sup>Note:

For information about performing a rolling upgrade to install a service pack or hotfix, see **How to: Install a Service Pack on a System with Minimal Downtime for Mirrored Databases**.

A rolling upgrade is a multi-stage process that consists of the following stages:

- Protect your data.
- If the session includes a witness, we recommend that you remove the witness. Otherwise, when the mirror server instance is being upgraded, database availability depends on the witness that remains connected to the principal server instance. After you remove a witness, you can upgrade it at any time during the rolling upgrade process without risking database downtime.

## <sup>ℤ</sup>Note:

For more information, see **Quorum: How a Witness Affects Database Availability**.

- If a session is running in high-performance mode, change the operating mode to high-safety mode.
- Upgrade each server instance that is involved in database mirroring. A rolling upgrade involves upgrading the server instance that is currently the mirror server, manually failing over each of its mirrored databases, and upgrading the server instance that was first the principal server (and is now the new mirror server). At this point, you will have to resume mirroring.

#### <sup> </</sup> Note:

Before starting a rolling upgrade, we recommend that you perform a practice manual failover on at least one of your mirroring sessions.

- Revert to high-performance mode, if it is required.
- Return the witness to the mirroring session, if it is required.

The procedures for these stages are described here.

#### <sup>I</sup>Important:

A server instance might be performing different mirroring roles (principal server, mirror server, or witness) in concurrent mirroring sessions. In this case, you will have to adapt the basic rolling upgrade process accordingly. For more information, see **Database Mirroring Sessions**.

#### <sup>□</sup> To protect your data before an upgrade (a best practice)

1. Perform a full database backup on every principal database.

#### To back up a database

- How to: Back Up a Database (SQL Server Management Studio).
- How to: Create a Full Database Backup (Transact-SQL)

2. Run the **DBCC CHECKDB** command on every principal database.

#### <sup>□</sup> To remove a witness from a session

1. If a mirroring session involves a witness, we recommend that you remove the witness before you perform a rolling upgrade.

#### To remove the witness

- How to: Remove the Witness from a Database Mirroring Session (SQL Server Management Studio)
- How to: Remove the Witness from a Database Mirroring Session (Transact-SQL)

# <sup>□</sup> To change a session from high-performance mode to high-safety mode

- 1. If a mirroring session is running in high-performance mode, before you perform a rolling upgrade, change the operating mode to high safety without automatic failover. Use one of the following methods:
  - In SQL Server Management Studio: Change the Operating mode option to High safety without automatic failover (synchronous) by using the Mirroring Page of the Database Properties dialog box. For information about how to access this page, see How to: Start the Configuring Database Mirroring Security Wizard (SQL Server Management Studio).
  - In Transact-SQL: Set transaction safety to FULL. For more information, see How to: Change Transaction Safety in a Database Mirroring Session (Transact-SQL)

<sup>ℤ</sup> Note:

For more information, see **Transact-SQL Settings and Database Mirroring Operating Modes**.

# $^{\Box}$ To perform the rolling upgrade

1. To minimize downtime, we recommend the following: Start the rolling upgrade by updating any mirroring partner that is currently the mirror server in all its mirroring sessions. You might have to update multiple server instances at this point.

### <sup>I</sup>Note:

A witness can be upgraded at any point in the rolling upgrade process. For example, if a server instance is a mirror server in Session 1 and is a witness in Session 2, you can upgrade the server instance now.

The server instance to upgrade first depends on the current configuration of your mirroring sessions, as follows:

- If any server instance is already the mirror server in all its mirroring sessions, upgrade the server instance to the new version.
- If all your server instances are currently the principal server in any mirroring sessions, select one server instance to upgrade first. Then, manually fail over each of its principal databases and upgrade that server instance.

After being upgraded, a server instance automatically rejoins each of its mirroring sessions.

#### To perform a manual failover

- How to: Manually Fail Over a Database Mirroring Session (SQL Server Management Studio)
- How to: Manually Fail Over a Database Mirroring Session (Transact-SQL).

For information about how manual failover works, see **Manual Failover**.

2. For each mirroring session whose mirror server instance has just been

upgraded, wait for the session to synchronize. Then, connect to the principal server instance, and manually fail over the session. On failover, the upgraded server instance becomes the principal server for that session, and the former principal server becomes the mirror server.

The goal of this step is for another server instance to become the mirror server in every mirroring session in which it is a partner.

#### Restrictions after you failover to an upgraded server instance.

After failing over from a SQL Server 2005 server instance to a SQL Server 2008 server instance, the database session is suspended. It cannot be resumed until the other partner has been upgraded. However, the principal server is still accepting connections and allowing data access and modifications on the principal database.

#### <sup>I</sup>Note:

Establishing a new mirroring session requires that the server instances all be running the same version of SQL Server.

- 3. After you fail over, we recommend that you run the **DBCC CHECKDB** command on the principal database.
- 4. Upgrade each server instance that is now the mirror server in all mirroring sessions in which it is a partner. You might have to update multiple servers at this point.

#### <sup>I</sup>Important:

In a complex mirroring configuration, some server instance might still be the original principal server in one or more mirroring sessions. Repeat steps 2–4 for those server instances until all instances involved are upgraded.

5. Resume the mirroring session.

#### <sup>I</sup>Note:

Automatic failover will not work until the witness has been upgraded.

6. Upgrade any remaining server instance that is the witness in all its mirroring sessions. After an upgraded witness rejoins a mirroring session, automatic failover becomes possible again. You might have to update multiple servers at this point.

# <sup>□</sup> To return a session to high-performance mode

- 1. Optionally, return to high-performance mode by using one of the following methods:
  - In SQL Server Management Studio: Change the Operating mode option to High performance (asynchronous) by using the Mirroring Page of the Database Properties dialog box.
  - In Transact-SQL: Use **ALTER DATABASE** to set transaction safety to OFF.

#### <sup>□</sup> To return a witness to a mirroring session

1. Optionally, in high-safety mode, reestablish the witness to each mirroring session.

#### To return a witness

- How to: Add or Replace a Database Mirroring Witness (SQL Server Management Studio)
- How to: Add a Database Mirroring Witness Using Windows Authentication (Transact-SQL)

## □ See Also

#### **Other Resources**

ALTER DATABASE Database Mirroring (Transact-SQL) BACKUP (Transact-SQL) How to: View the State of a Mirrored Database (SQL Server Management Studio) Database Mirroring Overview Synchronous Database Mirroring (High-Safety Mode) How to: Install a Service Pack on a System with Minimal Downtime for Mirrored Databases Role Switching During a Database Mirroring Session Database Mirroring Sessions Forced Service (with Possible Data Loss) How to: Start Database Mirroring Monitor Asynchronous Database Mirroring (High-Performance Mode)

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Upgrading SQL Server 2005 Log Shipping to SQL Server 2008

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading the Database Engine</u> >

It is possible to preserve log shipping configurations when upgrading from SQL Server 2005 to SQL Server 2008. This topic describes alternative scenarios and best practices for upgrading a log shipping configuration.

# <sup>ℤ</sup>Note:

**Backup compression** was introduced in SQL Server 2008 Enterprise. An upgraded log shipping configuration uses the **backup compression default** server-level configuration option to control whether backup compression is used for the transaction log backup files. The backup compression behavior of log backups can be specified for each log shipping configuration. For more information, see **How to: Enable Log Shipping (SQL Server Management Studio)**.

# <sup>□</sup> Protect Your Data Before the Upgrade

As a best practice, we recommend that you protect your data before a log shipping upgrade.

#### To protect your data

- 1. Perform a full database backup on every primary database. For more information, see:
  - How to: Back Up a Database (SQL Server Management Studio).
  - How to: Create a Full Database Backup (Transact-SQL)
  - BACKUP (Transact-SQL)
- 2. Run the **DBCC CHECKDB** command on every primary database.

# <sup>D</sup> Upgrading the Monitor Server Instance

The monitor server instance, if any, can be upgraded at any time.

While the monitor server is being upgraded, the log shipping configuration continues to work, but its status is not recorded in the tables on the monitor. Any alerts that have been configured will not be triggered while the monitor server is being upgraded. After the upgrade, you can update the information in the monitor tables by executing the **sp\_refresh\_log\_shipping\_monitor (Transact-SQL)** system stored procedure.

# The Upgrade Process for Configurations with a Single Secondary Server

The upgrade process described in this section assumes a configuration consisting of the primary server and only one secondary server. This configuration is represented in the following illustration, which shows a primary server instance, A, and a single secondary server instance, B.



For information about upgrading multiple secondary servers, see "Considerations for Upgrading Multiple Secondary Servers," later in this topic.

#### Upgrading the Secondary server instance

The upgrade process involves upgrading the secondary server instances of a SQL Server 2005 log shipping configuration to SQL Server 2008 before upgrading the primary server instance. Always upgrade the secondary server instance first. If the primary server were upgraded before a secondary server, log shipping would fail because a backup created on a newer version of SQL Server cannot be restored on an older version of SQL Server.

Log shipping continues throughout the upgrade process because the upgraded secondary servers continue to restore the log backups from the SQL Server 2005 primary server. The process for upgrading the secondary server instances depends partly on whether the log shipping configuration possesses multiple secondary servers. For more information, see "Considerations for Upgrading Multiple Secondary server instances," later in this topic.

While the secondary server instance is being upgraded, the log shipping copy and restore jobs do not run, so unrestored transaction log backups will accumulate. The amount of accumulation depends on the frequency of scheduled backup on the primary server. Also, if a separate monitor server has been configured, alerts might be raised indicating restores have not been performed for longer than the configured interval. Once the secondary server has been upgraded, the log shipping agents jobs resume and continue to copy and restore log backups from the primary server instance, server A. The amount of time required for the secondary server to bring the secondary database up to date varies, depending on the time taken to upgrade the secondary server and the frequency of the backups on the primary server.

#### <sup>ℤ</sup> Note:

During the server upgrade, the secondary database is not upgraded to a SQL Server 2008 database. It will get upgraded only if it is brought online.

## **Upgrading the Primary server instance**

When planning an upgrade, a significant consideration is the amount of time that your database will be unavailable. The simplest upgrade scenario involves the database being unavailable while you upgrade the primary server (scenario 1, below).

At the cost of a more complicated upgrade process, you can maximize your database availability by failing over the SQL Server 2005 primary server to a SQL Server 2008 secondary server before upgrading the original primary server (scenario 2, below). There are two variants of the failover scenario. You can switch back to the original primary server and keep the original log shipping configuration. Alternatively, you can remove the original log shipping configuration before upgrading the original primary server and later create a new configuration using the new primary server. This topic describes each of these scenarios.

### <sup>I</sup>Important:

Be sure to upgrade the secondary server instance before upgrading the primary server instance. For more information, see "Upgrading the Secondary server instance," earlier in this topic.

## **Scenario 1: Upgrade Primary Server Instance Without Failover**

This is the simpler scenario, but it causes more downtime than using failover. The primary server instance is simply upgraded and the database is unavailable during this upgrade.

Once the server is upgraded, the database is automatically brought back online, which causes it to be upgraded. After the database is upgraded, the log shipping jobs resume.

## Scenario 2: Upgrade Primary Server Instance with Failover

This scenario maximizes availability and minimizes downtime. It utilizes a controlled failover to the secondary server instance, which keeps the database available while the original primary server instance is upgraded. Downtime is limited to the relatively short time required to fail over, rather than the time required to upgrade the primary server instance.

Upgrading the primary server instance with failover involves three general procedures: performing a controlled failover to the secondary server, upgrading the original primary server instance to SQL Server 2008, and setting up log shipping on a SQL Server 2008 primary server instance. These procedures are described in this section.

#### <sup>I</sup> Important:

If you plan to have the secondary server instance as the new primary server instance, you need to remove the log shipping configuration. Log shipping will need to be reconfigured from the new primary to the new secondary, after the original primary server instance has been upgraded. For more information, see **Removing Log Shipping**.

#### Procedure 1. Perform a controlled failover to the secondary server

Controlled failover to the secondary server:

1. Manually perform a **tail-log backup** of the transaction log on the primary database specifying WITH NORECOVERY. This log backup captures any log records that have not been backed up yet and takes the database offline. Note that while the database is offline, the log shipping backup job will fail. The following example creates a tail log backup of the AdventureWorks database on the primary server. The backup file is named Failover\_AdventureWorks\_20080315.trn:

```
BACKUP LOG AdventureWorks
```

```
T0 DISK = N'\\FileServer\Log Shipping\AdventureWorks\Failover
WITH NORECOVERY;
```

GO

We recommend that you use a distinct file naming convention to differentiate the manually-created backup file from the backup files created by the log shipping backup job.

- 2. On the secondary server:
  - Ensure that all backups taken automatically by the log shipping backup jobs have been applied. To check which backup jobs have been applied, use the sp\_help\_log\_shipping\_monitor (Transact-SQL) system stored procedure on the monitor server or on the primary and secondary servers. The same file should be listed in the last\_backup\_file, last\_copied\_file, and last\_restored\_file columns. If any of the backup files have not been copied and restored, manually invoke the agent copy and restore jobs for the log shipping configuration. For more information, see How to: Start a Job (SQL Server Management Studio) or sp\_start\_job (Transact-SQL).
  - 2. Copy your the final log backup file that you created in step 1 from the file share to the local location that is used by log shipping on the secondary server.
  - 3. Restore the final log backup specifying WITH RECOVERY to bring the database online. As part of being brought online, the database will upgraded to SQL Server 2008.

The following example restores the tail log backup of the AdventureWorks database on the secondary database. The example uses the WITH RECOVERY option, which brings the database online:

```
RESTORE LOG AdventureWorks
```

```
FROM DISK = N'c:\log shipping\Failover_AdventureWorks_20(
WITH RECOVERY;
```

```
GO
```

# <sup>ℤ</sup> Note:

For a configuration that contains more than one secondary server, there are additional considerations. For more information, see "Considerations for Upgrading Multiple Secondary server instances," later in this topic.

- 4. Fail over the database by redirecting clients from the original primary server (server A) to the online secondary server (server B).
- 5. Take care that the transaction log of the secondary database does not fill while the database is online. To prevent the transaction log from filling, you might need to back it up. If so, we recommend that you back it up to a shared location, a *backup share*, to make the backups available for restoring on the other server instance.

# **Procedure 2. Upgrade the Original Primary Server Instance to SQL Server 2008**

After you upgrade the original primary server instance to SQL Server 2008, the database will still be offline and in the SQL Server 2005 format.

#### **Procedure 3. Set Up Log Shipping on SQL Server 2008**

The rest of the upgrade process depends on whether log shipping is still configured, as follows:

- If you have kept the SQL Server 2005 log shipping configuration, switch back to the original primary server instance. For more information, see "To switch back to the original primary server instance," later in this section.
- If you removed the log shipping configuration before failing over, create a new log shipping configuration in which the original secondary server instance is the new primary server instance. For more information, see "To keep the old secondary server instance as the new primary server instance," later in this section.

#### To switch back to the original primary server instance

 On the interim primary server (server B), back up the tail of the log using WITH NORECOVERY to create a tail-log backup and take the database offline. The tail log backup is named Switchback\_AdventureWorks\_20080315.trn.For example:

```
BACKUP LOG AdventureWorks
TO DISK = N'\\FileServer\Log Shipping\AdventureWorks\Switchba
WITH NORECOVERY;
```

GO

- 2. If any transaction log backups were taken on the interim primary database, other than the tail backup that you created in step 1, restore those log backups using WITH NORECOVERY to the offline database on the original primary server (server A). The database is upgraded to SQL Server 2008 format when the first log backup is restored.
- 3. Restore the tail-log backup, Switchback\_AdventureWorks\_20080315.trn, on the original primary database (on server A) using WITH RECOVERY to bring the database online.
- 4. Fail over back to the original primary database (on server A) by redirecting clients to the online secondary server from the original primary server.

After the database comes online, the original log shipping configuration will resume.

# To keep the old secondary server instance as the new primary server instance

Establish a new log shipping configuration using the old secondary server instance, B, as the primary server and the old primary server instance, A, as the new secondary server, as follows:

# <sup>☑</sup> Important:

The old log shipping configuration should have been removed from the original primary server at the start of the process before taking the manual transaction log backup that took the database offline.

- 1. To avoid performing a complete backup and restore of the database on the new secondary server (server A), apply the log backups from the new primary database to the new secondary database. In the example configuration, this involves restoring the log backups taken on server B to the database on server A.
- 2. Back up the log from the new primary database (on server B).
- 3. Restore the log backups to the new secondary server instance (server A) using WITH NORECOVERY. The first restore operation updates the database to SQL Server 2008.
- 4. Configure log shipping with the former secondary server (server B) as the

primary server instance.

#### <sup>Important:</sup>

If you use SQL Server Management Studio, specify that the secondary database is already initialized.

#### To configure log shipping

• How to: Enable Log Shipping (SQL Server Management Studio)

#### • How to: Enable Log Shipping (Transact-SQL)

5. Fail over the database by redirecting clients from the original primary server (server A) to the online secondary server (server B).

# <sup>I</sup> Important:

When you failover to a new primary database, you should ensure that its metadata is consistent with the metadata of the original primary database. For more information, see **Managing Metadata When Making a Database Available on Another Server Instance**.

# <sup>□</sup> Considerations for Upgrading Multiple Secondary server instances

This configuration is represented in the following illustration, which shows a primary server instance, A, and two secondary server instances, B and C.



Always upgrade all the secondary server instances before you upgrade the primary server.

# Upgrading With Failover and Switching Back to the Original Primary Server

When upgrading the primary instance with failover the process is more complex when there are multiple secondary server instances. In the following procedure, after all the secondary servers are upgraded, the primary server is failed over to one of the upgraded secondary databases. The original primary server is upgraded, and log shipping is failed over back to it.

- 1. Upgrade all the secondary server instances (server B and server C).
- 2. Obtain the tail of the transaction log of the primary database (on server A), and take the database offline, by backing up the transaction log using WITH NORECOVERY.
- 3. On the secondary server to which you plan to fail over (server B), bring the secondary database online, by restoring the log backup using WITH RECOVERY.
- 4. On every other secondary server (server C), leave the secondary database offline by restoring the log backup using WITH NORECOVERY.
   ✓ Note:

The log shipping copy and restore jobs will run on the secondary servers, but the jobs will do nothing because new log-backup files will not be placed on the backup share.

- 5. Fail over the database by redirecting clients from the original primary server (server A) to the online secondary server (server B). The online database becomes an interim primary server, keeping the database available while the original primary server is offline (server A).
- 6. Upgrade the original primary server (server A).
- 7. On the database to which you failed over—the interim primary database (on server B), manually back up the transaction log using WITH NORECOVERY. This takes the database offline.
- 8. Restore all transaction log backups that you created on the interim primary database (on server B) to every other secondary database (on server C) using WITH NORECOVERY. This allows log shipping to continue from the original primary database after its upgrade, without requiring a full database restore on each secondary database.
- 9. Restore the transaction log from the interim primary server (server B) to the original primary database (on server A) using WITH RECOVERY.

# Redeploying Log Shipping

If you do not want to migrate your log shipping configuration using one of the procedures shown above, you can redeploy log shipping from scratch by reinitializing your secondary database with a full backup and restore of the primary database. This may be a desirable option if you have a small database or if high availability is not crucial during the upgrade procedure.

For information about enabling log shipping using SQL Server Management Studio, see **How to: Enable Log Shipping (SQL Server Management Studio)**.

For information about enabling log shipping using Transact-SQL, see **How to: Enable Log Shipping (Transact-SQL)**.

# □ See Also

## Concepts

Migrating a SQL Server 2000 Log Shipping Configuration to SQL Server 2008

**Other Resources** 

Working with Transaction Log Backups Log Shipping Administration Log Shipping Tables and Stored Procedures

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

Migrating a SQL Server 2000 Log Shipping Configuration to SQL Server 2008



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading the Database Engine</u> >

You cannot upgrade a SQL Server 2000 log shipping configuration to SQL Server 2008. The Database Maintenance Plan Wizard, which was integral to log shipping in SQL Server 2000, is not used as part of the log shipping configuration beginning in SQL Server 2005. As a result, log shipping stops functioning when you upgrade a server to SQL Server 2008.

You can migrate a SQL Server 2000 log shipping configuration. This topic describes the methods of doing this:

- Migrating with failover allows you to maintain the availability of your database while you upgrade each server in the log shipping configuration. The secondary database does not need to be reinitialized.
- Migrating without failover offers a simpler procedure that does not require you to fail over to your secondary server. If you follow this procedure, your database is unavailable while you are upgrading the primary server. The secondary database does not need to be reinitialized.
- Redeploy log shipping completely, including reinitializing the secondary database from a full backup.

After you have migrated your log shipping configuration to SQL Server 2008, you can delete the tables and SQL Server Agent jobs that were created by SQL Server 2000 log shipping.

#### <sup>I</sup>Note:

**Backup compression** was introduced in SQL Server 2008 Enterprise. An upgraded log shipping configuration uses the **backup compression default** server-level configuration option to control whether backup compression is used for the transaction log backup files. The backup compression behavior of log backups can be specified for each log shipping configuration. For

more information, see **How to: Enable Log Shipping (SQL Server Management Studio)**.

# <sup>□</sup> Migrating with Failover

Migrating with failover involves keeping the primary server instance at the original version of SQL Server temporarily, while upgrading the secondary server instance. Upgrading a server instance upgrades only online databases. Offline databases, such as a log shipping secondary database, remain in the original version of SQL Server. As long as a database remains offline, log backups from the original version of SQL Server are restorable.

For purposes of clarity, this procedure refers to the SQL Server 2000 log shipping primary server instance as Server A and the SQL Server 2000 log shipping secondary server instance as Server B. This configuration is represented in the following illustration, which shows a primary server instance, A, and a single secondary server instance, B.



1. Upgrade the secondary server instance, Server B, to SQL Server 2008. When you upgrade Server B, the log shipping database remains a SQL Server 2000 database because it is offline. This database is upgraded in the next step. The SQL Server 2000 log shipping jobs that copy and restore log files will no longer function.

#### <sup>ℤ</sup>Note:

At this point, users can continue to access the primary database on Server A.

- 2. Fail over from the primary server instance, Server A, to the secondary server instance, Server B.
  - 1. To fail over, restore all unapplied transaction log backups from the primary database on server A to server B.
  - 2. Manually back up the tail of the transaction log on the primary database specifying WITH NORECOVERY. This log backup captures any log records that have not been backed up yet and takes the database offline.

The following example backs up the tail of the log of the AdventureWorks database on the primary server. The backup file is named Failover\_AdventureWorks\_20080315.trn:

```
BACKUP LOG AdventureWorks
   TO DISK = N'\\FileServer\Log Shipping\AdventureWorks\Fail
  WITH NORECOVERY;
```

GO

3. Bring the secondary database on server B online by restoring the transaction log tail WITH RECOVERY.

The failover process brings your secondary database online, after which it is automatically upgraded to a SQL Server 2008 database. The database upgrade process is fully logged.

#### Note:

After being upgraded, the log shipping database on Server B is available to users. Until SQL Server 2008 log shipping is configured on Server B, however, log backups from the database on Server B are not being taken.

- 3. Upgrade Server A to SQL Server 2008. The log shipping database temporarily remains a SQL Server 2000 database because it is offline.
- 4. On Server B, configure SQL Server 2008 log shipping with Server B as the primary server and Server A as the secondary server. After you begin shipping transaction logs to Server A, the log shipping database on Server A is upgraded to a SQL Server 2008 database when the first log backup is applied.

#### **Important:**

When you configure log shipping on Server B, be sure to specify the **No**, the secondary database is initialized option on the Initialize Secondary **Database** tab of the **Secondary Database Settings** dialog box. For more information, see How to: Enable Log Shipping (SQL Server Management Studio).

5. Optionally, if you want to make Server A your primary server again, fail over to Server A. For more information, see Changing Roles Between **Primary and Secondary Servers.** 

# <sup>□</sup> Migrating Without Failover

You can migrate your SQL Server 2000 log shipping configuration to SQL Server 2005 without using failover. This procedure allows you to easily upgrade both server instances in your log shipping configuration; however, your primary database is unavailable while you are upgrading the primary server instance to SQL Server 2008.

- 1. Upgrade the secondary server instance to SQL Server 2008. When you upgrade the secondary server instance, the log shipping database remains a SQL Server 2000 database because it is in an offline state. The SQL Server 2000 log shipping jobs that copy and restore log files will no longer function.
- 2. Upgrade the primary server to SQL Server 2008. The primary database is unavailable while the upgrade is in progress.
- 3. Configure log shipping from the primary server instance to the secondary server instance. Be sure to specify the No, the secondary database is initialized option on the Initialize Secondary database tab of the Secondary Database Settings dialog box. For more information, see How to: Enable Log Shipping (SQL Server Management Studio).

#### <sup>Important:</sup>

Specify the same backup share that you used with your SQL Server 2000 log shipping configuration. This ensures that all log backups are properly applied to the secondary database when you enable log shipping in SQL Server 2008.

Because the database upgrade process is a fully logged operation, your secondary database is upgraded to a SQL Server 2008 database when you start shipping logs to the secondary server instance.

# Redeploying Log Shipping

If you do not want to migrate your log shipping configuration, you can redeploy log shipping from scratch by reinitializing your secondary database with a full backup and restore of the primary database. This may be a desirable option if you have a small database or if high availability is not crucial during the upgrade procedure.

#### To enable log shipping

- How to: Enable Log Shipping (SQL Server Management Studio).
- How to: Enable Log Shipping (Transact-SQL).

# Removing SQL Server 2000 Log Shipping Tables and Jobs

SQL Server 2008 does not use any of the log shipping tables used by SQL Server 2000. You can therefore delete these tables after you upgrade your server to SQL Server 2008. These tables are as follows:

- log\_shipping\_databases
- log\_shipping\_monitor
- log\_shipping\_plan\_databases
- log\_shipping\_plan\_history
- log\_shipping\_plans
- log\_shipping\_primaries
- log\_shipping\_secondaries

You can also delete any log shipping SQL Server Agent jobs that were created for the log shipping configuration by SQL Server 2000.

# <sup>□</sup> See Also

#### Concepts

Upgrading SQL Server 2005 Log Shipping to SQL Server 2008

#### **Other Resources**

Working with Transaction Log Backups Log Shipping Administration Log Shipping Tables and Stored Procedures

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Considerations for Upgrading Analysis Services



#### - ▼

<u>Upgrading to SQL Server 2008</u> >

Upgrading is typically performed by SQL Server Setup. SQL Server Setup can be used to upgrade any or all of its components, including Analysis Services.

This topic highlights important considerations that you should be aware of before you upgrade Analysis Services. For step-by-step instructions on how to start Setup and select the components to upgrade, see <u>Upgrading to SQL Server</u> 2008. For information about command-line options for upgrading Analysis Services, see **Installing SQL Server from the Command Prompt**.

# <sup>ℤ</sup>Important:

To learn which Analysis Services features and functionality have been discontinued, deprecated, or changed see <u>SQL Server Analysis Services</u> <u>Backward Compatibility</u>.

# <sup>□</sup> Known Upgrade Issues

Before upgrading to SQL Server 2008 Analysis Services, review <u>Breaking</u> <u>Changes to Analysis Services Features in SQL Server 2008</u>. It is also recommended that you run Upgrade Advisor as described in <u>Using Upgrade</u> <u>Advisor to Prepare for Upgrades</u>.
# <sup>D</sup> Pre-Upgrade Checklist

Before upgrading to SQL Server 2008 Analysis Services, review the following requirements:

- Hardware and Software Requirements for Installing SQL Server 2008
- <u>Check Parameters for the System Configuration Checker</u>
- Security Considerations for a SQL Server Installation
- <u>Using Upgrade Advisor to Prepare for Upgrades</u>

## <sup>©</sup> Upgrading from SQL Server 2005 Analysis Services

You can upgrade an existing instance of SQL Server 2005 Analysis Services to SQL Server 2008 Analysis Services and, as part of the upgrade process, automatically migrate existing databases from the old instance to the new instance. Because the metadata and binary data is compatible between the two versions, you will retain the data after you upgrade and you do not have to manually migrate the data. To upgrade an existing instance of SQL Server 2005 Analysis Services, run Setup and specify the name of the existing instance as the name of the new instance. For more information, see <u>How to: Upgrade an</u> <u>Instance of Analysis Services</u>.

## <sup>D</sup> Upgrading from SQL Server 2000 Analysis Services

When you upgrade from SQL Server 2000 Analysis Services to SQL Server 2008 Analysis Services, because of changes to the underlying architecture, it is best to perform the upgrade in two steps. First, install SQL Server 2008 Analysis Services side-by-side with SQL Server 2000 Analysis Services. Then, use the Analysis Services Migration Wizard to migrate SQL Server 2000 Analysis Services databases to the SQL Server 2008 Analysis Services format. The Analysis Services Migration Wizard migrates SQL Server 2000 Analysis Services and re-creates the database metadata to SQL Server 2008 Analysis Services and re-creates the database structures in the SQL Server 2008 Analysis Services database format while leaving the original databases intact. For more information, see How to: Install Analysis Services Side-by-Side with an Earlier Version and 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 2008 Analysis Services the default instance on the server. For more information, see <u>How to: Rename an Instance of Analysis Services</u>.

To upgrade SQL Server 2000 Analysis Services

- 1. Install a named instance of SQL Server 2008 Analysis Services by using the SQL Server Installation Wizard or an alternative installation method. For more information, see **Considerations for Installing Analysis Services** or **Installing SQL Server from the Command Prompt**.
- 2. Use the Analysis Services Migration Wizard to migrate SQL Server 2000 Analysis Services database metadata to SQL Server 2008 Analysis Services. This step re-creates the database structures in the SQL Server 2008 Analysis Services database format while leaving the original databases intact. For more information, see <u>Migrating Existing Analysis</u> <u>Services Databases</u>.
- 3. Process the new SQL Server 2008 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 the earlier version of Analysis Services if it is no longer needed.

## $\Box$ See Also

## Concepts

Considerations for Installing Analysis Services

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Migrating Existing Analysis Services Databases



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Analysis Services</u> >

You can use the Analysis Services Migration Wizard to upgrade Microsoft SQL Server 2000 Analysis Services databases to Microsoft SQL Server 2008 Analysis Services. During migration, the wizard copies SQL Server 2000 Analysis Services database objects and then re-creates them on an instance of SQL Server 2008 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, you should migrate your databases one at a time, or in small batches. This will allow you to verify that each database object appears as expected 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**.

# <sup>□</sup> 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 2008 Analysis Services, use the format *computername\instancename*.

You can also decide 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 2008 Integration Services 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 2008 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 that uses 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 2008 Analysis Services objects, see <u>Migration</u>

Considerations (Analysis Services).

<sup>I</sup>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 states 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 2008 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 2008 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**.

## $\Box$ See Also

## Concepts

Considerations for Upgrading Analysis Services

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Migration Considerations (Analysis Services)



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Analysis Services</u> >

When you migrate from earlier versions of Microsoft SQL Server Analysis Services to SQL Server 2008 Analysis Services, 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:

- Breaking Changes to Analysis Services Features in SQL Server 2008
- <u>Migrating Existing Analysis Services Databases</u>
- Discontinued Analysis Services Functionality in SQL Server 2008
- Deprecated Analysis Services Functionality in SQL Server 2008

## <sup>□</sup> Analysis Services Migration Considerations

## **Analysis Services Objects**

The following table describes how SQL Server 2000 Analysis Services objects correspond to SQL Server 2008 Analysis Services objects.

SQL Server	SQL Server	
Analysis	Analysis	More information
Services	Services	
object	object	
Aggregation	Aggregation	Only aggregations for hybrid OLAP (HOLAP) and multidimensional OLAP (MOLAP) partitions are migrated.
Analysis	Analysis	
Services	Services	None.
database	database	
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	Database role	

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.
Multiple	Single parent- child hierarchy,	MDX expressions that refer to the original

parent-child hierarchies in a dimension	which is defined by settings on the parent attribute.	hierarchies no longer function. A SQL Server 2008 Analysis Services 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 replaced calculated cells and custom level formulas. Calculated cells are supported in SQL Server 2008 Analysis Services and are migrated from SQL Server 2000 Analysis Services as defined, but custom level formulas are migrated as assignments.

In SQL Server 2008 Analysis Services, 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 were introduced 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 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 Books Online.

### **Schema Rowset Changes**

Many schema rowsets were 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 occurred in SQL Server 2005 Analysis Services (SSAS).

Schema	Column	Change
DBSCHEMA_TABLES_INFO	All	This schema is no longer supported.
MDSCHEMA_CELL_FORMULAS	All	This schema is no longer

		supported.
MDSCHEMA_DIMENSIONS	DIMENSION_CARDINALITY	This column returns the number of dimension key attribute members.
MDSCHEMA_MEMBERS	LEVELS_LIST	This schema column contains null.
MDSCHEMA_MEMBERS	MEMBER_ORDINAL	This column is deprecated. The value 0 is returned instead of the member ordinal.

### **Unique Member Names**

In SQL Server 2005 Analysis Services (SSAS), unique member names were updated and they now differ from those in SQL Server 2000 Analysis Services. Because of restrictions on the unique member names, 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.

## <sup>D</sup> 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 2008 Analysis Services 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 2008 Analysis Services, you can use the Predictive Model Markup Language (PMML). The following example demonstrates how to 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, DMSCHEMA\_MINING\_MODEL\_XML.

You can use the PMML script to create a new mining model in an instance of SQL Server 2008 Analysis Services 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 2008 Analysis Services 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.

## $\Box$ See Also

## Concepts

Considerations for Upgrading Analysis Services

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Upgrading or Migrating Analysis Services How-to Topics



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Analysis Services</u> >

This section contains the following common procedures for upgrading and migrating databases to Microsoft SQL Server 2008 Analysis Services.

For common procedures for installing Analysis Services, see <u>Analysis Services</u> <u>Installation How-to Topics</u>.

## <sup>I</sup> 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 the installation of SQL Server.

# <sup>□</sup> In This Section

- How to: Upgrade an Instance of Analysis Services
  How to: Migrate Analysis Services Databases

## $\Box$ See Also

## Concepts

Considerations for Upgrading Analysis Services

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help How to: Upgrade an Instance of Analysis Services

# 

#### -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Analysis Services</u> > <u>Upgrading or</u> <u>Migrating Analysis Services How-to Topics</u> >

You can upgrade an existing instance of Microsoft SQL Server 2005 Analysis Services to Microsoft SQL Server 2008 Analysis Services and, as part of the upgrade process, migrate existing databases from the old instance to the new instance. To upgrade, run Setup and specify the name of the existing instance as the name of the new instance.

## <sup>I</sup>Important:

The 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.

## <sup>©</sup> To upgrade an instance of Analysis Services

- 1. Run Setup either by using the SQL Server Installation Wizard or at the command prompt. For more information about how to run Setup by using the SQL Server Installation Wizard, see <u>How to: Install Analysis</u> <u>Services by Using Setup</u>. For more information about how to run Setup at the command prompt, see **Installing SQL Server from the Command Prompt**.
- 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 at 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.

## $\Box$ See Also

## Tasks

How to: Migrate Analysis Services Databases

## Concepts

Considerations for Upgrading Analysis Services

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help How to: Migrate Analysis Services Databases



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Analysis Services</u> > <u>Upgrading or</u> <u>Migrating Analysis Services How-to Topics</u> >

You can migrate existing Analysis Services databases 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 SQL Server 2000 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.

### <sup>I</sup> 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 the installation of SQL Server.

# <sup>□</sup> 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 2008 Analysis Services server and instance name. Instead of designating a destination server, you can also decide to save the database schema to a script file and complete the migration later. You can do this by using an Analysis Services Execute DDL task in a SQL Server 2008 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 might want to review <u>Migration Considerations</u> (<u>Analysis Services</u>) to understand the differences between Analysis Services database versions.

## $\Box$ See Also

### Concepts

Migrating Existing Analysis Services Databases Migration Considerations (Analysis Services)

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Considerations for Upgrading Reporting Services



#### Ξ ¥

#### <u>Upgrading to SQL Server 2008</u> >

Upgrade is performed by SQL Server Setup. SQL Server Setup can be used to upgrade any or all SQL Server components, including Reporting Services. To upgrade an earlier version of SQL Server, run the SQL Server 2008 Setup program on a computer that has an earlier version of SQL Server installed. Setup detects the existing instances and prompts you to upgrade.

When you run Setup, you can select the option to upgrade from SQL Server 2000 or SQL Server 2005, or you can install a new instance of SQL Server 2008 Reporting Services that runs side-by-side existing installations.

This topic highlights important considerations that you should be aware of before you upgrade Reporting Services.

### <sup>I</sup> Important:

To learn which Reporting Services features have been discontinued, deprecated, or changed see <u>Reporting Services Backward Compatibility</u>.

# <sup>□</sup> Upgrading Editions and Versions

SQL Server 2008 Setup provides upgrade support for the following earlier editions of Reporting Services:

- SQL Server 2000 Reporting Services installation with Service Pack 2 (SP2).
- SQL Server 2005 Reporting Services.

Upgrading or downgrading from one SQL Server 2008 edition to another is supported.

## <sup>□</sup> Known Upgrade Issues

Before upgrading to SQL Server 2008 Reporting Services, review the Upgrade Advisor documentation and the <u>Online</u> for the latest information about upgrade issues. To obtain the Upgrade Advisor documentation, install Upgrade Advisor. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.

Known upgrade issues currently include the following:

- No upgrade support for a report server that uses a remote SQL Server 2000 Database Engine instance to host the report server database.
- No support for the SQL Server 2000 Report Server Web service in SQL Server 2008. This endpoint is discontinued. If you created or installed custom features that point to the ReportServer2000 endpoint, that custom functionality will no longer run. For more information about new endpoints, see **Report Server Web Service Endpoints**.
- No support for earlier versions of the Reporting Services WMI provider. The Reporting Services WMI provider is not backwards compatible with previous versions. After upgrade, you cannot use the SQL Server 2008 Reporting Services WMI provider with earlier versions of Reporting Services.

For more information about supported upgrade scenarios, see <u>Version and</u> <u>Edition Upgrades</u>. If you cannot run Setup to perform an in-place upgrade of your existing installation, you can install a new instance of SQL Server 2008 Reporting Services and then migrate your report server database and configuration files to the new instance. For more information about migrating Reporting Services, see <u>How to: Migrate a Reporting Services Installation</u>.

# <sup>D</sup> Pre-Upgrade Checklist

Before upgrading to SQL Server 2008 Reporting Services, review the following requirements:

- Review requirements to determine whether your hardware and software can support SQL Server 2008 Reporting Services. For more information, see <u>Hardware and Software Requirements for Installing SQL Server 2008</u>.
- Use System Configuration Checker (SCC) to scan the report server computer for any conditions that might prevent a successful installation of SQL Server 2008. For more information, see <u>Check Parameters for the System Configuration Checker</u>.
- Review security best practices and guidance for SQL Server. For more information, see <u>Security Considerations for a SQL Server Installation</u>.
- Run the Upgrade Advisor tool on the report server computer to determine any issues that might prevent you from successfully upgrading. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.
- Back up your symmetric key. For more information, see <u>Backing Up and</u> <u>Restoring Encryption Keys</u>.
- Back up your report server databases. For more information, see **Moving the Report Server Databases to Another Computer**.
- Back up the following report server configuration files: Rsreportserver.config, Rswebapplication.config, Rssvrpolicy.config, Rsmgrpolicy.config, Reportingservicesservice.exe.config, Web.config (for both the report server and Report Manager ASP.NET applications), and Machine.config (for ASP.NET if you modified it for report server operations).
- Back up any customizations to existing Reporting Services virtual directories in IIS.

Before you upgrade a production environment, always run a test upgrade in a pre-production environment that has the same configuration as your production environment.

# <sup>□</sup> How to Upgrade an Earlier Version of Reporting Services

Setup provides upgrade options that you can specify as a command-line argument or in the Setup wizard. Choose from the following topics for instructions on how to upgrade your installation.

How to: Upgrade to SQL Server 2008 (Setup)

Explains how to upgrade to SQL Server 2008 using the Installation Wizard.

How to: Install SQL Server 2008 from the Command Prompt

Explains how to upgrade to SQL Server 2008 as a command-line operation.

## Removing Files and Applications from the Previous Installation

Upgrade will not delete old log files, the obsolete RSWebApplication.config file, or virtual directory settings in IIS. Upgrade will not remove SQL Server 2005 Report Designer, Management Studio, or other client tools. If you no longer require them, be sure to remove these files and tools after upgrade is finished.

# <sup>D</sup> Upgrade Workflow

The upgrade process begins when you select an existing report server instance to upgrade.

- 1. If the report server database is on a remote computer and you do not have permission to update that database, Setup prompts you to provide credentials to update to a remote report server database. Be sure to provide credentials that have **sysadmin** or database update permissions.
- 2. Setup checks for conditions or settings that prevent upgrade and reads configuration settings. Examples include custom extensions deployed on the report server. If upgrade is blocked, you must either modify your installation so that upgrade is no longer blocked, or migrate to a new SQL Server 2008 instance. For more information, see the Upgrade Advisor documentation.
- 3. If upgrade can proceed, Setup prompts you to continue with the upgrade process.
- 4. Setup creates new folders for SQL Server 2008 program files. The program folders for a Reporting Services installation include MSRS10.<*instance name*>.
- 5. Setup adds the SQL Server 2008 report server program files, configuration tools, and command line utilities that are part of the report server feature.
  - 1. Program files from the previous version are removed.
  - 2. Report server configuration tools and utilities that are upgraded to the new version include the Reporting Services Configuration tool, command line utilities, and Report Builder. New tools installed with SQL Server 2008 include the new Report Designer Preview.
  - 3. Other client tools such as Business Intelligence Development Studio, SQL Server Management Studio, and Books Online are not upgraded. To obtain new versions of the tools, you can add them when you run Setup. Earlier versions will co-exist alongside SQL Server 2008 versions. If you installed samples, the earlier version will remain. Setup does not support upgrade for the SQL Server samples.
- 6. Setup reuses the service entry in Service Control Manager for the SQL Server 2008 Report Server service. This service entry includes the Report Server Windows service account.
- 7. Setup reserves new URLs based on existing virtual directory settings in IIS.

Setup might not remove virtual directories in IIS, so be sure to remove those manually after upgrade is finished.

- 8. Setup upgrades the report server databases to the new schema and modifies the **RSExecRole** by adding Database Owner permissions to the role. This step only occurs when you are upgrading from SQL Server 2000 Reporting Services or SQL Server 2005 Reporting Services prior to SP1.
- 9. Setup merges settings in the configuration files. Using the configuration files from the current installation as the basis, new entries are added. Obsolete entries are not removed, but they will no longer be read by the report server after upgrade is finished.
## $\Box$ See Also

## Concepts

Considerations for Installing Reporting Services Upgrading to SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help How to: Detect Version Information (Reporting Services)

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Reporting Services</u> >

Reporting Services consists of multiple applications, tools, and components. If you install or upgrade components in a single operation, all parts of an installation will be the same version. However, if you run multiple instances side-by-side, configure server components to run on different computers, or publish items to a central server, you might encounter version compatibility errors if you try to use client tools from one version with a report server that is a different version. Such errors are most likely to occur during publish operations or if you try to use the Reporting Services Configuration tool to connect to a report server from a previous release.

This topic can help you identify version or edition information so that you can determine whether to upgrade a component or use a different installation of a client or server instance. For more information about upgrading, see **Upgrade** (**Reporting Services**).

Version compatibility errors are unlikely to occur for report server database connections. The report server can automatically upgrade a report server database to which it is connected, thereby eliminating a major cause of the database version mismatch error that sometimes occurred in earlier versions. However, you might encounter an error in a scale-out deployment if you have not upgraded all of the nodes.

Report server version and edition information is most easily viewed in the **Report Server Status** page of the Reporting Services Configuration tool. However, if you cannot connect to the report server because of version incompatibility errors, you can use the alternate steps provided later in this topic.

## <sup> </</sup> Note:

You can retrieve version information programmatically. To retrieve server version information, use **ServerInfoHeaderValue**. To retrieve the database version, use **GetDatabaseVersionDisplayName Method (WMI)**.

## <sup>□</sup> To find version information for Reporting Services Configuration tool

• Start the Reporting Services Configuration tool. On the **Report Server Status** page, view the **Product Version** value and compare it to the Version Information table below in this topic.

# <sup>©</sup> To find version and edition information for a report server instance

- 1. Open the Report Server service trace log file in a text editor. The file name is ReportServerService\_<timestamp>.log and it is located in the \Reporting Services\ReportServer\logfiles folder.
- 2. Report Server version information is in **<Product>** at the top of the file.
- 3. Edition information for the report server can be found in the **INFO** statements further down in the file:
  - **Reporting Services starting SKU** shows the edition of the report server instance.

An alternative approach for viewing report server version information is to type the report server URL in the Address bar of a browser window. Version information can be found at the bottom of the page in this format: <release >: <buildnumber>. If the release is 10, you are connected to a SQL Server 2008 report server instance.

# <sup>□</sup> To find edition information for a database instance that hosts report server databases

- 1. Open the Report Server service trace log file in a text editor. The file name is ReportServerService\_<timestamp>.log and it is located in the \Reporting Services\ReportServer\logfiles folder.
- 2. Edition information for the database server that hosts report server database can be found in the **INFO** statements further down in the file:
  - **Catalog SQL Server Edition** shows the edition of the database server that hosts the report server database.

Version information is not available in log files, tools, or through the programmatic interfaces. Use the table at the end of this topic to obtain version information for the report server database.

#### <sup>©</sup> To find version for Report Designer and Model Designer

- 1. Start Business Intelligence Development Studio, click **Help**, and then click **About Visual Studio**.
- 2. In the list of installed products, click **Reporting Services**. The version information appears in the product details section of the dialog box.

## <sup>©</sup> To find version for Report Manager or Report Builder

• In SQL Server 2008, Report Manager runs in the Report Server service. If you know the version of the Report Serve service, you also know the version of Report Manager. However, if you point a Report Manager instance to a different Report Service instance, it is possible that the version of Report Manager and the report server could be out of sync. To verify the version of Report Manager, right-click a Web page in Report Manager and use the **View Source** command to view the HTML. The version information is embedded in the underlying HTML.

• To find version information for Report Builder, start Report Builder and click **About Report Builder** on the **Help** menu.

# <sup>□</sup> Version Information

The following table provides a list of the version numbers for the report server and report server database released to date:

Release	Report server	Report server database
SQL Server 2008	10.n.nnnn.nn	n/a
SQL Server 2005 Reporting Services Service Pack 2 (SP2)	9.0.3042.00	C.0.8.45
SQL Server 2005 Express Edition with Advanced Services	9.0.3042.00	C.0.8.45
SQL Server 2005 Reporting Services Service Pack 1 (SP1)	9.0.2049.00	C.0.8.43
SQL Server 2005 Reporting Services	9.0.1399.01	C.0.8.40
SQL Server 2000 Reporting Services Service Pack 2 (SP2)	8.00.1038	C.0.6.54
SQL Server 2000 Reporting Services Service Pack 1 (SP1)	8.00.0878	C.0.6.51
SQL Server 2000 Reporting Services	8.00.0743	C.0.6.43

## □ See Also

**Concepts** <u>Upgrading a Report Server Database</u>

**Other Resources** 

Tutorial: How to Locate and Start Reporting Services Tools Upgrade (Reporting Services) Reporting Services Components and Tools

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Upgrading a Report Server Database



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Reporting Services</u> >

The report server database provides internal storage for one or more report server instances. Because the report server database schema can change with each new release of Reporting Services, it is required that the database version match the version of the report server instance you are using. In most cases, a report server database can be upgraded automatically with no specific action on your part.

## <sup>I</sup>Note:

You can upgrade a report server database that runs on a SQL Server 2005 Database Engine instance without having to upgrade the Database Engine to SQL Server 2008. You can also use a SQL Server 2005 Database Engine instance to host a new report server database that you create for a SQL Server 2008 Reporting Services installation.

# <sup>©</sup> Ways to Upgrade a Report Server Database

The following list identifies all of the conditions under which a report server database is upgraded:

- Setup upgrades a single instance of a report server. Then the database schema is automatically upgraded after service startup and the report server determines that the database schema version does not match the server version.
- At service startup, the report server checks the database schema version to verify that it matches the server version. If the database schema version is an older version, it is automatically upgraded to the schema version that is required by the report server. Automatic upgrade is especially useful if you restored or attached an older report server database. A message is entered in the report server trace log file indicating that the database schema version was upgraded.
- The Reporting Services Configuration tool upgrades a local or remote report server database when you select an older version to use with a newer report server instance. In this case, you must confirm the upgrade action before it happens.

The Reporting Services Configuration tool no longer provides a separate Upgrade button or upgrade script. Those features are obsolete in SQL Server 2008 due to the automatic upgrade feature of the Report Server service.

After the schema is updated, you cannot rollback the upgrade to an earlier version. Always backup the report server database in case you need to recreate a previous installation.

# <sup>□</sup> How the Schema, Metadata, and Report Server Content is Updated

The report server database is upgraded in three stages:

- 1. The schema is upgraded automatically after setup and service startup, or when you select a SQL Server 2005 or 2000 report server database in the Reporting Services Configuration tool. In addition, the Report Server service checks the database version at startup. If the report server is connected to a database that is an earlier version, the report server will update the database during startup.
- 2. Security descriptors 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 <u>Upgrading Reports</u>.

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.

# <sup>D</sup> Upgrading a Report Server Database

If you are upgrading a Reporting Services installation that includes a 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 SQL Server **sysadmin** permissions on the database server that hosts the report server databases, 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 **sysadmin** permission to modify the schema on the remote computer, the database upgrade will succeed.

However, if you do not have permission to on the database 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, re-select the database, and click Apply. The database will be upgraded automatically."

At this point, 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 until you finish the upgrade process by upgrading the database manually.

#### 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 <u>How to: Start Reporting Services</u>

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 **Apply** 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

<u>Creating a Report Server Database</u> <u>Considerations for Upgrading Reporting Services</u> <u>How to: Migrate a Reporting Services Installation</u>

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Upgrading Reports



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Reporting Services</u> >

SQL Server 2008 introduces changes to the Report Definition Language (RDL), the report object model, and the rendering object model that affect reports created in earlier versions of the software. When you upgrade a SQL Server 2005 Reporting Services installation to a SQL Server 2008 Reporting Services installation, existing reports and snapshots that have been uploaded to a report server are automatically upgraded to the new schema the first time they are processed. If a report cannot be automatically upgraded, the report is processed using the backward-compatibility mode.

Existing report definition (.rdl) files are automatically upgraded when you open them in Report Designer in Business Intelligence Development Studio.

#### <sup>I</sup>Note:

For more information about the new RDL features, such as the Tablix data region or the revised Chart component, see **What's New (Reporting Services)**.

After you upgrade a report, you might notice additional errors, warnings, and messages. This is the result of changes to the internal report object model and processing components, which cause messages to appear when underlying problems in the report are detected. For more information, see <u>Reporting</u> <u>Services Backward Compatibility</u>.

# <sup>□</sup> Versions Supported by Upgrade

Reports that were created in any previous version of Reporting Services can be upgraded. This includes the following versions:

- SQL Server 2000
- SQL Server 2000 with Service Pack 1
- SQL Server 2000 with Service Pack 2
- SQL Server 2005
- SQL Server 2005 with Service Pack 1
- SQL Server 2005 with Service Pack 2

# <sup>©</sup> Report Definition (.rdl) Files and Report Designer

A report definition file includes a reference to the RDL namespace that specifies the version of the report definition schema that is used to validate the rdl file.

When you open an .rdl file in Report Designer in Business Intelligence Development Studio, if the report was created for a previous namespace, Report Designer automatically creates a backup file and upgrades the report to the current namespace. If you save the upgraded report definition, you have saved the converted .rdl file. As soon as you save it, you cannot open it in earlier versions of Report Designer. This is the only way you can upgrade a report definition file.

You can upload an .rdl file created in an earlier version of Reporting Services to a SQL Server 2008 report server and it is automatically upgraded on first use. The report server stores the report definition file in the original format. The report is automatically upgraded the first time it is viewed, but the stored report definition file remains unchanged.

#### <sup>I</sup>Note:

You cannot publish or upload a report that has the SQL Server 2008 report definition namespace to a SQL Server 2000 report server or SQL Server 2005 report server.

To identify the current RDL schema for a report, for a report server, or for Report Designer, see **How to: Identify the Report Schema Version**.

## <sup>©</sup> Published Reports and Report Snapshots

On first use, the report server tries to upgrade existing published reports and report snapshots to the new report definition schema, requiring no specific action on your part. When a user views a report or a report snapshot, or when the report server processes a subscription, the upgrading attempt occurs. The report definition is not replaced but continues to be stored on the SQL Server 2008 report server in its original schema. If a report cannot be upgraded, the report runs in backward-compatibility mode.

# <sup>B</sup> Backward-Compatibility Mode

A report that is successfully upgraded is processed by the SQL Server 2008 Reporting Services report processor. A report that cannot be upgraded is processed by the SQL Server 2005 Reporting Services report processor in backward-compatibility mode. A report cannot be processed by both report processors.

On first use, a report is either successfully upgraded or marked for backward compatibility.

Only the SQL Server 2008 Reporting Services report processor supports new features such as on-demand paging, Tablix and Gauge data regions, and enhanced Chart data regions. If a report cannot be upgraded, you can still view the rendered report but the new features are not available. To take advantage of the new features, a report must be successfully upgraded.

# <sup>©</sup> Upgrading a Report with Subreports

When a report contains subreports, one of four possible states can occur during upgrade:

- The main report and all subreports can be successfully upgraded. They are processed by the SQL Server 2008 Reporting Services report processor.
- The main report and all subreports cannot be upgraded. They are processed by the SQL Server 2005 Reporting Services report processor.
- The main report can be upgraded but one or more subreports cannot be upgraded. The main report is processed by the SQL Server 2008 Reporting Services report processor, but the rendered report shows the message "Error: Subreport could not be processed" in the location where the subreport that could not be upgraded would appear.
- The main report cannot be upgraded but one or more subreports can be upgraded. The main report is processed by the SQL Server 2008 Reporting Services report processor, but the rendered report shows the message "Error: Subreport could not be processed" in the location where the subreport would appear.

If you see the error "Error: Subreport could not be processed", you must change the definition of the main report or the subreport so that the reports can be processed by the same version of the report processor.

Drillthrough reports do not have this limitation because they are processed as independent reports.

# <sup>©</sup> Upgrading a Report with Custom Report Items

SQL Server 2005 Reporting Services reports might contain custom report items (CRIs) provided by third-party software vendors and installed by the system administrator on the report authoring computer and the report server. Reports that contain CRIs can be upgraded in the following ways:

- A SQL Server 2005 Reporting Services report server is upgraded to a SQL Server 2008 Reporting Services report server. Published reports on the report server are automatically upgraded on first use.
- A SQL Server 2005 Reporting Services report is uploaded to a SQL Server 2008 Reporting Services report server. The report is automatically upgraded on first use.
- A SQL Server 2005 Reporting Services report is opened in Report Designer in Business Intelligence Development Studio. A backup copy of the original report is created. One of the following two cases occurs:
  - 1. All CRIs in the report have no unsupported features. The CRIs are converted to report items in the new report definition schema, so the entire report is upgraded. If you save the file, it is saved in the current RDL namespace.
  - 2. One or more CRIs in the report have unsupported features. A dialog box prompts the user whether to convert the CRIs are leave them unchanged.

For more information, see <u>Opening a Report in Report Designer</u> later in this topic.

For information about identifying the current RDL namespace for a report server, BI Development Studio, or a report, see **How to: Identify the Report Schema Version**.

#### **Upgrading Reports on a Report Server**

The first time a SQL Server 2005 Reporting Services report runs on a report server that has been upgraded to a SQL Server 2008 Reporting Services report server, the report is automatically upgraded to the current report definition namespace supported by the report server. The report could have existed on the report server before the upgrade, or the report could have been uploaded via Report Manager or published to the report server from Report Designer in SQL Server 2005 BI Development Studio.

The following table lists the upgrade action that is performed by the report server for specific types of CRIs in a report.

CRI type	Report Server upgrade action	
	Upgrade not performed.	
Third-party CRIs	Processed by the SQL Server 2005 Reporting Services report processor.	
Dundas 2005 Chart CRI with no unsupported features	Upgraded to the latest RDL schema. All Dundas 2005 Chart CRIs are converted to Sql Server 2008 Reporting Services chart data regions.	
	Processed by the SQL Server 2008 Reporting Services report processor.	
Dundas 2005 Gauge CRI with no unsupported features	Upgraded to the latest RDL schema. All Dundas 2005 Gauge CRIs are converted to Sql Server 2008 Reporting Services gauge data regions.	
	Processed by the SQL Server 2008 Reporting Services report processor.	
Dundas 2005 Chart CRI with unsupported features	Upgrade not performed.	
	Processed by the SQL Server 2005 Reporting Services report processor.	
Dundas 2005 Gauge CRI with unsupported features	Upgrade not performed.	
	Processed by the SQL Server 2005 Reporting Services report processor.	

#### **Opening a Report with CRIs in Report Designer**

When you open a SQL Server 2005 Reporting Services report with CRIs in Report Designer in SQL Server 2008 Business Intelligence Development Studio, the report will be upgraded to the new report definition schema. Depending on the CRIs contained in the report, one of the following actions will take place:

- Third-party CRIs detected. If the version of the CRI that is installed on the report authoring computer is not compatible with the new RDL schema, the design surface shows a text box with a red X. You must contact your system administrator to install new versions of the CRI from third-party vendors that are compatible with the new RDL schema.
- Dundas 2005 Chart or Gauge CRIs detected and all instances contain supported functionality. All Dundas 2005 Chart and Gauge CRIs are converted to the Reporting Services Chart and Gauge report items that you see on the Toolbox. These are known as native chart and gauge report items.
- Dundas 2005 Chart or Gauge CRIs are detected and any instance has unsupported functionality. Unsupported functionality is described after this section. You can choose whether to convert all CRIs to native report items.
  - If you convert them, the report is upgraded to the new RDL schema and Dundas 2005 Chart and Gauge CRIs are converted to the corresponding native Chart and Gauge report items, but unsupported functionality is removed. In the rendered report, you might see differences in the way the CRI displays.
  - If you choose not to convert them, the report is upgraded to the new RDL schema but the CRIs are treated as third-party CRIs. You must work with the system administrator and the third-party vendors to install new CRIs that are compatible with the new report schema. If the new CRIs are not available, the report displays a text box with a red X in Report Designer.

Saving a report after it is upgraded in the report authoring environment is the only way to upgrade an existing report to the new report definition schema.

#### Unsupported Dundas 2005 Chart Custom Report Item Functionality

Unsupported functionality for Dundas 2005 Chart CRI includes the following features:

- Annotations.
- Custom legend items.
- Custom attributes with the following names:
  - CUSTOM\_CODE\_CS

- CUSTOM\_CODE\_VB
- CUSTOM\_CODE\_COMPILED\_ASSEMBLY For example, if your .rdl file contains the following section, you will need to remove it before upgrading:

Copy Code

```
<CustomProperty>
<Name>CUSTOM_CODE_CS</Name>
<Value>dXNpWERwegfdfgiobxxl3bmc... </Value>
</CustomProperty>
```

#### **Unsupported Dundas 2005 Gauge Custom Report Item Functionality**

Unsupported functionality for Dundas 2005 Gauge CRI includes the following features:

- Numeric indicators.
- State indicators.
- Custom Images.

## □ See Also

#### Concepts

Considerations for Upgrading Reporting Services Breaking Changes in SQL Server 2008 Reporting Services Behavior Changes in SQL Server 2008 Reporting Services Discontinued Functionality in SQL Server 2008 Reporting Services Upgrading a Report Server Database

#### **Other Resources**

Convert CRI Dialog Box Custom Report Items Threat and Vulnerability Mitigation (Reporting Services)

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Considerations for Upgrading Data Transformation Services

# 

#### - -

#### <u>Upgrading to SQL Server 2008</u> >

If SQL Server 2000 Data Transformation Services (DTS) is currently installed on your computer, you can upgrade from Data Transformation Services to SQL Server 2008 Integration Services without any software conflicts. There are no software conflicts because Integration Services is a completely new product and not a version upgrade from Data Transformation Services.

## <sup>ℤ</sup> Note:

If you want to upgrade from SQL Server 2005 Integration Services to SQL Server 2008 Integration Services, see **Considerations for Upgrading Integration Services**.

# <sup>B</sup> Before Upgrading Data Transformation Services

Before upgrading Data Transformation Services to Integration Services, we recommend that you review <u>Support for Data Transformation Services (DTS) in</u> <u>SQL Server 2008</u>.

We also recommended that you install and run Upgrade Advisor before you perform an upgrade. For more information, see <u>Using Upgrade Advisor to</u> <u>Prepare for Upgrades</u>.

# <sup>D</sup> Upgrading Data Transformation Services

You can upgrade an existing instance of SQL Server 2000 by using one of the following methods:

- Run SQL Server 2008 Setup and select the option to **Upgrade from SQL** Server 2000 or SQL Server 2005.
- Run setup.exe at the command prompt and specify the /ACTION=upgrade option. For more information, see the section, "Installation Scripts for Integration Services," in <u>How to: Install SQL</u> Server 2008 from the Command Prompt.

It is not possible to use SQL Server 2008 Setup to upgrade directly from SQL Server 6.5 or 7.0 to SQL Server 2008.

When you upgrade, you can upgrade both DTS and the Database Engine, just upgrade DTS, or just upgrade the Database Engine. If you upgrade only DTS or only the Database Engine, there are some limitations:

- If you upgrade only DTS, SQL Server 2008 Integration Services is fully functional. However, Integration Services will only be able to store packages in the file system, unless an instance of the SQL Server 2008 Database Engine is available on another computer. If an instance of the SQL Server 2008 Database Engine is available, Integration Services will be able to store packages in that instance.
- If you upgrade only the Database Engine, DTS remains functional. However, you will not have the functionality of Integration Services.

# <sup>D</sup> Upgrading Both Data Transformation Services and the Database Engine to SQL Server 2008

This section describes the effects of performing an upgrade that has the following criteria:

- You upgrade both DTS and an instance of the Database Engine to SQL Server 2008.
- Both DTS and the instance of the Database Engine to be upgraded are on the same computer.

#### What the Upgrade Process Does

The upgrade process does the following tasks:

- Upgrades the instance of the Database Engine to SQL Server 2008.
- Installs SQL Server 2008 Integration Services.
- Installs updated run-time support for DTS packages. This support includes the DTS runtime and DTS package enumeration in SQL Server Management Studio. DTS packages can continue to run alongside Integration Services packages on the same computer. For more information, see <u>How to: Install Support for Data Transformation Services Packages</u>.
- Leaves existing DTS packages intact in the existing DTS system tables in the **msdb** database.

#### What the Upgrade Process Does Not Do

The upgrade process does not do the following tasks:

• Does not migrate existing DTS packages to the format that Integration Services packages use. After upgrading, you can optionally use the DTS Package Migration Wizard to migrate DTS packages to the Integration Services format. During migration, the wizard copies DTS packages and then re-creates them in the Integration Services format. The source packages are left intact and are not modified. When you install Integration Services, Setup also installs the DTS Package Migration Wizard.

The DTS Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server.

For more information, see the following topics:

- <u>Migrating Data Transformation Services Packages</u>
- <u>How to: Migrate Data Transformation Services Packages to Integration</u> <u>Services</u>
- <u>Known DTS Package Migration Issues</u>
- Does not install the optional run-time, management, and design-time support for DTS packages. For more information, see <u>How to: Install</u> <u>Support for Data Transformation Services Packages</u>.

## What You Can Do After Upgrading

After the upgrade process finishes, you can do the following task:

• Install optional run-time and management support for DTS. For more information, see <u>How to: Install Support for Data Transformation Services</u> <u>Packages</u>.

After installing the optional run-time support for DTS, you can do the following tasks:

- To view DTS packages under the **Management\Legacy\Data Transformation Services** node in SQL Server Management Studio, connect to the upgraded instance of the Database Engine. (When you connect to the Integration Services service, you can only view Integration Services packages.)
- To run DTS packages within an Integration Services package, use the Execute DTS 2000 Package task. For more information, see **Execute DTS 2000 Package Task**.
- To run DTS packages at the command prompt, use the dtsrun.exe utility.
- To run DTS packages from SQL Server Agent jobs, call the dtsrun.exe utility from a job step.

For more information about how to run packages, see **Running Data** 

Transformation Services (DTS) Packages.

## □ See Also

#### Concepts

Support for Data Transformation Services (DTS) in SQL Server 2008

#### **Other Resources**

Interoperability and Coexistence (Integration Services)

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Upgrading Data Transformation Services How-to Topics

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> >

This section contains the following common procedures that describe how to upgrade from Microsoft SQL Server 2000 Data Transformation Services (DTS) to SQL Server Integration Services:

How to: Upgrade from Data Transformation Services to Integration
 Services

For information about how to install Integration Services, see <u>Considerations for</u> <u>Installing Integration Services</u>.

For information about how to ensure support for DTS when you upgrade to Integration Services, see <u>How to: Install Support for Data Transformation</u> <u>Services Packages</u>.

## □ See Also

#### Concepts

Support for Data Transformation Services (DTS) in SQL Server 2008

#### **Other Resources**

Migrating Data Transformation Services Packages

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Upgrade from Data Transformation Services to Integration Services



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Upgrading Data Transformation Services How-to Topics</u> >

The following procedure describes how to use Setup to upgrade an installation of SQL Server 2000 Data Transformation Services (DTS) to Integration Services. For more information, see <u>Considerations for Upgrading Data Transformation</u> <u>Services</u>.

SQL Server 2008 does not install run-time support for DTS packages. You have to install this run-time support before you can run DTS packages. For information about how to install support for DTS, see <u>How to: Install Support</u> for Data Transformation Services Packages. For more detailed information about support for DTS in SQL Server 2008, see <u>Support for Data Transformation</u> <u>Services (DTS) in SQL Server 2008</u>.

After upgrading, you can optionally use the DTS Package Migration Wizard to migrate DTS packages to the Integration Services format. During migration, the wizard copies 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 <u>Migrating Data Transformation Services Packages</u>.

#### <sup>ℤ</sup> Note:

The DTS Package Migration Wizard is available in the following editions of SQL Server: Standard, Enterprise, and Developer.

For more information about how to upgrade, see <u>How to: Upgrade to SQL</u> <u>Server 2008 (Setup)</u>.

<sup>©</sup> To upgrade from Data Transformation Services to Integration

#### Services

• Run SQL Server 2008 Setup, and then select **Upgrade from SQL Server 2000 or SQL Server 2005**.

—or—

• At the command prompt, run **setup.exe** and specify the /ACTION=upgrade option.

For more information about how to install at the command prompt, see the section, "Installation Scripts for Integration Services," in <u>How to:</u> <u>Install SQL Server 2008 from the Command Prompt</u>.

## □ See Also

#### Tasks

How to: Install Support for Data Transformation Services Packages

## Concepts

Support for Data Transformation Services (DTS) in SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.
### SQL Server Setup Help Migrating Data Transformation Services Packages



#### -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> >

Integration Services 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 Integration Services format, continue to run DTS packages by using the DTS runtime, or incorporate DTS packages into Integration Services solutions by using the Execute DTS 2000 Package task.

### <sup>☑</sup> Note:

For information on upgrading packages that were created in an earlier version of Integration Services, see <u>Upgrading Integration Services</u> <u>Packages</u>.

When you migrate DTS packages to the Integration Services format, the structure of the migrated package and the completeness of the migration depend on the contents of the original package. 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 Integration Services tasks and features.
- Packages contain DTS tasks and features that cannot be migrated directly to Integration Services tasks and features, but package functionality is preserved through the use of encapsulated DTS packages.
- 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.



The Package Migration Wizard leaves the original DTS packages intact and unchanged.

If all the tasks in the package can be mapped to new tasks in Integration Services, the structure of the migrated package is very similar to the structure of the DTS package. Conversely, if tasks do not map directly to Integration Services tasks, the migration changes the package structure. Each non-migrated task retains its DTS format and is encapsulated within an Execute DTS 2000 task. Custom tasks in packages are also migrated this way. For more information, see <u>Migrating Tasks</u>.

Other package elements such as precedence constraints, connections, and variables are migrated to the equivalent element in Integration Services. For more information, see <u>Migrating Precedence Constraints</u>, <u>Migrating Connections</u>, and <u>Migrating Variables</u>.

Package passwords are not migrated. However, Integration Services supports DTS package passwords in DTS packages and within the Execute DTS 2000 Package task. An Execute DTS 2000 Package task within an Integration Services package can pass the package password to a DTS package. For more information, see <u>Migrating Passwords</u>.

The ActiveX Script task in Integration Services is frequently able to run the script code that is migrated from the ActiveX scripts in DTS packages without additional revision. However, references to DTS package objects are not supported. For more information, see <u>Migrating Scripts</u>.

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.

### <sup>ℤ</sup>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 directly in BI Development Studio.

For general information about support for DTS in Integration Services, see <u>Support for Data Transformation Services (DTS) in SQL Server 2008</u>.

# □ In This Section

#### Using the DTS Package Migration Wizard

Describes the migration process that the Package Migration Wizard provides.

#### Known DTS 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.
<u>Migrating Scripts</u>

Describes how the Package Migration Wizard migrates ActiveX scripts.

### □ See Also

### Tasks

How to: Install Support for Data Transformation Services Packages

#### Concepts

Integration Services Backward Compatibility

#### **Other Resources**

**SQL Server Integration Services** 

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Using the DTS Package Migration Wizard



#### -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating Data Transformation Services Packages</u> >

Integration Services provides the DTS Package Migration Wizard for migrating SQL Server 2000 Data Transformation Services (DTS) packages that were created by using the DTS tools and object model.

### <sup>I</sup>∕ Note:

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server.

#### <sup>ℤ</sup>Note:

For information on upgrading packages that were created in an earlier version of Integration Services, see <u>Upgrading Integration Services</u> <u>Packages</u>.

### <sup> </</sup>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 DTS 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 <u>Migrating Tasks</u> and <u>Known DTS Package</u> <u>Migration Issues</u>.

Packages left in the DTS package format can be used in Integration Services solutions and can also continue to run as stand-alone DTS packages.

Many DTS 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 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.

# <sup>□</sup> 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 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\100\DTS\Binn** folder.

## <sup>□</sup> Select the Package Source

The Package Migration Wizard can migrate DTS 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.

#### <sup>I</sup> Important:

SQL Server 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 (DTS Package Migration Wizard)**.

# <sup>□</sup> Select the Package Destination

The Package Migration Wizard can migrate packages either to an **msdb** database in an instance of SQL Server, 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, 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 (DTS Package Migration Wizard)**.

### <sup>I</sup>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.

# <sup>□</sup> 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 (DTS Package Migration Wizard).

### <sup>B</sup> 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.

# <sup>□</sup> 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 (DTS Package Migration Wizard)**.

### <sup>□</sup> 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 (DTS Package Migration Wizard)** and **Migrating the Packages (DTS Package Migration Wizard)**.

# <sup>□</sup> 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.

### <sup>I</sup>∕ 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.

### $\Box$ See Also

### Tasks

How to: Install Support for Data Transformation Services Packages

#### **Other Resources**

Migrating Data Transformation Services Packages

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Known DTS Package Migration Issues



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating Data Transformation Services Packages</u> >

This is a summary of the issues that you may experience when migrating Microsoft SQL Server 2000 Data Transformation Services (DTS) packages to the Integration Services format by using the Package Migration Wizard.

The Package Migration Wizard migrates certain SQL Server 2000 DTS features that have no direct equivalent in Integration Services 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.

For general information about support for DTS in Integration Services, see <u>Support for Data Transformation Services (DTS) in SQL Server 2008</u>.

# <sup>□</sup> 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 Upgrade Advisor. The Package Migration Wizard and Upgrade Advisor identify and report the same potential package migration issues.

Issue Title	Issue Description
	The Integration Services 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
Reconfigure	restore former package behavior.
transaction	
settings after	<b>Corrective action</b> : After migration, re-create the desired
package migration	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 <b>Incorporating</b>
	Transactions in Packages.
	The Integration Services Package Migration Wizard does
Replace	not migrate ActiveX scripts associated with steps in SQL
functionality of	Server 2000 Data Transformation Services (DTS) packages.
ActiveX script	After migration, you must manually edit the resulting SSIS
attached to	package to restore former package behavior.
package steps	<b>Corrective action</b> : After migration, replace the missing
after package	script with a new Script task or other Integration Services
migration	objects as appropriate For more information see Script
	Task.
	The Integration Services Package Migration Wizard does
	not migrate the Analysis Services task in SOL Server 2000
	Data Transformation Services (DTS) packages. The
functionality of	migrated package will contain encapsulated DTS
Analysis Services	functionality which must eventually be replaced.
in Execute DTS	
III Execute D15	<b>Corrective action</b> : After migration, convert the Execute

2000 Package task	DTS 2000 Task that encapsulates the Analysis Services task
after package	to an Integration Services Analysis Services Processing
migration	task. For more information, see <b>Analysis Services</b>
0	Processing Task.
	The Integration Services Dackage Migration Wizard does
Penlace	not migrate complex Data Transformation tasks in SOI
functionality of	Sorver 2000 Data Transformation Sorvices (DTS) packages
complex Data	Server 2000 Data Halisiofiliation Services (D15) packages.
Complex Data	fine fingrated package will contain encapsulated D15
task onconsulated	functionanty which must eventually be replaced.
in Execute DTS	<b>Corrective action</b> : After migration rewrite the complex
III EXECULE DI 5 2000 Dackage tack	Data Transformation task by using Integration Services data
2000 Package lask	flow courses transformations and destinations. For more
alter package	information son <b>Creating Dackage Data Flow</b> and <b>Data</b>
IIIgration	Elow Eloments
	Flow Elements.
	The Integration Services Package Migration wizard does
	not migrate custom tasks in SQL Server 2000 Data
Donlago	Iransformation Services (D1S) packages. The migrated
Keplace	package will contain encapsulated D1S functionality which
runctionality of	must eventually be replaced.
custom task	<b>Corrective action</b> : After migration, re-create the logic of
Encapsulated III	the custom task by using a Script task or Script component
Execute D15 2000	or a sustom component if the desired functionality is not
Package lask after	or a custom component, if the desired functionality is not
package migration	combination of components. For information about
	combination of components. For information about
	(Integration Services)
	The Integration Services Deckage Migration Wigard door
	not migrate Data Driven Querry tacks in SOL Server 2000
	Data Transformation Services (DTS) packages. The
Replace	migrated package will contain encapsulated DTS
functionality of	functionality that must eventually be replaced
Data Driven	functionality that must eventually be replaced.
Query task	<b>Corrective action</b> : After migration, re-create the
encapsulated in	functionality of the Data Driven Query task by using
Execute DTS 2000	Integration Services data flow components. Consider the
Package task after	Slowly Changing Dimension Transformation the
package migration	Slowly Changing Dimension Hanstor mation, the

	Conditional Split Transformation, or the OLE DB
	Command Transformation when replacing the
	functionality of Data Driven Query tasks.
Replace functionality of Data Mining Prediction Query task encapsulated in Execute DTS	The Integration Services 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 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.
2000 Package task after package migration	<b>Corrective action</b> : 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 <b>Data Mining Query Task</b> and <b>Data Mining Query Transformation</b> .
	The Integration Services Package Migration Wizard does
Replace functionality of	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.
Dynamic	<b>Corrective action</b> : After migration, re-create the
Properties task after package migration	functionality of the Dynamic Properties task by using Integration Services features such as variables, property expressions, and package configurations. For more information, see <b>Integration Services Variables</b> and <b>Using</b> <b>Variables in Packages: Using Property Expressions in</b>
	Packages: and Package Configurations.
Replace functionality of Execute Package	The Integration Services Package Migration Wizard does not migrate Execute Packages tasks that load Meta Data Services packages. The migrated package will contain encapsulated DTS functionality which must eventually be replaced. SQL Server 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

task (with Meta	SQL Server 2000 Client Tools.
Data Services	
package)	<b>Corrective action</b> : After migration, move the Meta Data
encapsulated in	Services package to SQL Server storage (in the <b>msdb</b>
Execute DTS 2000	database) or to a structured storage file, and convert the
Package task after	Execute DTS 2000 Task that encapsulates the Execute
package migration	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 <b>Saving Packages</b> .
	The Integration Services Package Migration Wizard does
Renlace	not migrate Parallel Data Pump tasks in SQL Server 2000
functionality of	Data Transformation Services (DTS) packages. The
Parallel Data	migrated package will contain encapsulated DTS
Pumn task	functionality which must eventually be replaced.
encansulated in	
Execute DTS 2000	<b>Corrective action</b> : After migration, re-create the
Package task after	functionality of the Parallel Data Pump task by using
nackage migration	Integration Services data flow components. For more
puchage migration	information, see <b>Creating Package Data Flow</b> and <b>Data</b>
	Flow Elements.
	The Integration Services Package Migration Wizard does
	not migrate tasks that obtain their connection information
Replace	from Microsoft Data Link (.udl) files. The migrated
functionality of	package will contain encapsulated DTS functionality which
UDL connection	must eventually be replaced.
encapsulated in	Corrective action: After migration convert the Evecute
Execute DTS 2000	DTS 2000 Task that on cansulates the UDL file connection
Package task after	to an equivalent Integration Services task that uses an
package migration	Integration Services connection manager. For more
	information see Connection Managers and Creating
	Connection Managers
	The Integration Services Deckage Migration Wigard does
	not migrate ActiveX Script code that accesses the DTS
	object model through the Parent property of the
	Clabal Variables callestion in SOL Server 2000 Date
	Global variables collection in SQL Server 2000 Data Transformation Services (DTS) packages. After migration

script code that	you must manually edit the resulting SSIS package to
access the DTS	restore former package behavior.
object model after	
package migration	<b>Corrective action</b> : 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.

# <sup>□</sup> 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
	ODBC connections raise an error if you try to edit the
Delete and	connection after migration.
recreate ODBC	Ŭ
connections	<b>Corrective action</b> : If you have already migrated the package,
after package	you can delete and recreate the ODBC connection. This issue
migration	has been fixed in SQL Server Service Pack 1. Install Service
	Pack 1 or later to prevent this issue from recurring.

### $\Box$ See Also

### Concepts

Integration Services Backward Compatibility

© 2008 Microsoft Corporation. All rights reserved.



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating Data Transformation Services Packages</u> >

Many SQL Server 2000 Data Transformation Services (DTS) tasks map directly to Integration Services tasks, and migrating a package to Integration Services replaces the SQL Server 2000 version of a task with its Integration Services 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	FTP 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

### <sup>I</sup>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 Integration Services tasks:

- Analysis Services task (because the Integration Services 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, 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 non-task feature.

SQL Server 2000 task	SQL Server 2005 feature
Dynamic Properties	Configurations and property expressions. For more information, see <b>Creating Package Configurations</b> .

Transform Data	Data Flow Task and Data Flow Elements.
Data Driven Query	Consider the <b>Slowly Changing Dimension Transformation</b> , the <b>Conditional Split Transformation</b> , and the <b>OLE DB Command Transformation</b> when replacing the functionality of Data Driven Query tasks.

# <sup>□</sup> Custom Tasks

A package that contains custom tasks is migrated to SQL Server 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 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.

# <sup>□</sup> 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.

# <sup>□</sup> Dynamic Properties Task

Dynamic Properties tasks cannot be migrated to SQL Server. 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, configurations replace and enhance the functionality available in the Dynamic Properties task. For more information, see **Creating Package Configurations**.

### $\Box$ See Also

### **Other Resources**

Migrating Data Transformation Services Packages

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Migrating Precedence Constraints



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating Data Transformation Services Packages</u> >

Precedence constraints in a SQL Server 2000 Data Transformation Services (DTS) package migrate to the **PrecedenceConstraints** collection of the new package.

# <sup>©</sup> 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**.

### $\Box$ See Also

### **Other Resources**

Migrating Data Transformation Services Packages Precedence Constraints

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Migrating Variables



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating Data Transformation Services Packages</u> >

In Integration Services, 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.

Integration Services uses namespaces to organize variables. All DTS variables are migrated to the **User** namespace.
# <sup>©</sup> 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 **Integration Services Variables** and **Using Variables in Packages**.

#### **Other Resources**

Migrating Data Transformation Services Packages

#### SQL Server Setup Help Migrating Connections



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating Data Transformation Services Packages</u> >

Connections in SQL Server 2000 Data Transformation Services (DTS) packages are migrated to SQL Server 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 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**.

#### <sup>☑</sup> 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 <u>Migrating Tasks</u> and <u>Known</u> <u>DTS Package Migration Issues</u>.

Any connections that were used in SQL Server 2000 tasks but cannot be migrated to SQL Server 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.

# <sup>©</sup> 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 <b>False</b>
	by the Package Migration Wizard.
Description	A description of the connection manager. This property is left
	blank by the Package Migration Wizard.

#### **Other Resources**

Migrating Data Transformation Services Packages

#### SQL Server Setup Help Migrating Passwords



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating Data Transformation Services Packages</u> >

SQL Server 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 do not retain passwords, although you must provide the password during the migration process. For information about package protection levels in SQL Server, see **Setting the Protection Level of Packages**.

#### <sup>☑</sup>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 data transformation solutions. SQL Server 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.

# <sup>□</sup> 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 **Integration Services Packages**.
- Digitally sign the package and set properties to ensure that the signature is checked when the package is loaded. For more information, see **Using Digital Signatures with Packages**.

#### <sup>I</sup>Important:

When configured to check the signature of the package, Integration Services only checks whether the digital signature is present, is valid, and is from a trusted source. Integration Services does not check whether the package has been changed.

• 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 Overview (Integration Services)**.

#### **Other Resources**

Migrating Data Transformation Services Packages



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating Data Transformation Services Packages</u> >

The SQL Server 2000 version of the ActiveX Script task can be migrated to the SQL Server ActiveX Script task. However the SQL Server ActiveX Script task is provided only for backward compatibility. After migration, you should replace ActiveX Script tasks with Integration Services features.

If your scripts contain code to manage control flow in packages, you might be able to replace the scripts with new control flow features in Integration Services, such as looping containers and more flexible precedence constraints. For more information, see **Integration Services Containers** and **Precedence Constraints**.

You can replace other customer logic in your scripts with managed code in the Integration Services Script task. For more information on the Integration Services Script task, see **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 Integration Services features.

Other Resources ActiveX Script Task Script Task <u>Migrating Data Transformation Services Packages</u>

#### SQL Server Setup Help Migrating Data Transformation Services Packages How-to Topics

# 

#### Ξ 🖌

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> >

This section contains the following common procedures used for migrating SQL Server 2000 Data Transformation Services (DTS) packages to Microsoft Integration Services:

How to: Migrate Data Transformation Services Packages to Integration
Services

For common procedures used for installing Integration Services, see <u>Integration</u> <u>Services Installation How-to Topics</u>.

#### Concepts

<u>Considerations for Upgrading Data Transformation Services</u> <u>Integration Services Backward Compatibility</u>

SQL Server Setup Help

How to: Migrate Data Transformation Services Packages to Integration Services



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Considerations for Upgrading Data Transformation Services</u> > <u>Migrating</u> <u>Data Transformation Services Packages How-to Topics</u> >

You are not required to migrate SQL Server 2000 Data Transformation Services (DTS) packages to the Integration Services package format. DTS packages can continue to run alongside Integration Services packages.

#### <sup>I</sup>Note:

For information on upgrading packages that were created in an earlier version of Integration Services, see <u>Upgrading Integration Services</u> <u>Packages</u>.

If you want to migrate your DTS packages, SQL Server provides the DTS 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 <u>How to: Install Support for Data</u> <u>Transformation Services Packages</u>.

#### <sup>ℤ</sup>Note:

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server.

For more information about migrating DTS packages, see <u>Migrating Data</u> <u>Transformation Services Packages</u>. For more information about the upgrade and migration options available to users of Data Transformation Services, see <u>Considerations for Upgrading Data</u> <u>Transformation Services</u>.

For general information about support for DTS in Integration Services, see <u>Support for Data Transformation Services (DTS) in SQL Server 2008</u>.

#### <sup>©</sup> 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 <u>Known DTS Package Migration Issues</u>.
- 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: Install Support for Data Transformation Services Packages

#### Concepts

Integration Services Backward Compatibility

# SQL Server Setup Help

#### Considerations for Upgrading Replicated Databases

# 

#### - -

#### <u>Upgrading to SQL Server 2008</u> >

SQL Server 2008 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 2000 service pack 3 (SP3) is the minimum version required to participate in a replication topology with SQL Server 2008. There is no minimum version required if you use SQL Server 2005.
- 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 Subscriber to a transactional publication can be any version within two versions of the Publisher version. For example: a SQL Server 2000 Publisher running can have SQL Server 2008 Subscribers; and a SQL Server 2008 Publisher can have SQL Server 2000 Subscribers.
  - A Subscriber to a merge publication can be any version less than or equal to the Publisher version.

#### <sup>ℤ</sup>Note:

This topic is available in the Setup Help documentation and in SQL Server Books Online. Topic links that appear as bold text in the Setup Help documentation refer to topics that are only available in Books Online.

# Run the Log Reader Agent for Transactional Replication Before Upgrade

Before you upgrade to SQL Server 2008, you must make sure that all committed transactions from published tables have been processed by the Log Reader Agent. To make sure that all transactions have been processed, perform the following steps for each database that contains transactional publications:

- 1. Make sure that the Log Reader Agent is running for the database. By default, the agent runs continuously.
- 2. Stop user activity on published tables.
- 3. Allow time for the Log Reader Agent to copy transactions to the distribution database, and then stop the agent.
- 4. Execute **sp\_replcmds** to verify that all transactions have been processed. The result set from this procedure should be empty.
- 5. Execute **sp\_replflush** to close the connection from **sp\_replcmds**.
- 6. Perform the server upgrade to SQL Server 2008.
- 7. Restart SQL Server Agent and the Log Reader Agent if they do not start automatically after the upgrade.

# <sup>©</sup> Run Agents for Merge Replication After Upgrade

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 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)
- Replication Agent Executables Concepts

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 <u>Using Multiple Versions of SQL Server 2008</u> in a Replication Topology.

# <sup>D</sup> Upgrading to Standard, Workgroup, or Express Editions

Before upgrading from one edition of SQL Server 2008 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 2008 Replication Features" in the topic Features Supported by the Editions of SQL Server 2008.

# <sup>□</sup> New Replication Agent Security Model

In versions of SQL Server prior to SQL Server 2005, agents ran, by default, under the context of the SQL Server Agent service account. SQL Server 2005 introduced 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 and Protection (Replication)** and **Replication Agent Security Model**.

The new security model has the following implications for upgrading and running SQL Server 2000 in a topology:

- Replication scripts created from SQL Server 2000 should be upgraded for SQL Server 2008 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 SQL Server 2000 to SQL Server 2008 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.

<sup>I</sup>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 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 2008 to a Subscriber running SQL Server 2000 uses the new security model because the Merge Agent or Distribution is created at the Distributor.
  - A Publisher running SQL Server 2000 with a Distributor running SQL Server 2008 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.
- SQL Server 2005 and SQL Server 2008 use the same security model.

## <sup>©</sup> 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 an Earlier 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.

#### □ See Also

#### Reference

Version and Edition Upgrades

#### Concepts

Replication Backward Compatibility Using Multiple Versions of SQL Server 2008 in a Replication Topology Upgrading to SQL Server 2008

Other Resources Administration (Replication) What's New (Replication)

SQL Server Setup Help Considerations for Upgrading SQL Server Management Tools

# 

#### - -

#### <u>Upgrading to SQL Server 2008</u> >

SQL Server 2008 supports upgrade from SQL Server 2000 or SQL Server 2005. This topic documents support and behavior for upgrading SQL Server Management Tools and management components such as SQL Server Agent, Database Mail, Maintenance Plans, XPStar, and XPWeb.

### <sup>Important:</sup>

For local installations, you must run SQL Server Setup as an administrator. If you run SQL Server Setup from a remote share, you must use a domain account that has read and execute permissions on the remote share.

# <sup>□</sup> Known Upgrade Issues

Consider the following issues before you upgrade to SQL Server 2008:

#### For all upgrade scenarios:

- All TSX servers should be upgraded before the MSX server is upgraded. For more information about MSX/TSX in SQL Server, see <u>Automating</u> <u>Adminstration Across an Enterprise</u> in SQL Server Books Online.
- All components in an instance of SQL Server must be upgraded at the same time. Version numbers of the Database Engine, Analysis Services, and Reporting Services components must be the same in an instance of SQL Server 2008.
- You can add components to an existing installation of SQL Server at the time that you upgrade to SQL Server 2008. For more information, see <u>How</u> to: Upgrade to SQL Server 2008 (Setup).
- SQL Server Client Tools, such as SQL Server Management Studio, SQL Server Profiler, the Database Engine Tuning Advisor, sqlcmd, osql, and so on, are not upgraded to SQL Server 2008. Instead, Client Tools run side-by-side with tools from previous SQL Server versions. SQL Server 2008 supports importing settings from earlier versions of SQL Server Client Tools.
- Authentication from SQL Server Agent to SQL Server will be updated from SQL Server Authentication to Windows Authentication during upgrade. SQL Server Authentication is not supported in SQL Server 2008.
- Data for jobs and alerts will be preserved during upgrade to SQL Server 2008.
- If SQLMail is being used in the instance to be upgraded, associated XPs will be supported and enabled after the upgrade. Otherwise, they will be off.
- Database Mail, also known as SQLiMail, will be upgraded with the Database Engine component of SQL Server 2008. By default, Database Mail will be off after upgrade. Any schema updates should be reconciled with an update script after upgrade.

#### **Upgrading from SQL Server 2000:**

- During an upgrade from SQL Server 2000 to SQL Server 2008, all user proxy accounts that existed before upgrading will be changed to the temporary global proxy account **UpgradedProxyAccount**. The **UpgradedProxyAccount** is granted access only to those subsystems that were explicitly used, and does not have access to all subsystems after the upgrade is complete.
- Although token substitution for SQL Server Agent alerts and job syntax has changed from SQL Server 2000, tokens will not be automatically updated during the upgrade to SQL Server 2008.

Tasks

How to: Upgrade to SQL Server 2008 (Setup)

**Reference** <u>Version and Edition Upgrades</u>

#### Other Resources Backward Compatibility

SQL Server Setup Help Upgrade How-to Topics



-

<u>Upgrading to SQL Server 2008</u> >

The topics in this section describe how to upgrade a SQL Server stand-alone instance.

Торіс	Description
<u>How to: Upgrade to SQL Server</u> 2008 (Setup)	The procedural topic to upgrade a stand- alone instance of SQL Server to SQL Server 2008.
• <u>How to: Upgrade a SQL</u> <u>Server Failover Cluster</u> <u>Instance (Setup</u> )	The procedural topic to upgrade an instance of a SQL Server failover cluster to SQL Server 2008.
• <u>How to: Install SQL</u> <u>Server 2008 from the</u> <u>Command Prompt</u>	Provides sample syntax for installing SQL Server 2008 at a command prompt.

#### Reference

Version and Edition Upgrades

#### Concepts

Backward Compatibility

SQL Server Setup Help How to: Upgrade to SQL Server 2008 (Setup)

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Upgrade How-to Topics</u> >

The SQL Server Installation Wizard provides a single feature tree for upgrade of SQL Server components. You can also install SQL Server 2008 side by side with an earlier version, or migrate existing databases and configuration settings from an earlier SQL Server version, and apply them to an instance of SQL Server 2008.

For more information, see these topics:

- <u>Version and Edition Upgrades</u>
- For more information about support for side-by-side configurations, see <u>Working with Multiple Versions and Instances of SQL Server</u>.
- How to: Upgrade a SQL Server Failover Cluster Instance (Setup)
- How to: Install SQL Server 2008 from the Command Prompt
- Using the Copy Database Wizard
- How to: Upgrade SQL Server with the Copy Database Wizard

# <sup>©</sup> 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, review the following topics:

- <u>Upgrading to SQL Server 2008</u>
- Hardware and Software Requirements for Installing SQL Server 2008
- <u>Check Parameters for the System Configuration Checker</u>
- <u>Security Considerations for a SQL Server Installation</u>
- SQL Server Database Engine Backward Compatibility

# <sup>©</sup> Procedure

#### <sup>©</sup> To upgrade to SQL Server 2008

1.

Insert the SQL Server installation media, and from the root folder, double-click setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click setup.exe.

• If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.

• Windows Installer 4.5 is also required, and may be installed by the Installation Wizard. If you are prompted to restart your computer, restart, and then run SQL Server 2008 setup.exe again.

• When prerequisites are installed, the Installation Wizard will launch the SQL Server Installation Center. To upgrade an existing instance of SQL Server 2008, click **Upgrade from SQL Server 2000 or SQL Server 2005**.

• If Setup support files are required, SQL Server Setup will install them. If you are instructed to restrat your computer, restart before you continue.

• The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information about log files, see <u>How to: View SQL Server</u> <u>2008 Setup Log Files</u>.

• On the Product key page, click a radio button to indicate whether you are upgrading to a free edition of SQL Server, or whether you have a PID key for a production version of the product. For more information, see <u>Editions and</u> <u>Components of SQL Server 2008</u> and <u>Version and Edition Upgrades</u>.

• On the License Terms page, read the license agreement, and then select the check box to accept the licensing terms and conditions. To continue, click **Next**. To end Setup, click **Cancel**.

• On the Select Instance page, specify the instance of SQL Server to upgrade.

• On the Feature Selection page, the features to upgrade will be pre-selected. A description for each component group appears in the right-hand pane after you select the feature name. Note that you cannot change the features to be upgraded, and you cannot add features during the upgrade operation. To add features to an upgraded instance of SQL Server 2008 after the upgrade operation is complete, see <u>How to: Add Features to an Instance of SQL Server 2008</u> (Setup).

• On the Instance Configuration page, specify whether to install a default or a named instance. For more information, see <u>Instance Configuration</u>.

**Instance ID suffix** — By default, the instance name is used as the Instance ID suffix. This is used to identify installation directories and registry keys for your instance of SQL Server. This is the case for default instances and named instances. For a default instance, the instance name and instance ID suffix would be MSSQLSERVER. To use a non-default instance ID suffix, select the **Instance ID suffix** check box and provide a value.

#### <sup>☑</sup> Note:

Typical stand-alone instances of SQL Server 2008, whether default or named instances, do not use a non-default value for the **Instance ID suffix** check box.

**Instance root directory** — By default, the instance root directory is C:\Program Files\Microsoft SQL Server\. To specify a non-default root directory, use the field provided, or click **Browse** to locate an installation folder.

All SQL Server service packs and upgrades will apply to every component of an instance of SQL Server.

**Detected instances and features** — The grid will show instances of SQL Server that are on the computer where Setup is running. If a default instance is already installed on the computer, you must install a named instance of SQL Server 2008. To continue, click **Next**.
• The Disk Space Requirements page calculates the required disk space for the features you specify, and compares requirements to the available disk space on the computer where Setup is running. For more information, see <u>Disk Cost</u> <u>Summary</u>.

• Work flow for the remainder of this topic depends on the features you have specified for your installation. You might not see all of the pages, depending on your selections.

• On the Server Configuration — Service Accounts page, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features you are upgrading.

Authentication and login inromation will be carried forward from the previous instance of SQL Server. 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, are started manually, or are disabled. Microsoft recommends that you configure service accounts individually, so that SQL Server services are granted the minimum permissions they need to complete their tasks. For more information, see <u>SQL Server</u> <u>Configuration - Service Accounts</u> and <u>Setting Up Windows Service Accounts</u>.

To specify the same login account for all service accounts in this instance of SQL Server, provide credentials in the fields at the bottom of the page.

**Security Note** Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

• Use the **Server Configuration** — **Collation** tab to specify non-default collations for the Database Engine and Analysis Services. For more information, see <u>SQL Server Configuration - Collation</u>.

• On the Full-Text Search Upgrade Options page, specify the upgrade options for the databases being upgraded. For more information, see <u>Full-Text Search</u> <u>Upgrade Options</u>.

• On the **Error and Usage Reporting** page, specify the information you would like to send to Microsoft that will help to improve SQL Server. By default,

options for error reporting and feature usage are enabled. For more information, see <u>Error and Usage Report Settings</u>.

• The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified before the upgrade operation begins.

• The Ready to Upgrade page displays a tree view of installation options that were specified during Setup. To continue, click **Install**.

• During installation, the progress page provides status so you can monitor installation progress as Setup proceeds.

• After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.

• If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For information about Setup log files, see <u>How to: View SQL Server 2008 Setup Log Files</u>.

## <sup>□</sup> Next Steps

After upgrading to SQL Server 2008, complete the following tasks:

- **Register your servers** Upgrade removes registry settings for the previous instance of SQL Server. After upgrading, you must reregister your servers.
- **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 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 row or page counts that are not valid, 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 selectively installs and activates key services and features. For more information about surface area configuration, see the readme file for this release.

## $\Box$ See Also

#### **Concepts** Backward Compatibility

#### Other Resources Upgrading to SQL Server 2005

SQL Server Setup Help How to: Upgrade to a Different Edition of SQL Server 2008 (Setup)



- -

<u>Upgrading to SQL Server 2008</u> > <u>Upgrade How-to Topics</u> >

The SQL Server Installation Wizard provides a single feature tree for upgrade of SQL Server components. You can also install SQL Server 2008 side-by-side with an earlier version, or migrate existing databases and configuration settings from an earlier SQL Server version, and apply them to an instance of SQL Server 2008.

For more information, see these topics:

- <u>Version and Edition Upgrades</u>
- <u>Working with Multiple Versions and Instances of SQL Server</u>
- <u>How to: Upgrade a SQL Server Failover Cluster Instance (Setup)</u>
- How to: Install SQL Server 2008 from the Command Prompt
- How to: Upgrade SQL Server with the Copy Database Wizard

## <sup>©</sup> 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 you upgrade the Database Engine, review the following topics:

- <u>Upgrading to SQL Server 2008</u>
- Hardware and Software Requirements for Installing SQL Server 2008
- <u>Check Parameters for the System Configuration Checker</u>
- <u>Security Considerations for a SQL Server Installation</u>
- <u>SQL Server Database Engine Backward Compatibility</u>

#### <sup>I</sup>Important:

For the SQL Server edition change to be activated, you must restart SQL Server services. This will result in application down time while services are offline.

## <sup>©</sup> Procedure

#### <sup>©</sup> To upgrade to SQL Server 2008

1.

Insert the SQL Server installation media. From the root folder, double-click setup.exe. To install from a network share, locate the root folder on the share, and then double-click setup.exe.

• If the Microsoft .NET Framework version 2.0 installation dialog box appears, select the check box to accept the .NET Framework 2.0 License Agreement. Click **Next**. To quit SQL Server 2008 installation, click **Cancel**. When installation of .NET Framework 2.0 is complete, click **Finish**.

• Windows Installer 4.5 is also required, and might be installed by the Installation Wizard. If you are prompted to restart your computer, restart, and then run SQL Server 2008 setup.exe again.

• When prerequisites are installed, the Installation Wizard will launch the SQL Server Installation Center. To upgrade an existing instance of SQL Server 2008 to a different edition, click **Advanced**, and then select **Edition Upgrade**.

• If Setup support files are required, SQL Server Setup will install them. If you are instructed to restart your computer, restart before you continue.

• The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information about log files, see <u>How to: View SQL Server</u> <u>2008 Setup Log Files</u>.

• On the Product Key page, select a radio button to indicate whether you are upgrading to a free edition of SQL Server, or whether you have a PID key for a production version of the product. For more information, see <u>Editions and</u> <u>Components of SQL Server 2008</u> and <u>Version and Edition Upgrades</u>.

• On the License Terms page, read the license agreement, and then select the check box to accept the licensing terms and conditions. To continue, click **Next**. To end Setup, click **Cancel**.

• On the Select Instance page, specify the instance of SQL Server to upgrade.

• The System Configuration Checker will run one more set of rules to validate your computer configuration before the edition upgrade operation begins.

• The Ready to Upgrade page shows a tree view of installation options that were specified during Setup. To continue, click **Install**.

• During installation, the progress page provides status so you can monitor installation progress as Setup proceeds.

• After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.

• If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For information about Setup log files, see <u>How to: View SQL Server 2008 Setup</u> <u>Log Files</u>.

• If you upgraded from SQL Server Express, you must perform additional steps before you can use your upgraded instance of SQL Server:

- Enable the SQL Server Agent service in Windows SCM.
- Provision the SQL Server Agent service account by using SQL Server Configuration Manager.
- Users that were provisioned as Sysadmins will remain provisioned after the upgrade. Specifically, BUILTIN\Users will remain provisioned. Disable, remove, or reprovision these accounts, as it is required.
- Sizes and recovery mode for TempDB and ModelDB will remain unchanged after the upgrade. Reconfigure these settings, as it is required.
- Template databases will remain on the computer after the upgrade. Remove or reconfigure them, as it is required.

## $\Box$ See Also

#### Concepts

Upgrading to SQL Server 2008 Backward Compatibility

SQL Server Setup Help Backward Compatibility



-

<u>Upgrading to SQL Server 2008</u> >

The following sections contain backward compatibility information for SQL Server components. This content includes information about deprecated features, discontinued features, breaking changes, and behavior changes.

## <sup>□</sup> In This Section

Торіс	Description
SQL Server Backward Compatibility	Contains topics that describe changes in SQL Server that might require you to make changes to your applications. Features that are included in this topic area include data programmability, surface area configuration tools, Setup, SQL Server services, and other broad functionality changes.
<u>SQL Server</u> Database Engine Backward Compatibility	Contains topics that describe Database Engine changes in SQL Server that might require you to make changes to your applications.
<u>SQL Server</u> Analysis <u>Services</u> Backward Compatibility	Contains topics that describe Analysis Services changes in SQL Server that might require you to make changes to your applications.
<u>Integration</u> <u>Services</u> <u>Backward</u> Compatibility	Describes Integration Services changes in SQL Server that might require you to make changes to your existing Data Transformation Services applications.
Reporting <u>Services</u> Backward Compatibility	Contains topics that describe SSRS changes in SQL Server that might require you to make changes to your existing Reporting Services solutions.
<u>Replication</u> <u>Backward</u> Compatibility	Contains topics that describe Database Engine changes in SQL Server that might require you to make changes to existing Replication solutions.

## $\Box$ See Also

#### Concepts

Installing SQL Server 2008 Upgrading to SQL Server 2008 Migrating to SQL Server 2008

SQL Server Setup Help SQL Server Backward Compatibility - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> >

Topics in the backward compatibility section describe changes in SQL Server behavior between versions of SQL Server.

Features included in this topic area include data programmability, surface area configuration tools, SQL Server Setup, SQL Server services, and other broad functionality changes.

Торіс	Description	
<u>Behavior Changes to</u>	Deprecated SQL Server features in this release. This	
<u>Database Engine</u>	topic includes coverage of Data Programmability	
<u>Features in SQL</u>	features, Surface Area Configuration tools, and Setup	
<u>Server 2008</u>	functionality.	
<u>Behavior Changes to</u>	Discontinued SQL Server functionality in this release.	
<u>Database Engine</u>	This topic includes coverage of Data Programmability	
<u>Features in SQL</u>	features, Surface Area Configuration tools, and Setup	
<u>Server 2008</u>	functionality.	
<u>Behavior Changes to</u>	Changes that might require changes to applications. This	
<u>Database Engine</u>	topic includes coverage of Data Programmability	
<u>Features in SQL</u>	features, Surface Area Configuration tools, and Setup	
<u>Server 2008</u>	functionality.	
Behavior Changes to	Other changes to database features in this release. This	
Database Engine	topic includes coverage of Data Programmability	
Features in SQL	features, Surface Area Configuration tools, and Setup	
Server 2008	functionality.	

#### SQL Server Setup Help Deprecated SQL Server Features in SQL Server 2008 ■ ✓

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Backward Compatibility</u> >

This topic describes the deprecated features that are still available in SQL Server 2008. These features are scheduled to be removed in a future release of SQL Server. Deprecated features should not be used in new applications.

## Features Not Supported in the Next Version of SQL Server

The following SQL Server Database Engine features will not be supported in the next version of SQL Server. Do not use these features in new development work, and modify applications that currently use these features as soon as possible. The Feature name column appears in trace events as the ObjectName, in performance counters and sys.dm\_os\_performance\_counters as the instance\_name. The Feature ID appears in trace events as the ObjectId.

Category	Deprecated feature	Replacement	Feature name	Featur ID
SQL Server services	ADHelper service	None	ADHelper	
Data Programmability	SOAP/HTTP endpoints with <b>CREATE</b> <b>ENDPOINT (Transact-</b> <b>SQL)</b> and <b>ALTER</b> <b>ENDPOINT (Transact-</b> <b>SQL)</b>	Windows Communications Foundation (WCF) or ASP.NET	Native XML Web Services	21
Data Programmability	sys.soap_endpoints (Transact-SQL)	Windows Communications Foundation (WCF) or ASP.NET	Native XML Web Services	22
Data Programmability	sys.endpoint_webmethods (Transact-SQL)	Windows Communications Foundation (WCF) or ASP.NET	Native XML Web Services	23

# SQL Server Setup Help Discontinued SQL Server Features in SQL Server 2008

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Backward Compatibility</u> >

This topic describes features that are no longer available in SQL Server 2008.

#### Surface Area Configuration Tool

The Surface Area Configuration tool (SAC) for SQL Server has been removed from SQL Server 2008. Replacement functionality in this release is as follows:

Category	Discontinued feature	Replacement	
		The Surface Area Configuration tool is discontinued for SQL Server 2008. The following table shows what you can use to configure settings, options, and component features in the February 2008 CTP.	
		Settings and component features How to configure	
		Protocols, connection, and startup options	Use SQL Server Configuration Manager.
		Database Engine features	Use Policy-Based Management, the property settings in SQL Server Management Studio, or sp_Configure.
Tools	Surface Area Configuration tool	Analysis Services features	Use the property settings in SQL Server Management Studio.

	Reporting Services - EnableIntegrated Security property	Use the property settings in SQL Server Management Studio.
	Reporting Services - "Schedule events and report delivery" and "Web service and HTTP access"	Edit the RSReportServer.config configuration file.
	Command line options	No support in this release.
	SOAP and Service Broker endpoints	Use <b>CREATE</b> ENDPOINTand ALTER ENDPOINT.

#### **Discontinued Command Prompt Parameters for SQL Server Setup**

The following table shows Setup command prompt parameters from earlier versions of SQL Server that are not supported in SQL Server 2008.

Discontinued parameter	Replacement parameter
ADDLOCAL	/ACTION=Uninstall and /FEATURES
DISABLENETWORKPROTOCOLS	/TCPENABLED for TCP/IP <sup>1</sup>
DISABLENETWORKPROTOCOLS	/NPENABLED for Named Pipes <sup>1</sup>
	/SQLUSERDBDIR

	/SQLUSERDBLOGDIR	
INSTALLSQLDATADIR	/SQLBACKUPDIR	
	/SQLTEMPDBDIR	
	/SQLTEMPDBLOGDIR	
REINSTALL	No equivalent in this release.	
REINSTALLMODE	No equivalent in this release.	
REMOVE	/ACTION=Uninstall and /FEATURES	
SAMPLEDATABASE	No equivalent in this release.	
SAVESYSDB	No equivalent in this release.	
SKUUPGRADE	No equivalent in this release.	
UPGRADE	/ACTION=Upgrade and /FEATURES	
USESYSDB	No equivalent in this release.	

<sup>1</sup>These parameters are valid only for installation.

For more information, see <u>How to: Install SQL Server 2008 from the Command</u> <u>Prompt</u>.

# SQL Server Setup Help Breaking Changes to SQL Server Features in SQL Server 2008

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Backward Compatibility</u> >

This topic describes breaking changes in SQL Server. These changes might break applications, scripts, or functionalities that are based on earlier versions of SQL Server. You might encounter these issues when you upgrade. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.

## <sup>□</sup> Subhead

Insert section body here.

# SQL Server Setup Help Behavior Changes to SQL Server Features in SQL Server 2008

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Backward Compatibility</u> >

This topic describes behavior changes in SQL Server 2008. Behavior changes affect how features work or interact in SQL Server 2008 as compared to earlier versions of SQL Server.

## <sup>□</sup> Remarks

Insert Remarks here.

Feature	For more information	
SQL Server	Updating an Application to SQL Server 2008 Native Client	
Native Client	from SQL Server 2005 Native Client	

#### SQL Server Setup Help Support for Previous Versions of SQL Server ■ ✓

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Backward Compatibility</u> >

Topics in this section describe SQL Server support for the following legacy components and features:

See	For information about
Distributed Management Objects	Support for Distributed Management Objects.
© 2008 Microsoft Corporation. All rights reserved.	

SQL Server Setup Help Distributed Management Objects



- -

Backward Compatibility > SQL Server Backward Compatibility > Support for Previous Versions of SQL Server >

The Distributed Management Objects feature is installed by SQL Server 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**.

## $\Box$ See Also

#### Reference

Editions and Components of SQL Server 2008

#### SQL Server Setup Help

Data Transformation Services (DTS)

- -

Backward Compatibility > SQL Server Backward Compatibility > Support for Previous Versions of SQL Server >

Data Transformation Services (DTS) has been replaced by SQL Server Integration Services.

SQL Server 2008 does not install DTS support. For more information, see <u>Support for Data Transformation Services (DTS) in SQL Server 2008</u>.

## <sup>D</sup> Data Transformation Services Is Deprecated

Data Transformation Services (DTS) is deprecated. You should plan to remove dependencies on DTS in anticipation of future releases of SQL Server. DTS itself will continue to be supported in SQL Server 2000, SQL Server 2005, and SQL Server 2008 as part of our ongoing commitment to provide support for those versions of SQL Server. However, the ability to migrate or run DTS packages might not be supported in the next release.

The DTS functionality that will be deprecated includes the DTS runtime, the DTS API, the Package Migration Wizard for migrating DTS packages to the next version of Integration Services, support for DTS package maintenance in SQL Server Management Studio, and the Execute DTS 2000 Package task.

For information on how to migrate DTS packages to the Integration Services package format, see <u>Migrating Data Transformation Services Packages</u>.

SQL Server Setup Help SQL Server Database Engine Backward Compatibility

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> >

Topics in the backward compatibility section describe changes in behavior between versions of SQL Server.

Торіс	Description	
Deprecated Database Engine Features in SQL Server 2008	Deprecated Database Engine features in this release.	
Discontinued Database Engine Functionality in SQL Server 2008	Discontinued Database Engine features in this release.	
Breaking Changes to Database Engine Features in SQL Server 2008	Database Engine feature changes that might require changes to applications.	
<u>Behavior Changes to Database</u> Engine Features in SQL Server 2008	Other changes to database features in this release.	
© 2008 Microsoft Corporation. All rights reserved.		

#### SQL Server Setup Help Deprecated Database Engine Features in SQL Server 2008

## 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Database Engine Backward</u> <u>Compatibility</u> >

This topic describes the deprecated SQL Server Database Engine features that are still available in SQL Server 2008. These features are scheduled to be removed in a future release of SQL Server. Deprecated features should not be used in new applications.

You can monitor the use of deprecated features by using the SQL Server Deprecated Features Object performance counter and trace events. For more information, see **Using SQL Server Objects**.

# □ Features Not Supported in the Next Version of SQL Server

The following SQL Server Database Engine features will not be supported in the next version of SQL Server. Do not use these features in new development work, and modify applications that currently use these features as soon as possible. The **Feature name** value appears in trace events as the **ObjectName** and in performance counters and **sys.dm\_os\_performance\_counters** as the instance name. The **Feature ID** value appears in trace events as the **ObjectId**.

Category	Deprecated feature	Replacement
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	80 compatibility level and upgrade from version 80.	Compatibility levels are only available for the last two versions. For more information about compatibility levels, see <b>ALTER DATABASE</b> <b>Compatibility Level</b> <b>(Transact-SQL)</b> .
Metadata	DATABASEPROPERTY	DATABASEPROPERTYEX
Database objects	WITH APPEND clause on triggers	Re-create the whole trigger.
Instance	Default setting of <b>disallow results</b>	Default setting of <b>disallow</b> <b>results from triggers</b> option

options	<b>from triggers</b> option = 0	= 1
Database options	sp_dboption	ALTER DATABASE
Query hints	FASTFIRSTROW hint	OPTION (FAST n).
Remote servers	sp_addremotelogin sp_addserver sp_dropremotelogin sp_helpremotelogin sp_remoteoption	Replace remote servers by using linked servers.
Remote servers	@@remserver	Replace remote servers by using linked servers.
Remote	SET	Replace remote servers by
Security	sp_dropalias	Replace aliases with a combination of user accounts and database roles. Use <b>sp_dropalias</b> to remove aliases in upgraded databases.
SET options	SET DISABLE_DEF_CNST_CHK	None. Option has no effect.
SET options	SET ROWCOUNT for INSERT, UPDATE, and DELETE statements	TOP keyword
Transact-SQL syntax	Use of *= and =*	Use ANSI join syntax. For more information, see FROM (Transact-SQL).
Transact-SQL syntax	COMPUTE / COMPUTE BY	Use ROLLUP
System tables	sys.database_principal_aliases	Use roles instead of aliases.
Transact- SQL	The RAISERROR (Format: RAISERROR integer string) syntax is deprecated.	Rewrite the statement using the current RAISERROR syntax.
		Although the Database Engine still supports connections from existing

		applications that use the DB-Library and Embedded SQL APIs, it does not include the files or documentation required to do programming work on applications that use these APIs. A future version of the
		connections from DB-
		applications. Do not use
		DB-Library or Embedded
	DB-Library	SQL to develop new
Other		applications. Remove any
	Embedded SQL for C	dependencies on either DB-
		Library or Embedded SQL
		when you are modifying
		existing applications.
		the SOL Client namespace
		or an API such as OLE DB
		or ODBC SOL Server 2008
		does not include the DB-
		Library DLL required to run
		these applications. To run
		DB-Library or Embedded
		SQL applications, you must
		have available the DB-
		Library DLL from SQL
		Server version 6.5, SQL
		Server 7.0, or SQL Server
		2000.

# □ Features Not Supported in a Future Version of SQL Server

The following SQL Server 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
Compatibility levels	sp_dbcmptlevel
Compatibility levels	Database compatibility level 90
XML	Inline XDR Schema Generation
Backup and	BACKUP { DATABASE   LOG } TO TAPE
restore	BACKUP { DATABASE   LOG } TO device_that_is_a_tape
Backup and restore	sp_addumpdevice 'tape'
Backup and restore	sp_helpdevice
	Korean_Wansung_Unicode
Collations	Lithuanian_Classic
	SQL_AltDiction_CP1253_CS_AS
	Hindi
Collations	Macedonian

Collations	Azeri_Latin_90
	Azeri_Cyrilllic_90
Configuration	SET ANSI_NULLS OFF and ANSI_NULLS OFF database opt SET ANSI_PADDING OFF and ANSI_PADDING OFF databa SET CONCAT_NULL_YIELDS_NULL OFF and CONCAT_NULL_YIELDS_NULL OFF database option SET OFFSETS
Connections	SQLOLEDB is not a supported provider.
Data types	sp_addtype sp_droptype
Data types	timestamp syntax for rowversion data type
Data types	Ability to insert null values into <b>timestamp</b> columns.
Data types	' <b>text in row'</b> table option
Data types	Data types: <ul> <li>text</li> <li>ntext</li> <li>image</li> </ul>
Database management	sp_attach_db sp_attach_single_file_db
	CREATE DEFAULT

Database objects	DROP DEFAULT
	sp_bindefault
	sp_unbindefault
	CREATE RULE
	DROP RULE
Database objects	sp_bindrule
	sp_unbindrule
Database objects	sp_change_users_login
Database objects	sp_depends
Database objects	sp_renamedb
Database objects	sp_getbindtoken
Database options	sp_bindsession
Database options	sp_resetstatus
Database options	TORN_PAGE_DETECTION option of ALTER DATABASE
DBCC	DBCC DBREINDEX
DBCC	DBCC INDEXDEFRAG
DBCC	DBCC SHOWCONTIG
	DBCC PINTABLE
	DBCC UNPINTABLE
Extended properties	<b>Level0type</b> = 'type' and <b>Level0type</b> = 'USER' to add extended p or level-2 type objects.
	srv_alloc

	srv_convert
	srv_describe
	srv_getbindtoken
	srv_got_attention
	srv_message_handler
	srv_paramdata
	srv_paraminfo
	srv_paramlen
	srv_parammaxlen
	srv_paramname
	srv_paramnumber
	srv_paramset
	srv_paramsetoutput
	srv_paramstatus
Extended stored	srv_paramtype
programming	srv_pfield
	srv_pfieldex
	srv_rpcdb
	srv_rpcname
	srv_rpcnumber
	srv_rpcoptions
	srv_rpcowner
---	--
	srv_rpcparams
	srv_senddone
	srv_sendmsg
	srv_sendrow
	srv_setcoldata
	srv_setcollen
	srv_setutype
	srv_willconvert
	srv_wsendmsg
Extended stored procedure programming	sp_addextendedproc sp_dropextendedproc
	xp_grantlogin
Extended stored procedures	xp_revokelogin
	xp_loginConfig
Functions	fn_get_sql
Index options	sp_indexoption
Index options	CREATE TABLE, ALTER TABLE, or CREATE INDEX synta> parentheses around the options.
Database objects	Ability to return result sets from triggers
Instance options	<pre>sp_configure option 'allow updates'</pre>
	sp_configure options:
Instance options	<ul> <li>'locks'</li> <li>'open objects'</li> </ul>

	• 'set working set size'
Instance options	<pre>sp_configure options 'user instances enabled'</pre>
Instance options	<pre>sp_configure option 'priority boost'</pre>
Instance options	<ul> <li>sp_configure option 'remote proc trans'</li> </ul>
Linked servers	A linked server uses OLEDB.
Locking	sp_lock
Mail	SQL Mail
Metadata	FILE_ID INDEXKEY PROPERTY
Native XML Web Services	The CREATE ENDPOINT or ALTER ENDPOINT statement w option. EXT_endpoint_webmethods
	EXT_soap_endpoints
Programmability	SQL Server Database Management Objects (SQL-DMO)
Removable databases	<ul> <li>sp_certify_removable</li> <li>sp_create_removable</li> </ul>

Removable databases	sp_dbremove		
Security	The ALTER LOGIN WITH SET CREDENTIAL syntax		
Security	sp_addapprole sp_dropapprole		
Security	sp_addlogin sp_droplogin		
Security	sp_adduser		
Security	sp_grantdbaccess		
Security	sp_addrole		
Security	sp_password		
Security	sp_changeobjectowner		
Security	sp_defaultdb sp_defaultlanguage		
Security	sp_denylogin sp_grantlogin sp_revokelogin		
Security	USER_ID		
Security	sp_srvrolepermission sp_dbfixedrolepermission		
	GRANT ALL		

Security	DENY ALL		
	REVOKE ALL		
Security	PERMISSIONS intrinsic function		
Security	SETUSER		
	Microsoft.SQLServer.Management.Smo.Information class		
	Microsoft.SQLServer.Management.Smo.Settings class		
SMO classes	Microsoft.SQLServer.Management.Smo.DatabaseOptions cl		
	Microsoft.SqlServer.Management.Smo.DatabaseDdlTrigger. property		
	net send notification		
SQL Server Agent	Pager notification		
	ActiveX subsystem		
SQL Server	Solution Explorer integration in SQL Server Management Studi		
Studio	Source Control integration in SQL Server Management Studio		
	sysaltfiles		
	syscacheobjects		
	syscolumns		
	syscomments		
	sysconfigures		
	sysconstraints		
	syscurconfigs		

	sysdatabases
	sysdepends
	sysdevices
	sysfilegroups
	sysfiles
	sysforeignkeys
	sysfulltextcatalogs
	sysindexes
System tables	sysindexkeys
	syslockinfo
	syslogins
	sysmembers
	sysmessages
	sysobjects
	sysoledbusers
	sysopentapes
	sysperfinfo
	syspermissions
	sysprocesses
	sysprotects
	sysreferences

	sysremotelogins sysservers	
	systypes	
	sysusers	
System tables	sys.numbered_procedures sys.numbered_procedure parameters	
System	fn_virtualservernodes	
functions	fn_servershareddrives	
System views	sys.sql_dependencies	
Table compression	The use of the vardecimal storage format.	
Table compression	Use of the <b>sp_db_vardecimal_storage_format</b> procedure.	
Table compression	Use of the <b>sp_estimated_rowsize_reduction_for_vardecimal</b>	
Table hints	Specifying NOLOCK or READUNCOMMITTED in the FROM UPDATE or DELETE statement.	
Table hints	Specifying table hints without using the WITH keyword.	
Table hints		
Table hints		
	WRITETEXT	
Textpointers	UPDATETEXT	

	READTEXT	
Toutnointora	TEXTPTR()	
Textpointers	TEXTVALID()	
Transact-SQL	:: function-calling sequence	
Transact-SQL	Three-part and four-part column references in SELECT list	
Transact-SQL	A string enclosed in quotation marks used as a column alias for SELECT list: 'string_alias' = expression	
Transact-SQL	Numbered procedures	
Transact-SQL	table_name.index_name syntax in DROP INDEX	
Transact-SQL	Not ending Transact-SQL statements with a semicolon.	
Transact-SQL	GROUP BY ALL	
Transact-SQL	ROWGUIDCOL as a column name in DML statements.	
Transact-SQL	IDENTITYCOL as a column name in DML statements.	
Transact-SQL	Use of #, ## as temporary table and temporary stored procedure	
Transact-SQL	Use of @, @@, or @@ as Transact-SQL identifiers.	
Transact-SQL	Use of DEFAULT keyword as default value.	
Transact-SQL	Use of a space as a separator between table hints.	
Transact-SQL	The select list of an aggregate indexed view must contain COU compatibility mode	

Transact-SQL	The indirect application of table hints to an invocation of a mult valued function (TVF) through a view.	
	ALTER DATABASE syntax:	
Transact-SQL	MODIFY FILEGROUP READONLY	
	MODIFY FILEGROUP READWRITE	

## □ See Also

### Reference

Discontinued Database Engine Functionality in SQL Server 2008

#### SQL Server Setup Help

#### Discontinued Database Engine Functionality in SQL Server 2008

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Database Engine Backward</u> <u>Compatibility</u> >

This topic describes the Database Engine features that are no longer available in SQL Server 2008.

Category	Discontinued feature	Replacement
Aliases	sp_addalias	Replace aliases with a combination of user accounts and database roles. For more information, see <b>CREATE</b> <b>USER (Transact-SQL)</b> and <b>CREATE ROLE (Transact- SQL)</b> . Remove aliases in upgraded databases by using <b>sp_dropalias</b> .
APIs	Registered Servers API	Replaced by a new registered servers API that supports new SQL Server 2008 features.
Backup and restore	DUMP statement	BACKUP
Backup and restore	LOAD statement	RESTORE
		None. The transaction log is

Backup and restore	BACKUP LOG WITH NO_LOG	automatically truncated when the database is using the simple recovery model. If you must 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 must remove the log backup chain from a database, switch to the simple recovery model.
Backup and restore	BACKUP TRANSACTION	BACKUP LOG
Compatibility level	60, 65, and 70 compatibility levels	Databases must be set to at least compatibility level 80.
DBCC	DBCC CONCURRENCYVIOLATION	None
Groups	sp_addgroup	Use roles.
Groups	sp_changegroup	Use roles.
Groups	sp_dropgroup	Use roles.
Groups	sp_helpgroup	Use roles.

Sample databases	Northwind and pubs	Use <b>AdventureWorks</b> . <b>Northwind</b> and <b>pubs</b> are available as downloads, or can be copied from a previous installation of SQL Server. For more information, see <b>AdventureWorks Sample</b> <b>Databases</b> .
Tools	Surface Area Configuration Tool	The Surface Area Configuration Tool is discontinued for SQL Server 2008. For more information, see <u>Backward Compatibility</u> .
Web Assistant	sp_makewebtask sp_dropwebtask sp_runwebtask sp_enumcodepages	We recommend that you use SQL Server Reporting Services instead.

## $\Box$ See Also

### Reference

Deprecated Database Engine Features in SQL Server 2008

SQL Server Setup Help

Breaking Changes to Database Engine Features in SQL Server 2008



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Database Engine Backward</u> <u>Compatibility</u> >

This topic describes breaking changes in the Database Engine. These changes might break applications, scripts, or functionalities that are based on earlier versions of SQL Server. You might encounter these issues when you upgrade. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.

## <sup>□</sup> Collations

Feature	Description
New	SQL Server 2008 introduces new collations that are in full alignment with the collations provided by Windows Server 2008. These 80 new collations have improved linguistic accuracy and are denoted by *_100 version references. If you choose a new collation for your server or database, be aware that the collation may not be recognized by clients with older client drivers. Unrecognized collations can cause the application to return errors and fail. Consider the following solutions:
collations	<ul> <li>Upgrade the client operating system so that the underlying system collations are updated.</li> <li>If your client has database client software installed, consider applying a service update to the database client software.</li> <li>Choose an existing collation that maps to a code page on the client.</li> </ul> For more information, see Setting and Changing Collations.

# □ Common Language Runtime (CLR)

Feature	Description
CLR Assemblies	When a database is upgraded to SQL Server 2008, the <b>Microsoft.Sql</b> new data types is automatically installed. Upgrade Advisor rules dete conflicting names. The Upgrade Advisor will advise renaming of any renaming any conflicting type, or using two-part names in the code to If a database upgrade detects a user assembly with a conflicting name assembly and put the database into suspect mode.
	If a user type with conflicting name exists during the upgrade, no spe upgrade, both the old user type, and the new system type, will exist. T through two-part names.
CLR Assemblies	SQL Server 2008 installs .NET Framework 3.5 SP1, which updates li (GAC). If you have unsupported libraries registered in a SQL Server application may stop working after upgrading to SQL Server 2008. T libraries in the GAC does not update assemblies inside SQL Server. I Server database and in the GAC, the two copies of the assembly must an error will occur when the assembly is used by SQL Server CLR in <b>Supported .NET Framework Libraries</b> .
	databases with the ALTER ASSEMBLY statement. For more informa
	To detect whether you are using any unsupported .NET framework life following query in your database.
	SELECT name FROM sys.assemblies WHERE clr_name LIKE '%publ

# Dynamic Management Views

View	Description
eve dm ac eve infa	Removed the <b>cpu_ticks_in_ms</b> and
sys.um_0s_sys_nn0	<b>sqlserver_start_time_cpu_ticks</b> co
	The <b>resource_semaphore_id</b> colun
	a unique ID in SQL Server 2008. Th
sys.dm_exec_query_resource_semaphores	change can affect troubleshooting q
sys.dm_exec_query_memory_grants	execution. For more information, se
	sys.dm_exec_query_resource_sem
	(Transact-SQL).

## <sup>□</sup> Errors and Events

Feature	Description
Login	In SQL Server 2005, error 18452 is returned when a SQL login is used
orrorc	to connect to a server configured to use only Windows Authentication.
enois	In SQL Server 2008, error 18456 is returned instead.

## <sup>□</sup> Showplan

Feature	Description		
	A new <b>SeekPredicateNew</b> element is added to the Showplan XML sch		
	( <b>SqlPredicatesType</b> ) is converted into an <b><xsd:choice></xsd:choice></b> item. Instead (		
	elements, one or more SeekPredicateNew elements may now appear i		
	elements are mutually exclusive. <b>SeekPredicate</b> is maintained in the S		
Showplan	backwards compatibility; however, query plans created in SQL Server		
XML	SeekPredicateNew element. Applications that expect to retrieve only 1		
schema	node		
	ShowPlanXML/BatchSequence/Batch/Statements/StmtSimple/QueryP		
	may fail if the <b>SeekPredicate</b> element does not exist. Rewrite the appli		
	SeekPredicate or SeekPredicateNew element in this node. For more i		
	Enhancements on Partitioned Tables and Indexes.		
Showplan	A new <b>IndexKind</b> attribute is added to the <b>ObjectType</b> complex type :		
XML	Applications that strictly validate SQL Server plans returned by agains		
schema	fail.		

## <sup>□</sup> Transact-SQL

Feature	Description
	In SQL Server 2005, when the DDL $\epsilon$
	fires the value 'object' is returned in t
	<b>Object Type</b> element of the EVENTE
ALTER_AUTHORIZATION_DATABASE	xml for this event when the entity typ
DDL event	the securable in the data definition
	language (DDL) operation is an object
	SOL Server 2008, the actual type (for
	example, 'table', or 'function') is retur
	If an invalid style is passed to the
	CONVERT function, an error is retur
	when the type of conversion is binary
CONVERT	character or character to binary. In ea
CONVERI	versions of SQL Server, the invalid st
	set to the default style for binary-to-
	character and character-to-binary
	conversions.
	EXECUTE permission cannot be gra
	denied, or revoked to assemblies. Thi
GRANT/DENY/REVOKE EXECUTE on	permission has no affect and now cau
assemblies	an error. Grant, deny, or revoke EXE
assemblies	permission on the stored procedures of
	functions that reference the assembly
	method instead.
	Permissions cannot be granted, denie
GRANT/DENY/REVOKE permissions on	revoked to system types. In earlier ve
system types	of SQL Server, these statements succ
5 51	but have no effect. In SQL Server 200
	error is returned.
	10 prevent nondeterministic behavior
	OUTPUT clause cannot reference a
	function when that column is defined
	runcuon when that column is defined

	one of the following methods:
OUTPUT clause	<ul> <li>A subquery.</li> <li>A user-defined function that per user- or system-data access, or is assumed to perform such access</li> <li>A computed column that contain its definition a user-defined func- that performs user- or system-data access.</li> </ul>
	When SQL Server detects such a colu in the OUTPUT clause, error 4186 is raised. For more information, see <b>MSSQLSERVER_4186</b> .
OUTPUT INTO clause	The target table of the OUTPUT INT clause cannot have any enabled trigg
READPAST table hint	<ul> <li>The READPAST table hint cannot be specified when the READ_COMMITTED_SNAPSHOT database option is set to ON and eithe following conditions are true:</li> <li>The transaction isolation level o session is READ COMMITTED</li> <li>The READCOMMITTED table is also specified in the query.</li> </ul>
	To specify the READPAST hint in th cases, remove the READCOMMITT table hint if present, and include the READCOMMITTEDLOCK table hin the query.
	The following column names that are returned in the result set of the <b>sp_helpuser</b> stored procedure have changed.
sp_helpuser	Previous column New colum

	name	name
	GroupName	RoleName
	Group_name	Role_name
	Group_id	Role_id
	Users_in_group	Users_in_rol
Transparent Data Encryption	Transparent data encryp performed at the I/O lev structure is unencrypted encrypted only when the disk. Both the database are encrypted. Third-par bypass the regular SQL for accessing pages (for scanning the data or log fail when a database use data is encrypted in the applications can leverag Cryptographic API to de for decrypting the data of Server.	otion (TDE) is yel: the page l in memory a e page is writt files and log f rty application Server mecha example, by files directly es TDE becau files. Such ge the Windov evelop a solut butside of SQ.

# <sup>□</sup> XQuery

Feature	Description
Datetime support	<ul> <li>In SQL Server 2005, the data types xs:time, xs:date, and xs:dateTime do not have timezone support. Timezone data is mapped to the UTC timezone. SQL Server 2008, provides standard conformant behavior, resulting in the following changes:</li> <li>Values without timezone are validated.</li> <li>The provided timezone or the absence of a timezone is preserved.</li> <li>The internal storage representation is modified.</li> <li>Resolution of stored values is increased.</li> <li>Negative years are disallowed.</li> </ul>
	Modify applications and XQuery expressions to account for the new type values. For more information, see <b>Using XML with time, date, datetime2, and datetimeoffset Data Types</b> .

## □ See Also

#### Reference

Behavior Changes to Database Engine Features in SQL Server 2008 Deprecated Database Engine Features in SQL Server 2008 Discontinued Database Engine Functionality in SQL Server 2008

#### **Other Resources**

SQL Server Database Engine Backward Compatibility ALTER DATABASE Compatibility Level (Transact-SQL)

SQL Server Setup Help

Behavior Changes to Database Engine Features in SQL Server 2008



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Database Engine Backward</u> <u>Compatibility</u> >

This topic describes behavior changes in the Database Engine. Behavior changes affect how features work or interact in SQL Server 2008 as compared to earlier versions of SQL Server.

## <sup>©</sup> Query Processor Architecture

SQL Server 2008 changes the way queries on partitioned tables and indexes are processed. Queries on partitioned objects that use the USE PLAN hint for a plan generated by SQL Server 2005 might contain an invalid plan. For more information, see <u>Considerations for Upgrading the Database Engine</u>. For more information about query processing on partitioned objects, see **Query Processing Enhancements on Partitioned Tables and Indexes**.

## <sup>□</sup> Linked Servers

SQL Server 2008 changes the transaction semantics of INSERT...EXECUTE statements that execute against a loopback linked server. In SQL Server 2005, this scenario is not supported and causes an error. In SQL Server 2008, an INSERT...EXECUTE statement can execute against a loopback linked server when the connection does not have multiple active result sets (MARS) enabled. When MARS is enabled on the connection, the behavior is the same as in SQL Server 2005.

## <sup>□</sup> tempdb Database

In earlier versions of SQL Server, the PAGE\_VERIFY database option is set to NONE for the **tempdb** database and cannot be modified. In SQL Server 2008, the default value for the **tempdb** database is CHECKSUM for new installations of SQL Server. When upgrading an installation of SQL Server, the default value remains NONE. The option can be modified. We recommend that you use CHECKSUM for the **tempdb** database.

## <sup>□</sup> Plan Guides

In SQL Server 2008, if a plan guide cannot be honored, the query compiles using a different plan and no error is returned. In SQL Server 2005, an error is raised and the query fails.

Plan guides created in SQL Server 2005 may not be valid after upgrading to SQL Server 2008. Invalid plan guides will not cause the application to fail, but the plan guide will not be used. We recommend re-evaluating and testing plan guide definitions when you upgrade your application to a new release of SQL Server. Performance tuning requirements and plan guide matching behavior may change. Although an invalid plan guide will not cause a query to fail, the plan is compiled without using the plan guide. After you upgrade a database to SQL Server 2008, you should perform the following tasks to validate existing plan guides by using the **sys.fn\_validate\_plan\_guide** function. Alternatively, you can monitor for invalid plan guides by using the Plan Guide Unsuccessful event in SQL Server Profiler.

#### **Partitioned Table Queries That Use the USE PLAN Hint**

SQL Server 2008 changes the way queries on partitioned tables and indexes are processed. Queries on partitioned objects that use the USE PLAN hint might contain an invalid plan. We recommend the following procedures after upgrading to SQL Server 2008.

When the USE PLAN hint is specified directly in a query:

- 1. Remove the USE PLAN hint from the query.
- 2. Test the query.
- 3. If the optimizer does not select an appropriate plan, tune the query and then specify the USE PLAN hint with the desired query plan.

When the USE PLAN hint is specified in a plan guide:

1. Use the **sys.fn\_validate\_plan\_guide** function to check the validity of the plan guide. Alternatively, you can check for invalid plans by using the Plan Guide Unsuccessful event in SQL Server Profiler.

2. If the plan guide is invalid, drop the plan guide. If the optimizer does not select an appropriate plan, tune the query and then specify the USE PLAN hint with the desired query plan.

For more information about query processing on partitioned objects, see **Query Processing Enhancements on Partitioned Tables and Indexes**.

## ■ **REPLACE** Function

In SQL Server 2005, trailing spaces specified in the input parameters to the REPLACE function are trimmed. In SQL Server 2008, trailing spaces are preserved. For applications that rely on the previous behavior of the function, use the RTRIM function when specifying the input parameters for the function. For example, the following syntax will trim trailing spaces from each input parameter in the REPLACE function: REPLACE(RTRIM(*input\_parameter*), RTRIM(*input\_parameter*)).

## □ See Also

#### Reference

Breaking Changes to Database Engine Features in SQL Server 2008 Deprecated Database Engine Features in SQL Server 2008 Discontinued Database Engine Functionality in SQL Server 2008

#### **Other Resources**

SQL Server Database Engine Backward Compatibility ALTER DATABASE Compatibility Level (Transact-SQL)

## SQL Server Setup Help Full-Text Search Backward Compatibility



- -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Database Engine Backward</u> <u>Compatibility</u> >

Topics in the backward compatibility section describe changes in the behavior of full-text search between SQL Server 2005 and SQL Server 2008.

See	For topics about
Deprecated Full-Text Search Features in SQL Server 2008	Deprecated full-text search features in this release.
Discontinued Full-Text Search Features in SQL Server 2008	Discontinued full-text search features in this release.
Breaking Changes to Full-Text Search in SQL Server 2008	Full-text search changes that might require changes to applications.
Behavior Changes to Full-Text Search in SQL Server 2008	Other changes in behavior to full-text search.

## $\Box$ See Also

## **Other Resources**

SQL Server Database Engine Backward Compatibility

## SQL Server Setup Help Deprecated Full-Text Search Features in SQL Server 2008

# 

#### - -

Backward Compatibility > SQL Server Database Engine Backward Compatibility > Full-Text Search Backward Compatibility >

This topic describes the deprecated full-text search features that are still available in SQL Server 2008. These features are scheduled to be removed in a future release of SQL Server. Deprecated features should not be used in new applications.

You can monitor the use of deprecated features by using the **SQL Server:Deprecated Features** object performance counter and trace events. For more information, see <u>Using SQL Server Objects</u>.

# □ Features Not Supported in the Next Version of SQL Server

No full-text search features will be discontinued in the next release of SQL Server.
# □ Features Not Supported in a Future Version of SQL Server

The following full-text search 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.

The **Feature name** value appears in trace events as the **ObjectName** and in performance counters and **sys.dm\_os\_performance\_counters** as the instance name. The **Feature ID** value appears in trace events as the **ObjectId**.

Category	Deprecated feature	Replacement	
Full-text search	CREATE FULLTEXT CATALOG option: IN PATH ' <i>rootpath</i> ' ON EU ECROUR <i>filearou</i> n	None.	C N
Full-text search	DATABASEPROPERTYEX property: IsFullTextEnabled	None.	D
Full-text search	FULLTEXTCATALOGPROPERTY property: • LogSize • PopulateStatus	None.	F
Full-text search	FULLTEXTSERVICEPROPERTY property: • ConnectTimeout • DataTimeout • ResourceUsage	None.	F F
	<pre>sp_configure options:</pre>		ei 21

Full-text search	<ul><li>ft crawl bandwidth</li><li>ft notify bandwidth</li></ul>	None.	si
Full-text search	<pre>sp_detach_db option: [ @keepfulltextindexfile = ] 'KeepFulltextIndexFile'</pre>	None.	sī
Full-text search	sp_fulltext_catalog	CREATE FULL CATALOG ALTER FULLTEXT CATALOG DROP FULLTEXT CATALOG	sī
	sp_fulltext_column	CREATE FULL INDEX	sĮ
Full-text search	sp_fulltext_database	ALTER FULLTEXT INDEX	sţ
	sp_fulltext_table	DROP FULLTEXT INDEX	sĮ
	sp_help_fulltext_catalogs		sţ
Full-text search	sp_help_fulltext_catalog_components		sĮ
	sp_help_fulltext_catalogs_cursor	sys.fulltext_catalogs	sI
	sp_help_fulltext_columns	sys.fulltext_index_columns	sţ
	sp_help_fulltext_columns_cursor	sys.fulltext_indexes	sĮ
	sp_help_fulltext_tables		sĮ
	sp_help_fulltext_tables_cursor		sī
	<pre>sp_fulltext_service action values:</pre>		
Full-text search	<ul> <li>clean_up, connect_timeout , and data_timeout return zero</li> <li>resource_usage has no function.</li> </ul>	None	

Full-text search	<pre>sys.dm_fts_active_catalogs columns:     is_paused     previous_status     previous_status_description     row_count_in_thousands     status     status_description     worker_count</pre>	dı dı None. dı dı dı dı dı dı dı
Full-text search	<pre>sys.dm_fts_memory_buffers column:     row_count</pre>	None. <b>d</b> i
Full-text search	<ul> <li>sys.fulltext_catalogs columns:</li> <li>path</li> <li>data_space_id</li> <li>file_id columns</li> </ul>	fu None. fu fu

\* The **SQL Server:Deprecated Features** object does not monitor occurrences of CREATE FULLTEXT CATLOG ON FILEGROUP *filegroup*.

# □ See Also

### Reference

Deprecated Database Engine Features in SQL Server 2008

### Concepts

Breaking Changes to Full-Text Search in SQL Server 2008

### **Other Resources**

SQL Server, Deprecated Features Object

### SQL Server Setup Help Discontinued Full-Text Search Features in SQL Server 2008

-

Backward Compatibility > SQL Server Database Engine Backward Compatibility > Full-Text Search Backward Compatibility >

No full-text search features are discontinued in SQL Server 2008, though some have been deprecated. For more information, see <u>Deprecated Full-Text Search</u> <u>Features in SQL Server 2008</u>.

# □ See Also

### Reference

Discontinued Database Engine Functionality in SQL Server 2008

### Concepts

Breaking Changes to Full-Text Search in SQL Server 2008 Behavior Changes to Full-Text Search in SQL Server 2008

## SQL Server Setup Help Breaking Changes to Full-Text Search in SQL Server 2008

# 

#### - -

Backward Compatibility > SQL Server Database Engine Backward Compatibility > Full-Text Search Backward Compatibility >

This topic describes breaking changes in full-text search. These changes might break applications, scripts, or functionalities that are based on earlier versions of SQL Server. You might encounter these issues when you upgrade. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.

# <sup>□</sup> Breaking Changes

The following breaking changes apply to Full-Text Search in SQL Server 2008.

Feature	Scenario
<b>CONTAINSTABLE</b> with user-defined types (UDTs)	The full-text key is a SQL Server user- defined type, for example, MyType = char(1).
<i>top_n_by_rank</i> parameter (of the CONTAINSTABLE and <b>FREETEXTTABLE</b> Transact-SQL statements)	<i>top_n_by_rank</i> queries using 0 as the parameter.
CONTAINSTABLE and <b>ItemCount</b>	Delete rows from base table before it pushes changes to MSSearch.
ItemCount	Table contain null documents or type columns.
Catalog <b>ItemCount</b>	Blob column with a NULL extension.

UniqueKeyCount	Querying a unique key count from a catalog, for example, two tables (table1 and table2) each with three words: word1, word2, and word3.
<b>sp_fulltext_pendingchanges</b> when updating key column	Update the full-text key column on one row of a 2-row table, and run sp_fulltext_pendingchanges.
Inline functions	Inline functions with a full-text operator
sp_fulltext_database	Enable or disable full-text search by using sp_fulltext_database.
Locale-specific stop words	Queries in locale-specific variants of a parent language, such as Belgian French and Canadian French.

Thesaurus stemming process	Using thesaurus and Inflectional forms (stemming).
Full-text catalog path and filegroup	Working with full-text catalogs.
sys.fulltext_catalogs	Using the <b>path, data_space_id</b> , and <b>file_id</b> of this catalog view.
sys.sysfulltextcatalogs	Using the <b>path</b> column of this deprecated system table.

sp_help_fulltext_catalogs sp_help_fulltext_catalogs_cursor	Using the <b>PATH</b> column of these deprecated stored procedures.
sp_help_fulltext_catalog_components	Using <b>sp_help_fulltext_catalog_components</b> of this stored procedure.
DATABASEPROPERTY and DATABASEPROPERTYEX	Using the <b>IsFullTextEnabled</b> property.

# <sup>□</sup> See Also

### Concepts

Behavior Changes to Full-Text Search in SQL Server 2008

### **Other Resources**

Full-Text Search Obtaining Full-Text Property Values Using Transact-SQL Functions

# SQL Server Setup Help Behavior Changes to Full-Text Search in SQL Server 2008

# 

#### - -

Backward Compatibility > SQL Server Database Engine Backward Compatibility > Full-Text Search Backward Compatibility >

This topic describes behavior changes in full-text search. Behavior changes affect how features work or interact in SQL Server 2008 as compared to earlier versions of SQL Server.

In SQL Server 2008, the Full-Text Engine is integrated as a database service into the relational database as part of the server query and storage engine infrastructure. The new full-text search architecture achieves the following goals:

- Integrated storage and management—Full-text search is now integrated directly with the inherent storage and management features of SQL Server, and the MSFTESQL service no longer exists.
  - Full-text indexes are stored inside the database filegroups, rather than in the file system. Administrative operations on a database, such as creating a backup, automatically affect its full-text indexes.
  - A full-text catalog is now a virtual object that does not belong to any filegroup; it is a logical concept that refers to a group of fulltext indexes. Therefore, many catalog-management features have been deprecated, and deprecation has created breaking changes for some features. For more information, see <u>Deprecated</u> <u>Database Engine Features in SQL Server 2008</u> and <u>Breaking</u> <u>Changes to Full-Text Search in SQL Server 2008</u>.

### <sup>☑</sup> Note:

SQL Server 2005 Transact-SQL DDL statements that specify full-text catalogs work correctly.

• Integrated query processing—The new full-text search query processor is

part of the Database Engine and is fully integrated with the SQL Server Query processor. This means that, the query optimizer recognizes fulltext query predicates and automatically executes them as efficiently as possible.

- Enhanced administration and troubleshooting—Integrated full-text search provides tools to help you analyze search structures such as the full-text index, the output of a given word breaker, stopword configuration, and so forth. For more information, see **Troubleshooting Full-Text Search**.
- Stopwords and stoplists have replaced noise words and noise-word files. A stoplist is a database object that facilitates manageability tasks for stopwords and improves the integrity between different server instances and environments. For more information, see **Stopwords and Stoplists**.
- SQL Server 2008 includes new word breakers for many of the languages that exist in SQL Server 2005. Only the word breakers for English, Korean, Thai, and Chinese (all forms) remain the same. For other languages, if a full-text catalog was imported when a SQL Server 2005 database was upgraded to SQL Server 2008, one or more languages used by the full-text indexes in full-text catalog might now be associated with new word breakers that might behave slightly differently from the imported word breakers. For more information about how to guarantee a total match between queries and the full-text index content, see <u>Full-Text Search Upgrade</u>.
- A new FDHOST Launcher (MSSQLFDLauncher) service has been added. For more information, see **Getting Started with Full-Text Search**.
- Full-text indexing works with a FILESTREAM column in the same way that it does with a varbinary(max) column. The FILESTREAM table must have a column that contains the file name extension for each FILESTREAM BLOB. For more information, see Querying varbinary(max) and xml Columns, Full-Text Search Filters, Full-Text Indexing and Querying Process, and sys.fulltext\_document\_types (Transact-SQL). The full-text engine indexes the contents of the FILESTREAM BLOBs. Indexing files such as images might not be useful. When a FILESTREAM BLOB is updated it is reindexed.

### <sup>Z</sup> Important:

For essential information about the impact of the new full-text search architecture on existing applications and scripts, we strongly recommend that you see <u>Deprecated Database Engine Features in SQL Server 2008</u> and <u>Breaking Changes to Full-Text Search in SQL Server 2008</u>.

### □ See Also

Concepts Full-Text Search Upgrade

Other Resources Full-Text Search Architecture Full-Text Search Backward Compatibility Full-Text Search Fundamentals Getting Started with Full-Text Search

SQL Server Setup Help SQL Server Analysis Services Backward Compatibility

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> >

The topics in this section describe the changes in behavior between versions of Microsoft SQL Server Analysis Services

# <sup>□</sup> In This Section

Topics	Description
Discontinued Analysis	Describes features that existed in earlier versions of SQL
Services Functionality	Server Analysis Services but that have been removed in
in SQL Server 2008	later versions.
Deprecated Analysis Services Functionality in SQL Server 2008	Describes features that have been retained in SQL Server Analysis Services for backward compatibility, but which will be removed in a future version of SQL Server.
<u>Behavior Changes to</u> <u>Analysis Services</u> <u>Features in SQL Server</u> 2008	Describes features that have changed in SQL Server Analysis Services.
<u>Breaking Changes to</u> Analysis Services Features in SQL Server 2008	Describes issues that you might encounter when you upgrade SQL Server Analysis Services.
© 2008 Microsoft Corporation All rights reserved	

#### SQL Server Setup Help

### Deprecated Analysis Services Functionality in SQL Server 2008

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Analysis Services Backward</u> <u>Compatibility</u> >

This topic describes the deprecated Analysis Services features that are still available in SQL Server 2008. These features are scheduled to be removed in a future release 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	
MDX	NON_EMPTY_BEHAVIOR query optimizer hint was turned on by default	The NON_EMPTY_BEHAVIOR query optimizer hint will be turned off by default in a future release. It is an MDX optimization hint that can produce incorrect results when it is not used correctly
Other	CELL_EVALUATION_LIST intrinsic cell property	Blank in SQL Server Analysis Services.
Objects	COM assemblies	COM assemblies might pose a security risk.

# $\Box$ See Also

### **Other Resources**

SQL Server Analysis Services Backward Compatibility

#### SQL Server Setup Help

#### Discontinued Analysis Services Functionality in SQL Server 2008

- -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Analysis Services Backward</u> <u>Compatibility</u> >

This topic describes Analysis Services features that are no longer available in SQL Server 2008.

# <sup>□</sup> 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.
Othor	Active Directory	
Oulei	Registration	
Other	Skipped levels in parent- child hierarchies	The <b>SkippedLevelsColumn</b> property, which indicates how many levels to skip when creating a member under a parent in a parent-child hierarchy, is no longer
		The Surface Area Configuration Tool is
Tools	Surface Area	discontinued for SQL Server 2008. For
	Configuration Tool	more information, see <u>Backward</u> <u>Compatibility</u> .

# $\Box$ See Also

### **Other Resources**

SQL Server Analysis Services Backward Compatibility

SQL Server Setup Help

Breaking Changes to Analysis Services Features in SQL Server 2008



### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Analysis Services Backward</u> <u>Compatibility</u> >

This topic describes breaking changes in Analysis Services. These changes might break applications, scripts, or functionalities that are based on earlier versions of SQL Server. You might encounter these issues when you upgrade. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.

# Breaking changes to Analysis Services in SQL Server 2008

The following table lists the issues that you might encounter when you upgrade a SQL Server 2005 Analysis Services (SSAS) Analysis Services database to SQL Server 2008.

Issue Type	Issue Description
The shallow	
exists function	In SQL Server 2005 Analysis Services (SSAS), the shallow ex
now works	function did not work with named sets that contained enumera
differently with	members or crossjoins of enumsets. For backward compatibilit
named sets that	the original release version and SP1 of SQL Server 2005 Analy
contain	Services (SSAS), set the configuration property
enumerated	"ConfigurationSettings\OLAP\Query\NamedSetShallowExists
members or	to 1, or for backward compatibility with SQL Server 2005 Ana
crossjoins of	Services (SSAS) SP2, set it to 2.
enumsets.	
VBA functions	
handle null	
values and empty	
values differently	In SQL Server 2005 Analysis Services (SSAS), VBA functions
than they were	returned 0 or an empty string when either null values or empty
handled in SQL	values were used as arguments. In SQL Server 2008, they will
Server 2005	null.
Analysis Services	
(SSAS) Analysis	
Services.	
	By default, SQL Server 2008 does not install the DSO (Decisio
	Support Objects) backward compatibility component. The back
	compatibility package is installed by default but the DSO com
	of the package will be disabled. Since the SQL Server Analysis
	Services Migration Wizard relies on this component, it will fai
	unless the component is installed. To install the DSO compone
	ule following:

	1. Open Control Panel.
	2. In Windows XP or Windows Server 2003, select <b>Add or</b>
	<b>Remove Programs</b> . In Windows Vista and Windows Serv
	2008, select <b>Programs and Features</b> .
	3. Right-click Microsoft SQL Server 2005 Backward
	<b>Compatibility</b> , and select <b>Change</b> .
	4. In the Backward Compatibility Setup wizard, click <b>Next</b> .
The Migration	5. On the Program Maintenance page, select <b>Modify</b> , and th
Wizard will fail	click <b>Next</b> .
because DSO is	6. On the Feature Selection page, if Decision Support Objec
not installed by	(DSO) is not available, click the down arrow and select <b>T</b>
Default.	feature will be installed on local hard drive. Click Next
	7. On the Ready to Modify the Program page, click <b>Install</b> .
	8. When installation is finished, click <b>Finish</b> .
	You can remove DSO after migration is complete by following
	previous steps, changing the option for DSO to " <b>This feature</b>
	not be available."
	If the backward compatibility package is not installed, you can
	it from the SQL Server 2008 distribution media. Note that ther
	versions for each target architecture (x86, x64, ia64). These ve
	can be found at the following locations:
	x86\Setup\x86\SQLServer2005_BC.msi
	x64\Satup\x64\SQLSamor2005_BC_msi
	x04/Setup/x04/SQLServerz005_bC.IIISi
	ia64\Setup\ia64\SQLServer2005_BC.msi
	The server manages the Data folder and creates or drops folder
	objects are created, deleted, and altered. Therefore, specifying
It is not	partition storage location inside the Data folder is strongly
recommended to put the partition location in the Data folder.	discouraged, especially in the subfolders for databases, cubes,
	dimensions. Although the server allows you to do this with Cre
	Alter, it will display a warning. When you upgrade databases f
	Analysis Services 2005 to Analysis Services 2008 that have pa
	storage locations in the Data folder, it will work. Restore or Sy
	will require that you move partition storage locations outside the

You might get	
unexpected	
results for queries	ProClarity Analytics Server and Microsoft Office Performance
that use the	Server 2007 use the EXISTING keyword in MDX incorrectly i
"EXISTING"	certain scenarios. Due to software updates that were made in
MDX keyword in	Analysis Services 2008, these queries might return unexpected
ProClarity	results. ProClarity Analytics Server and Microsoft Office
Analytics Server	PerformancePoint Server 2007 will be issuing a service release
and Microsoft	address this issue.
Office	
PerformancePoint	
Server 2007.	

# Breaking changes to Analysis Services in SQL Server 2005

The following table lists the issues that you might encounter when you upgrade a SQL Server 2000 Analysis Services database to SQL Server 2008.

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 2008 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. For example, an OLAP mining model based on a linked cube cannot be migrated because the linked cube on which the mining model 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 2008 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, because autoexist functionality automatically removes tuples that do not exist in the dimension from any cross-join of sets containing members from the migrated hierarchies. To resolve this issue, you should review calculations that involve multiple hierarchies in the same dimension
Browsing experience is different when disabled levels are used	Since SQL Server 2005 Analysis Services, hidden or disabled levels in hierarchies are no longer supported. Hidden or disabled levels are migrated as visible and enabled levels. Calculations that involve hierarchies containing such levels might return unexpected results. After upgrading, review and verify calculations that involve hierarchies that previously contained hidden or disabled levels.
Bucketing may be different for grouping levels	Since Microsoft SQL Server 2005 Analysis Services (SSAS), automatic grouping in might return a different set of member groups. Calculations that rely on these member groups might return unexpected results. After upgrading, review and verify

	calculations that rely on member groups.
	In SQL Server 2000 Analysis Services and earlier versions, Analysis Services used only neutral language identifiers, also known as primary language identifiers, for example LANG_ENGLISH (0x09) for English and LANG_CHINESE (0x04) for Chinese.
Conversion from neutral language to specific language may produce unexpected results	To support translation and collation options, Analysis Services now uses specific language identifiers, which are a combination of a primary language identifier and a sublanguage identifier used for a specific culture. For example, the combination of the primary language identifier LANG_ENGLISH (0x09) and the sublanguage identifier SUBLANG_ENGLISH_AUS (0x03) describes Australian English.
	Migrating from neutral to specific language identifiers can change the expected translation and collation behavior, producing unexpected results. After upgrading, review and validate objects such as dimensions, hierarchies, and members for which the language identifier has changed.
Cube role commands are not supported	Microsoft SQL Server 2008 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	In previous versions of Analysis Services, the following properties are supported on dimensions and levels: custom rollup formulas, custom member formulas, All member formulas, and custom level formulas. In SQL Server 2005 Analysis Services, these properties were replaced by functionality supported in Multidimensional Expressions (MDX) scripts, and are upgraded to MDX scripts during migration.
Custom	Aggregations that were manually generated in previous versions

aggregations	of Analysis Services are not migrated in Microsoft SQL Server
migrated	Storage Design Wizard are migrated. To resolve this issue, manually create the aggregations by using XMLA scripts.
Data members always exist in parent-child dimensions	Previous versions of Analysis Services gave you the option of not including data members in parent-child dimensions. The <b>DataMembers</b> property of a dimension in previous versions of Analysis Services supported three options: <b>None</b> , <b>Hidden</b> , or <b>Visible</b> . The <b>None</b> option is not available in Microsoft SQL Server 2008 Analysis Services. Data members are always included in parent attributes. To hide the data members in parent-child dimensions, you can set the <b>DataMembers</b> property of a dimension to <b>Hidden</b> . The <b>MembersWithData</b> property for the parent attribute supports only two options: <b>NonLeafDataHidden</b> or <b>NonLeafDataVisible</b> .
Database role commands are not supported	Microsoft SQL Server 2008 Analysis Services does not support command objects on database roles and will not migrate commands from previous versions of Analysis Services.
DefaultMember is migrated into MDX script	In previous versions of Analysis Services, the default member of a dimension is specified by a Multidimensional Expressions (MDX) expression, contained in the <b>DefaultMember</b> property of the dimension. In Microsoft SQL Server 2005 Analysis Services (SSAS), this property was replaced by functionality supported in MDX scripts, and the property is upgraded to a MDX script during migration.
Dimension and hierarchy renaming by migration may cause different query results	Dimension hierarchies in SQL Server 2000 Analysis Services are internally represented as separate dimensions, and a naming convention is used to identify them. Migration to Microsoft SQL Server 2008 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 earlier versions of Analysis Services.
Drillthrough settings are not migrated	While drillthrough exists in Microsoft SQL Server 2008 Analysis Services, drillthrough settings are not migrated from previous versions of Analysis Services.
Hidden levels are visible after	Analysis Services no longer supports hidden or disabled levels in hierarchies. Hidden or disabled levels are migrated as visible

migration	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 (SSAS), this feature has been replaced by linked dimensions and linked measure groups.
Member unique names may change during migration	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.
Remote partitions are not migrated	Remote partitions are not migrated from Microsoft SQL Server 2000 Analysis Services to Microsoft SQL Server 2008 Analysis Services. Upgrade the server to SQL Server 2008 Analysis Services and then manually create the remote partitions.
Some mining model algorithm parameters are not supported	Previous versions of Analysis Services support using the MINIMUM_LEAF_CASES parameter with the Microsoft Decision Trees algorithm, and the MINIMUM_CLUSTER_CASES parameter with the Microsoft Clustering algorithm. Since SQL Server 2005 Analysis Services, both of these parameters have been renamed to MINIMUM_SUPPORT. If these parameters were used in mining models created using the previous version of Analysis Services , the parameters are not migrated.
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 (SSAS), certain MDX functions are either no longer supported, or behave differently than in previous versions of Analysis Services.
The CREATE	

KPI Command Introduces a New Keyword	A new keyword, <b>KPI</b> , has been introduced to the CREATE KPI command. If existing objects have the name KPI, then the new keyword will conflict with them.
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	Since Microsoft SQL Server 2005 Analysis Services (SSAS), unary operators on hierarchies other than parent-child hierarchies are not supported 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 (SSAS), this feature was 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 was updated in Microsoft SQL Server 2005 Analysis Services (SSAS). Data from writeback tables constructed by previous versions of Analysis Services cannot be migrated into the new writeback table structure.

# $\Box$ See Also

### **Other Resources**

SQL Server Analysis Services Backward Compatibility

SQL Server Setup Help

Behavior Changes to Analysis Services Features in SQL Server 2008



### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>SQL Server Analysis Services Backward</u> <u>Compatibility</u> >

This topic describes behavior changes in Analysis Services. Behavior changes affect how features work or interact in SQL Server 2008 as compared to earlier versions of SQL Server.
# Behavior Changes

Feature	SQL Server 2000 Behavior	SQL Server 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 level 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		

Other AutoCommit and AutoRollback in update.	transaction fail. In SQL Server 2000 Analysis Services, the Update statement created its own transaction. In SQL Server Analysis Services, it does not.
--	---

### $\Box$ See Also

### **Other Resources**

SQL Server Analysis Services Backward Compatibility

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Reporting Services Backward Compatibility



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> >

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.

# <sup>□</sup> In This Section

- Deprecated Features in SQL Server 2008 Reporting Services
- <u>Discontinued Functionality in SQL Server 2008 Reporting Services</u>
- Breaking Changes in SQL Server 2008 Reporting Services
- Behavior Changes in SQL Server 2008 Reporting Services

About Report Authoring and Publishing in SQL Server 2008

In this release, new reports are authored using a new version of the Report Definition Language (RDL) schema. Reports using the new schema must be authored in the new stand-alone Report Designer application, not the Report Designer that is integrated into SQL Server Business Intelligence Development Studio.

The new stand-alone Report Designer application provides a way to author reports with new features, such as Tablix data regions and expanded Chart data regions. Reports that use the new schema cannot be published to previous versions of Reporting Services report servers. To view reports that use the new RDL schema, you must deploy them (using the Report Designer application) or upload them (using Report Manager) to a SQL Server 2008 Reporting Services report server.

For more information about RDL, see **Report Definition Language Reference**. For more information about report authoring, see **What's New in Report Authoring**.

About Report Processing and Rendering in SQL Server 2008

In this release, the report processing engine has been redesigned to process and render report pages only when they are viewed or exported. For more information, see **What's New in Report Processing and Rendering**.

In this release, the report rendering object model has been replaced. For more information, see <u>Breaking Changes in SQL Server 2008 Reporting Services</u>.

### $\Box$ See Also

### **Concepts** Backward Compatibility

### Other Resources Upgrade (Reporting Services)

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Deprecated Features in SQL Server 2008 Reporting Services

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Reporting Services Backward Compatibility</u> >

This topic describes the deprecated Reporting Services features that are still available in SQL Server 2008. These features are scheduled to be removed in a future release of SQL Server. Deprecated features should not be used in new applications.

### $\Box$ See Also

### Concepts

Backward Compatibility

#### **Other Resources**

Upgrade (Reporting Services) Reporting Services Backward Compatibility

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Discontinued Functionality in SQL Server 2008 Reporting Services

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Reporting Services Backward Compatibility</u> >

This topic describes Reporting Services features that are no longer available in SQL Server 2008. It does not include announcements about discontinued support for specific versions of the operating system or Microsoft Internet Information Services (IIS). For more information about system prerequisites, see <u>Hardware</u> and Software Requirements for Installing SQL Server 2008.

### <sup>□</sup> Manual Upgrade for Report Server Database

Starting in SQL Server 2005 Service Pack 2 (SP2), the Report Server service is able to auto-detect the version of the report server database and upgrade it to the schema that matches the version of the current report server instance. When an older version of the database is detected, you are prompted to upgrade it. If you proceed to upgrade it, the schema is updated to the new format and you cannot roll it back to a previous format.

Because the auto-upgrade feature has been added, there is no longer any need for you to create a database upgrade script or for having a manual upgrade option in the Reporting Services Configuration tool. In SQL Server 2008 Reporting Services (SSRS), the button to create the script and the button that performs the upgrade action are removed from the Database Setup page in the Reporting Services Configuration tool. For more information about upgrade support, see Upgrading a Report Server Database.

# □ Internet Explorer 5.5 Support

Deprecation of Microsoft Internet Explorer 5.5 support was announced in Reporting Services 2005 Service Pack 2. This version of Internet Explorer is not supported in the SQL Server 2008 release of Reporting Services.

# <sup>□</sup> SQL Server 2000 Report Server Web Service Endpoint

The SQL Server 2000 Report Server Web service endpoint is discontinued in this release. For more information about current endpoints, see **Report Server Web Service Endpoints** in SQL Server Books Online.

# BITML OWC Rendering Extension

The HTML with Office Web Components (OWC) rendering extension is discontinued in this release.

# <sup>B</sup> HTML 3.2 Rendering Extension

The HTML 3.2 format in the HTML rendering extension is discontinued in this release. The rendering extension is no longer included in a Reporting Services installation.

### <sup>©</sup> Report Builder Runs in Full Trust Mode Only

For Reporting Services in native mode, running Report Builder in partial trust mode is discontinued in this release. Reporting Services in either native mode or SharePoint mode supports running Report Builder in full trust mode only.

# <sup>□</sup> Surface Area Configuration Tool

The Surface Area Configuration Tool is discontinued for SQL Server 2008. For more information, see <u>Backward Compatibility</u>.

# <sup>□</sup> Notification Services

Notification Services has been removed from SQL Server 2008. You can use existing Reporting Services functionality, such as data-driven subscriptions, to obtain some of the functionality that was provided by Notification Services in previous versions of SQL Server.

### $\Box$ See Also

### Concepts

Backward Compatibility

#### **Other Resources**

Upgrade (Reporting Services) Reporting Services Backward Compatibility

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Breaking Changes in SQL Server 2008 Reporting Services

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Reporting Services Backward Compatibility</u> >

This topic describes breaking changes in Reporting Services. These changes might break applications, scripts, or functionalities that are based on earlier versions of SQL Server. You might encounter these issues when you upgrade, or in custom scripts or reports. For more information, see <u>Using Upgrade Advisor</u> to <u>Prepare for Upgrades</u>.

**Report Server Breaking Changes** 

**Report Builder Breaking Changes** 

**Report Processing Breaking Changes** 

**Report Rendering Breaking Changes** 

For more information about new features, see **What's New (Reporting Services)**.

# <sup>□</sup> Report Server Breaking Changes

This section describes breaking changes to the report server and management tools.

Feature	Description			
	Reporting Services no longer depends on IIS to provide access to the SOAP endpoint. URLs no longer include Web sites in IIS. Reporting Services uses HTTP.SYS directly to listen for requests on a specific port that you define for report server URLs.			
	This enhancement is a breaking change for some deployments:			
IIS and ASP.NET	<ul> <li>If you have scripts, tools, or diagnostic processes that include reviewing IIS metadata or properties, you must now develop new approaches for managing a report server deployment.</li> <li>If you implemented an ISAPI filter for security, you must move the ISAPI filter to be hosted in ISA server or create a new HTTP Module that does the same operations in Reporting Services.</li> <li>If you use custom virtual directory settings, you might not be able to configure equivalent settings or URLs in the new report server implementation. In some cases, upgrade operations cannot create equivalent URLs for the report server or Report Manager.</li> </ul>			
	Upgrade Advisor will detect breaking changes by checking for ISAPI filters and customized virtual directories. Upgrade Advisor cannot check for all possible customizations. Your installation might pass the Upgrade Advisor check but still break or produce unexpected errors.			
	On supported editions of 32-bit Windows XP SP2, IIS 5.1 and Reporting Services cannot use the same port. You cannot configure both IIS 5.1 and a report server to listen on the default HTTP port (port 80).			

Port conflicts on Windows XP	<ul> <li>IIS 5.1 does not use HTTP.SYS for Web applications hosted on the Web server. This means there is no common queue management for requests that come over the same port, and there is no common repository of registered and reserved URLs.</li> <li>This issue results in the following behavior for SQL Server 2008 Reporting Services upgrades on Windows XP: <ul> <li>On 32-bit editions of Windows XP, if you upgrade an existing SQL Server 2005 Reporting Services, installation to SQL Server 2008 Reporting Services, the report server is configured to listen on port 8080.</li> <li>On 64-bit editions of Windows XP, if you upgrade an existing SQL Server 2005 Reporting Services installation to SQL Server 2008 Reporting Services, the report server is configured to listen on port 8080.</li> <li>On 64-bit editions of Windows XP, if you upgrade an existing SQL Server 2008 Reporting Services, the report server is configured to listen on port 80.</li> <li>If you perform a SQL Server 2008 build-to-build upgrade, the report server continues to listen on the same port that was configured prior to the upgrade.</li> </ul> </li> <li>After upgrade completes, you can use the Reporting Services Configuration tool to change the port on which the report server listens if you want to use a different port.</li> <li>For more information about supported Windows operating systems for SQL Server 2008, see Hardware and Software Requirements for Installing SQL Server 2008.</li> </ul>
Reporting Services	The Reporting Services Windows Management Instrumentation (WMI) Provider is not compatible with the previous version. The new version includes additional methods to support URL registration. Because there can only be one version of the Reporting Services WMI provider for a report server installation, this version replaces the previous version. This

Windows Management Instrumentation (WMI) Provider	change represents a breaking change for some deployments. If you created script or tools that call the WMI provider, you must revise your code to use the new version. For more information, see <b>Reporting Services WMI Provider</b> . This change also prevents users from connecting to a SQL Server 2005 instance in SQL Server Management Studio when the user specifies the <i><server_name< i="">&gt;<i>\<instance_name< i="">&gt; format to connect. Instead, users must type the report server URL to connect.</instance_name<></i></server_name<></i>
Consolidation of services and applications	The Report Server Web service, Report Manager, and the background processing application are consolidated into a single service. You cannot start or stop them separately.
Reporting Services configuration files	Reporting Services configuration files are also consolidated. The RSReportServer.config file is the primary configuration file for Report Manager and the Report Server Web service. The RSWebApplication.config file is obsolete. The following RSWebApplication.config settings have been moved to the RSReportServer.config file: • ReportServerUrl • ReportServerUrl • ReportServerExternalUrl • ReportBuilderTrustLevel • DeliveryUI settings for delivery extensions • DisplayErrorLink The following settings are obsolete and are no longer used: • ReportServerVirtualDirectory • MaxActiveReqForOneUser If you modified the RSWebApplication.config file in a previous installation, the file will not be deleted when you upgrade to SQL Server 2008. You should delete the file manually; all settings within the file are ignored in this release.
Reporting Services trace logs	ReportServerService_ <timestamp>.log is the primary trace log for all applications that run in the service. The following files are obsolete and are no longer created in SQL Server 2008: ReportServerWebApp_<timestamp>.log,</timestamp></timestamp>

	ReportServer_ <timestamp>.log, and</timestamp>
	ReportServerService_main_ <timestamp>.log.</timestamp>
Reporting Services Configuration tool	Reporting Services Configuration tool no longer supports the Upgrade Database or Grant Rights features that allowed you to upgrade or grant permissions as independent operations or generate script templates for performing these tasks. In this release, both upgrading and database permissions are handled as internal operations.
SQL Server	In Management Studio, the Home folder is removed in this
Management Studio	release. You cannot view, manage, distribute or secure report server content in Management Studio.
Report Manager	In Report Manager, the following links are removed from Site Settings: Configure item-level role definitions, Configure system-level role definitions, Manage jobs. Report Manager no longer supports the ability to create, modify, or delete role definitions. You must use Management Studio to manage which tasks are in specific roles. Similarly, job management has moved from Report Manager to Management Studio.
E-mail subscriptions	E-mail subscriptions will not work for e-mail aliases in the <b>Sender</b> , <b>To</b> , <b>Cc</b> , <b>Bcc</b> , and <b>Reply-To</b> fields when the report server or the remote SMTP server is upgraded to Windows Vista or Windows Server 2008. This issue occurs because Windows Server 2003 contains a feature that resolves aliases to their full e-mail addresses. Reporting Services depended on that feature to allow for using e-mail aliases instead of full e-mail addresses. However, to help filter out false e-mail addresses, Windows Vista and Windows Server 2008 do not contain this feature. To work around this issue, you must configure the <b>DefaultHostName</b> property in configuration. For information about resolving this issue, see the Microsoft Knowledge Base article <u>945601</u> : "SQL Server 2005 Reporting Services e-mail subscriptions do not work in Windows Vista and in Windows Server 2008 if you use aliases as e-mail addresses."
	The SQL Server 2008 Reporting Services Add-in for SharePoint Technologies provides report rendering, processing, management capabilities, and data-driven subscriptions when

SQL Server 2008 Reporting Services Add- in for SharePoint Technologies	you run a SQL Server 2008 report server instance in SharePoint Integrated mode. The add-in download contains a Report Viewer Web part, Web application pages, and support for using either Windows SharePoint Services (WSS) or Microsoft Office SharePoint Services (MOSS). The SQL Server 2008 Reporting Services Add-in for SharePoint
	Technologies requires a SQL Server 2008 report server instance because this add-in is not supported with earlier versions of SQL Server. If you have a pre-SQL Server 2008 report server, and you install or upgrade to the SQL Server 2008 Reporting Services Add-in for SharePoint Technologies, the report server will not function as expected. For example, you will be unable to configure database access by using the <b>Grant Database</b>
	Access page and render reports using a SharePoint product or technology. To resolve this issue, you must either upgrade your report server instance to SQL Server 2008, or you must uninstall the SQL Server 2008 Reporting Services Add-in for SharePoint Technologies and re-install the SQL Server 2005 Reporting Services Add-in for SharePoint Technologies.
	For more information about the SQL Server 2008 Reporting Services Add-in for SharePoint Technologies, see the <u>Microsoft</u> <u>SQL Server 2008 Reporting Services Add-in Readme</u> .
Basic authentication	In SQL Server 2008 Reporting Services, only NETWORK and NETWORK_CLEARTEXT logon types are supported with Basic authentication; Interactive and BATCH logon types are not supported.

# Report Builder Breaking Changes

This section describes breaking changes to Report Builder.

### **Report Builder Runs in Full Trust Mode Only**

In earlier versions of Reporting Services running in native mode, SQL Server 2005 Report Builder could be started using the following URLS:

- **Full trust** For example, http://<*servername*>/reportserver/reportbuilder/reportbuilder.application
- **Partial trust** For example, http://<*servername*>/reportserver/reportbuilder/reportbuilderlocalintranet.ap

For both URLs, *<servername>* is the name of the computer that specifies the report server. For both URLs, **reportserver** is the name of the report server instance.

In this release, you must use the full trust URL to run Report Builder. When you use the full trust URL for the first time, you might be prompted to grant a higher level of permissions for the application.

#### <sup>I</sup>Note:

If Report Builder does not run, or if you get an error, contact the system administrator. You might not have the permissions that you need to grant a higher level of trust for this application.

After you grant these permissions the first time, you do not have to set them again.

In this release, if you use the partial trust URL, the following error appears when you open or save a report, or switch report servers:

"Failed. An error occurred while processing your request. Save your report and restart the application."

# Report Processing Breaking Changes

Report processing architecture is fundamentally changed in this release by providing on-demand report processing. On-demand report processing significantly reduces memory usage on a report server.

### **RDL Upgrade Breaking Changes**

The following RDL elements are not supported when you upgrade an existing report:

• **Object Identifiers in RDL limited to 256 characters** Identifiers for objects in RDL (for example, **textboxID**) were previously unrestricted in length. In this release, the length of object identifiers is restricted to 256 characters. Identifiers still must be CLS-compliant.

#### **Interactivity Information Saved Only For the Last Request**

In earlier versions of Reporting Services, snapshots saved all possible combinations of interactive choices, such as drillthrough information and toggle choices. You could view page five of a report, but programmatically toggle an item on page one by keeping track of the correct ID for the toggle.

In SQL Server 2008, interactivity information is generated and saved only for the last rendering request. You cannot view a page and programmatically toggle an item on another page. You can only toggle drill down items on the current report page.

#### **Report Object Model Namespace Change**

In this release, the Report Object Model namespace has changed. This namespace provides read-only access from custom code to global collections like **Fields**, **Parameters**, and **ReportItems**. If existing custom code explicitly uses a fully qualified reference to an earlier namespace, this change is a breaking change.

It is recommended that you do not use fully qualified references to access builtin collections from your code. By not explicitly specifying the namespace, custom code references resolve to the version of the report object model for the currently installed version of Reporting Services.

For example, in SQL Server 2005 Reporting Services, the following two examples contrast the use of a fully qualified reference to a parameter (Microsoft.ReportingServices.ReportProcessing.ReportObjectModel.Para and a default reference to a parameter (Parameter) for passing a report parameter to a custom function that returns the report parameter label.

The following example is not recommended. It shows the fully qualified reference for a report parameter. The second example is recommended and shows the default namespace reference for a report parameter.

DO NOT USE FULLY QUALIFIED NAMESPACE REFERENCES FOR RUN-TIME COLLECTIONS

Public Function ShowParams(ByVal reportparameter As \_

Microsoft.ReportingServices.ReportProcessing.ReportObjectModel.Param

—

As String

Return reportparameter.Label

End Function

The recommended way to access a run-time collection is shown in the following example:

Public Function ShowParams(ByVal reportparameter As Parameter) \_

As String

Return reportparameter.Label

End Function

# Report Rendering Breaking Changes

Report rendering architecture is fundamentally changed in this release to provide more consistent rendering for paging and layout among different renderers.

#### New Rendering Object Model and Consistent Pagination

The Rendering Object Model (ROM) has changed for SQL Server 2008. Earlier versions of the rendering object model are no longer supported. Accessing the Rendering Object Model from a multi-threaded rendering extension (and switching context from multiple threads) is not supported.

The new ROM makes the rules for rendering pages more consistent. For more information, see **Understanding Pagination in Reporting Services**.

#### **Redesigned CSV Data Renderer**

In earlier versions of Reporting Services, when you exported a report to a CSV file format, the data was formatted in a way that preserved the way the data appeared on the report page. For matrix data regions, this resulted in a data format that was inconvenient to import into other applications in order to continue to work with the data.

In this release, when you export a report to a CSV file, you can choose between two supported formats: Default mode and Compliant mode. Default mode is optimized for Excel. Compliant mode is optimized for third-party applications. For more information, see **Exporting to a CSV File**.

The earlier format for CSV files is no longer available. However, for reports that do not use matrix data regions, you can use Compliant mode to get a file format closest to the earlier CSV file format.

# Aggregates with Conditional Visibility in Page Headers and Footers

In earlier versions of Reporting Services, different renderers used different rules

to determine which items with conditional visibility to include on a report page. For example, aggregate calculations were not performed for hidden items in printed reports, but were calculated for hidden items in reports that you viewed with a browser or in Excel.

In this release, all renderers use the same set of rules to determine which items are on a page.

### No Formula Support in Excel

In earlier versions of Reporting Services, there was limited support for translating expressions in RDL to Microsoft Excel formulas. In this release, when you export a report to Excel, RDL expressions are not translated to Excel formulas.

### **Overlapping Items**

In earlier versions of Reporting Services, if a report had overlapping items on the report design surface, publishing the report produced a warning ("Overlapping report items are not supported in all renderers."), but the report items remained in their original location on the design surface. In SQL Server 2008, report items may be moved to correct overlapping boundaries when a report is viewed or exported to a renderer based that does not support overlapping items. For more information, see **Understanding Rendering Behaviors**.

### □ See Also

#### Concepts

Behavior Changes in SQL Server 2008 Reporting Services

#### Other Resources Upgrade (Reporting Services) Reporting Services Backward Compatibility

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Behavior Changes in SQL Server 2008 Reporting Services



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Reporting Services Backward Compatibility</u> >

This topic describes behavior changes in Reporting Services. Behavior changes affect how features work or interact in SQL Server 2008 as compared to earlier versions of SQL Server.

**Report Server Configuration and Management Tools** 

**Report Authoring** 

**Report Processing** 

**Report Rendering** 

# Behavior Changes for Report Server Configuration and Management Tools

Reporting Services includes several tools and applications that you use to configure the server and manage content and operations. In this release, each tool is aligned to a specific purpose: configuration, administration, and content management. To promote consistency within a tool and to remove overlapping functionality, features and tasks have been added and removed from tools. If you were accustomed to using a tool to perform a given task, you might now need to use a different tool to accomplish the same task.

Feature	Description				
	Color-code status icons have been removed. New URL				
Reporting	configuration pages replace the pages for creating virtual				
Services	directories. The workflow for creating and configuring a report				
Configuration	server database has been revised. You now use a wizard to create				
	or update database connection.				
SQL Server	Management Studio supports only server administration tasks.				
Management	You can connect to and configure a report server that runs in				
Studio	native mode or in SharePoint integrated mode.				
	Report Manager is used to view and manage report server				
	content. This release introduces the ability to manage report				
	models. You can now set model item security and associate				
	clickthrough reports to entities in a model.				
Report					
Manager When viewing a report in Report Manager, because of the					
	changes introduced by on-demand report processing, the toolbar				
	displays a page estimate with a question mark instead of the				
	actual number of pages for a report. You can still click the Last				
	Page button and navigate to the end of the report.				

#### **Tasks Supported by Tools**

Tasks	Report Server	Management	Report	Command
	Configuration	Studio	Manager	line

				utilities
Reserve URLs	Х			Х
Set the service account and password	Х			X
Create the report server				
database or change	Х			Х
connection information				
Configure report server	v			v
scale-out	Λ			Λ
Back up, restore, change	x			x
keys or delete encrypted data	21			Λ
Configure unattended	x			x
execution account				
Configure report server e-	х			
mail				
Enable My Reports		X		
Enable logging on report		Х		
execution				
Enable client-side printing		X		
Set server defaults for report history		Х		
Create or modify role		x		
definitions		$\Lambda$		
View status of a running				
report or model process and		Х		
stop it if it is taking too long				
Grant permissions to report				
server items and operations			Х	
by creating role assignments				
Define and manage the				
report server folder hierarchy			Х	
View reports, report models,				
shared data sources,			Х	
resources, and folders				
Upload report definition			$\mathbf{V}$	
(.rur), report model (.smdl),			Λ	

and resource files			
Create and manage shared schedules	х	Х	
Create and manage linked reports		Х	
Create and manage report history		Х	
Create and manage shared data sources, and any data source properties defined in an individual report		x	
Schedule when data processing occurs for a report or configure a report to run as a report execution snapshot		x	
Subscribe to report deliveries and create and manage data- driven subscriptions		х	
Create data-driven subscriptions		Х	
Use Report Builder to create, modify, and save reports		Х	
Generate models, associate clickthrough reports to entities in a model, and set model item security		X	

# <sup>□</sup> Report Authoring

In earlier versions of Reporting Services, the four data regions (Table, Matrix, List, and Chart) were distinct report items with their own layout behavior and properties. In this release of Reporting Services, the Table, Matrix, and List data regions have been replaced by a new flexible grid layout called a Tablix data region that uses predefined templates to create the former data regions. The Tablix data region enables you to combine aspects of tables and matrices into flexible report layouts. The Chart data region remains a separate report item. New chart types, such as Polar, Radar, and Funnel, have been added to the Chart data region. For more information about the new chart types, see **Working with Tablix Data Regions**.

#### Preserving White Space in a Report Body or Rectangle Container

Extra white space is no longer removed by default. When you render a report that had extra white space on the report body when viewed on the report design surface, the trailing white space after the last report item on the page is preserved. This may result in more pages for an existing report. To remove the white space, set the report property **ConsumeContainerWhitespace** to **true**.

For more information, see What's New in Report Authoring.

# <sup>□</sup> Report Processing

Report processing has been redesigned in SQL Server 2008. Reports are now processed and rendered page by page as a report user interactively reads through a report. The amount of data on each page influences the rendering time for each page. The total number of pages is determined when a report is rendered. For some renderers, an estimated number of pages is displayed until all pages in a report have been rendered.

#### Images

Images are no longer retrieved during the initial session when a report is rendered. Images are retrieved when they are accessed for the first time during on-demand processing.

For history and execution snapshots, images are retrieved at snapshot creation time.

# **Execution Log: TimeDataRetrieval, TimeProcessing, TimeRendering**

Report log entries for **TimeDataRetrieval**, **TimeProcessing**, and **TimeRendering** are logged at the initial request to the report processor.

#### **Error Detection on Export**

In earlier versions of Reporting Services, the entire report was processed before any page could be viewed. Errors in expressions for the **Visibility.Hidden** RDL property were detected before a report could be exported. If you could view the first page of a report, you could export the entire report without error.

In this release, reports are processed page-by-page. If errors exist in an expression for the **Visibility.Hidden** RDL property, the error may not be detected until the page on which the error exists is rendered for export. In this case, the entire export fails. Being able to view a few pages of a report successfully does not guarantee that you can export the entire report. You must
try to export the report and wait for a successful completion before you know that the report exports without error.

Expression evaluation for group, sort, and filter operations continue to behave in the same way as previous Reporting Services versions. Errors in these expressions are detected by the report processing component, and are reported as critical errors before the first page of a report is rendered.

# <sup>□</sup> Report Rendering

Report rendering redesign has introduced the following behavior changes when rendering an existing report.

### **Page Breaks**

In earlier versions of Reporting Services, soft page break renderers handled report items in a container (in a rectangle or in the report body) in the following way: page breaks from the top-most and bottom-most report items were applied to the container in order to minimize extra blank pages. In the new rendering object model, page breaks that you set on report items, known as logical page breaks, always cause a new page to be rendered. No attempt is made to eliminate extra pages.

For more information, see Understanding Pagination in Reporting Services.

### **RepeatWith Items**

In earlier versions of Reporting Services, soft page break renderers included report items on a page when the **RepeatWith** property was set to **true**. These report items were not counted when calculating page size because of the flexible nature of page size for a soft page break renderer, nor were they counted when you set **InteractiveHeight** to control the amount of data on a page. In SQL Server 2008, these items will be counted toward the total page size. The result is that pages may contain less data, but setting the value for **InteractiveHeight** has more influence on the page size.

For more information, see **Understanding Rendering Behaviors**.

### **Nested Subreports and Data Regions in Excel**

In earlier versions of Reporting Services, nested data regions and subreports in table and matrix cells were not supported when you exported a report to Microsoft Office Excel. In SQL Server 2008, this limitation has been removed. You can design reports that use nested data regions and subreports in a data

region, export the report to the Excel renderer, and view the nested report items. For more information, see **Exporting to Microsoft Excel**.

### □ See Also

Concepts Backward Compatibility

**Other Resources** 

What's New (Reporting Services) Upgrade (Reporting Services) Reporting Services Backward Compatibility

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Integration Services Backward Compatibility

- -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> >

This section describes changes between versions of SQL Server Integration Services. It covers features that are no longer available or are scheduled to be removed in a future release. It also describes changes to the product that are known to break, or to change the behavior of, an existing application that includes Integration Services functionality.

# <sup>□</sup> In This Section

- Deprecated Integration Services Features in SQL Server 2008
- Discontinued Integration Services Functionality in SQL Server 2008
- Breaking Changes to Integration Services Features in SQL Server 2008
- Behavior Changes to Integration Services Features in SQL Server 2008
- Support for Data Transformation Services (DTS) in SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help Deprecated Integration Services Features in SQL Server 2008

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Integration Services Backward Compatibility</u> >

This topic describes the deprecated Integration Services features that are still available in SQL Server 2008. These features are scheduled to be removed in a future release of SQL Server. Deprecated features should not be used in new applications.

## Data Transformation Services (DTS)

Data Transformation Services (DTS) is deprecated. You should plan to remove dependencies on DTS in anticipation of future releases of SQL Server. DTS itself will continue to be supported in SQL Server 2000, SQL Server 2005, and SQL Server 2008 as part of our ongoing commitment to provide support for those versions of SQL Server. However, the ability to migrate or run DTS packages might not be supported in the next release.

The DTS functionality that will be deprecated includes the DTS runtime, the DTS API, the Package Migration Wizard for migrating DTS packages to the next version of Integration Services, support for DTS package maintenance in SQL Server Management Studio, and the Execute DTS 2000 Package task.

For information on how to migrate DTS packages to the Integration Services package format, see <u>Migrating Data Transformation Services Packages</u>.

# □ ActiveX Script Task

The ActiveX Script task in SQL Server 2008 is provided only for backward compatibility with DTS and will be removed in the next release 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 can re-create the functionality of your ActiveX scripts by using the new control flow containers and the more flexible precedence constraints available in Integration Services. Or, you can rewrite your scripts by using the Integration Services Script task.

For information on how to migrate scripts, see <u>Migrating Scripts</u>.

### $\Box$ See Also

### Concepts

Support for Data Transformation Services (DTS) in SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

Discontinued Integration Services Functionality in SQL Server 2008

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Integration Services Backward Compatibility</u> >

This topic describes Integration Services features that are no longer available in SQL Server 2008.

# <sup>□</sup> Microsoft Visual Studio for Applications (VSA)

In SQL Server 2008, Visual Studio Tools for Applications (VSTA) has replaced Visual Studio for Applications (VSA) as the scripting environment for the Script task and Script transformation. While this change does not require scripts to be rewritten, it does require that Script tasks and transformations that contain VSA scripts be upgraded when you migrate existing Integration Services packages to SQL Server 2008.

In SQL Server 2008, VSA scripts that are contained in Script tasks or transformations will be converted to VSTA before Integration Services runs the scripts.

### $\Box$ See Also

### Concepts

Deprecated Integration Services Features in SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Breaking Changes to Integration Services Features in SQL Server 2008

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Integration Services Backward Compatibility</u> >

This topic describes breaking changes in Integration Services. These changes might break applications, scripts, or functionalities that are based on earlier versions of SQL Server. You might encounter these issues when you upgrade. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.

# <sup>B</sup> Breaking Changes

The following table describes breaking changes between the version of Integration Services that is included with SQL Server 2008 and the version of Integration Services that is included in earlier releases.

Feature	Description
Provider names in connection strings	In SQL Server 2008, the names for the SQL Server Native Client provider and OLE DB Provider for Analysis Services have changed, and their connection strings use a different value. To automatically update connection strings that use these provider names, use the SSIS Package Upgrade Wizard to upgrade SQL Server 2005 packages. You can also update these connection strings automatically or manually in BI Development Studio.
Third-party and custom components	In SQL Server 2008, Integration Services has changed its application programming interfaces (APIs). To be compatible with these new APIs, all third-party and custom components must be recompiled.
© 2008 Mic	rosoft Corporation. All rights reserved.

# SQL Server Setup Help Behavior Changes to Integration Services Features in SQL Server 2008

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Integration Services Backward Compatibility</u> >

This topic describes behavior changes in Integration Services. Behavior changes affect how features work or interact in SQL Server 2008 as compared to earlier versions of SQL Server.

# <sup>□</sup> Behavior Changes

The following table describes changes in behavior between the version of Integration Services that is included in SQL Server 2008 and the version of Integration Services that is included in earler releases.

	SQL Server 2005	SQL Server 2008
Issue	Integration Services	Integration Services
	Behavior	Behavior
Returning a NULL value in a package variable from the Execute SQL task	Returns the value as 11/30/1999.	Returns the value as 12/30/1899.
Developing scripts in the Script task and Script component	Uses Visual Studio for Applications (VSA) as the scripting environment.	Uses Visual Studio Tools for Applications (VSTA) as the scripting environment. To migrate packages that contain scripts, use the SSIS Package Upgrade Wizard.
Using the Lookup transformation	Cannot persist the cached reference table or share this table among multiple Lookup transformations.	Can persist the cached reference table or share this table among multiple Lookup transformations. To use the new features of the Lookup transformation, consider revising your packages.

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

Support for Data Transformation Services (DTS) in SQL Server 2008



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Integration Services Backward Compatibility</u> >

SQL Server 2008 provides optional 32-bit management, run-time, and designtime support for SQL Server 2000 Data Transformation Services (DTS) packages that were created by using the DTS tools and object model.

### <sup>Important:</sup>

The features descibed in this topic, including run-time support for DTS packages, require the manual installation of additional, optional components that are not available during setup. For more information, see the sections, "Installing run-time support for DTS packages " and "Installing design-time support for DTS packages," later in this topic.

Support for DTS in SQL Server 2008 includes the following features, which are described in more detail later in this topic:

- The DTS runtime, the object model that it exposes, and the dtsrun.exe command prompt utility.
- The Execute DTS 2000 Package task, for executing DTS packages within Integration Services packages.
- The ActiveX Script task, for backward compatibility only.
- The DTS Package Migration Wizard, for migrating DTS packages to the Integration Services package format.
- The Upgrade Advisor rules for DTS packages, for identifying potential issues that may be encountered when migrating packages.

SQL Server 2008 does not include support for the following DTS features:

• There is no 64-bit design-time or run-time support for DTS packages. On

a 64-bit computer, DTS packages, and Integration Services packages that run DTS packages, can run only in 32-bit mode. To run DTS packages in 32-bit mode, you have to install and use the 32-bit version of the **dtexec** utility (dtexec.exe). To install the 32-bit version of the **dtexec** utility, select Client Tools or Business Intelligence Development Studio during setup.

• There is also no 32-bit design-time or run-time support for DTS packages on Itanium-based operating systems. Therefore, you cannot create, view, modify, or run DTS packages on Itanium-based operating systems.

# <sup>□</sup> Sections in This Topic

The following links connect you to the various sections in this topic:

#### **General information**

- General compatibility between DTS and SSIS
- Managing DTS packages

#### **Designing and running DTS packages**

- Installing run-time support for DTS packages
- <u>Running DTS packages at the command prompt</u>
- Running DTS packages from SQL Server Agent jobs
- <u>Running DTS packages from Integration Services packages</u>
- Installing design-time support for DTS packages

#### **Migrating DTS packages**

- <u>Migrating DTS packages</u>
- <u>Migrating dtsrun commands to dtexec commands</u>

#### **Limited support**

- Support for ODBC data sources
- <u>Support for Repository packages</u>
- <u>Support for Integration Services in SQL Server Express</u>

# □ General Compatibility between DTS and Integration Services

DTS and Integration Services are completely different products. Integration Services is not a version upgrade from DTS. Therefore, there are no software conflicts between DTS and Integration Services. You can do the following procedures:

- Install both DTS and Integration Services on the same server.
- Run both DTS and Integration Services packages on the same server. Even if the SQL Server 2000 tools are not present on the server, you can run DTS packages by using the updated version of the DTS runtime that is installed with Integration Services.
- Save DTS packages in the **msdb** database of an instance of SQL Server 2000, or an instance that is running SQL Server 2005 or a later version. However, you can save SQL Server 2008 Integration Services packages only on an instance that is running SQL Server 2008.
- Use an optional download of the DTS Designer to modify DTS packages even if the SQL Server 2000 tools are not present on the computer. However, you cannot open or modify DTS packages directly in Business Intelligence Development Studio.

<sup>I</sup> Note:

For information about how to download the updated version of the DTS Designer, see the section, <u>Installing Design-time Support for SQL Server</u> 2000 DTS Packages.

When you upgrade an existing instance of SQL Server to SQL Server 2008, the system tables (for example, **sysdtspackages**) that DTS uses in the **msdb** database remain intact. DTS packages are not automatically migrated. For information about how to migrate packages, see the section, <u>Migrating DTS packages</u>.

# <sup>□</sup> Managing DTS Packages

After you install the optional run-time support for DTS as described in the section, "<u>Installing run-time support for DTS packages</u>," you can manage DTS packages from SQL Server Management Studio. These DTS packages can be stored in an instance of SQL Server 2000, or imported to and stored in an instance that is running SQL Server 2005 or a later version.

### <sup>III</sup> Important:

For important information about how to secure DTS packages, see **Securing DTS Packages Stored in SQL Server**.

To view DTS packages in Management Studio, first connect to an instance of SQL Server. (Do not connect to an instance of Integration Services.) Then, in the Object Browser under the **Management\Legacy\Data Transformation Services** node, you will be able to perform the following tasks:

- When you select the **Data Transformation Services** node in Management Studio, you can view and open packages, import them to SQL Server from structured storage files, or start the DTS Package Migration Wizard.
- When you select an individual DTS package in Management Studio, you can open the package to modify it, migrate it, export it to a structured storage file, or delete it.

#### <sup>I</sup>Note:

To open or modify DTS packages, a separate download of the DTS designer components is required. See the section, <u>Installing Design-time</u> <u>Support for SQL Server 2000 DTS Packages</u>. DTS packages cannot be opened or modified directly in BI Development Studio.

### <sup>□</sup> Installing Run-time Support for DTS Packages

Although you can run DTS packages and Integration Services packages on the same computer, SQL Server 2008 does not install run-time support for DTS packages. You have to install this run-time support before you can run and manage DTS packages. For information about how to install run-time support for DTS, see <u>How to: Install Support for Data Transformation Services</u> <u>Packages</u>.

The backward compatibility files that provide run-time support for DTS packages also include run-time support for the SQL Server 2000 Analysis Services Processing task and its dependency, Decision Support Objects (DSO). However, the backward compatibility files do not include design-time support for modifying DTS packages that contain the SQL Server 2000 Analysis Services Processing task. You can modify packages that contain this task only when you have SQL Server 2000 Analysis Services installed on the same computer.

The backward compatibility files that provide run-time support for DTS packages do not include support for the SQL Server 2000 Data Mining Prediction Query task.

# <sup>□</sup> Running DTS Packages at the Command Prompt

You can run DTS packages at the command prompt. Integration Services installs the **dtsrun.exe** utility when it installs DTS run-time support.

### <sup>©</sup> Running DTS Packages from SQL Server Agent Jobs

To run DTS packages from a SQL Server Agent job, you must set the job step to **Operating system (CmdExec)** and use the **dtsrun** utility (dtsrun.exe) to run the package. You cannot set the job step to **SQL Server Integration Services Package**. For more information, see **How to: Run a Package**.

# Running DTS Packages from Integration Services Packages

You can also run DTS packages as part of an Integration Services package. Integration Services provides the Execute DTS 2000 Package task for running DTS packages within an Integration Services package. For more information, see **Execute DTS 2000 Package Task**.

### <sup>□</sup> Installing Design-time Support for DTS Packages

DTS packages cannot be opened or modified directly in Business Intelligence Development Studio. In addition, neither SQL Server 2008, nor SQL Server 2005, installs the DTS package designer for modifying DTS packages. However, you can download and install the DTS package designer, and use it to modify DTS packages. For information about how to install design-time support for DTS packages, see <u>How to: Install Support for Data Transformation Services</u> <u>Packages</u>. After installing this download, you can view and modify DTS packages from Management Studio, or from the Execute DTS 2000 Package Task Editor in Business Intelligence Development Studio.

For information about how to manage certain DTS application properties that affect the DTS design environment, see **Setting DTS Application Properties**.

# Migrating DTS Packages

SQL Server provides the DTS Package Migration Wizard for migrating DTS packages to the Integration Services package format. When you select Integration Services for installation, Setup also installs the DTS Package Migration Wizard. You can optionally use the DTS Package Migration Wizard to upgrade 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.

### <sup>I</sup>∕ Note:

The DTS Package Migration Wizard is available in the SQL Server Standard, SQL Server Enterprise, and SQL Server Developer.

Many packages can be fully migrated to Integration Services, especially DTS packages that use tasks, connections, and other objects that have an equivalent in Integration Services. Other packages can be migrated successfully by encapsulating certain tasks or features that cannot be migrated in an Execute DTS 2000 Package task. For more information, see <u>Migrating Data</u> <u>Transformation Services Packages</u>.

Before running the DTS Package Migration Wizard, you might want to install and run Upgrade Advisor to analyze your existing packages for migration issues. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.

### <sup>I</sup>Note:

Upgrade Advisor cannot analyze, and the DTS 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 DTS Package Migration Wizard can migrate, SQL Server 7.0 DTS packages that are saved as structured storage files. Also, the Upgrade Advisor can analyze, and the DTS Package Migration Wizard can migrate, SQL Server 7.0 DTS packages that are saved in the **msdb** database of an instance of SQL Server 7.0 DTS packages that are saved in the **msdb** database of an instance of SQL Server 7.0 DTS packages that are saved in the **msdb** database of an instance of SQL Server 7.0 that has been upgraded to SQL Server 2005 or a later version.

# <sup>B</sup> Migrating Command Lines for Executing Packages

To understand how the command-line options for the SQL Server 2000 DTS **dtsrun** utility (dtsrun.exe) map to the options for the Integration Services **dtexec** utility (dtexec.exe), see **Mapping dtsrun Options to dtexec Options**.

# <sup>□</sup> Support for ODBC Data Sources

To load or save data to an ODBC data source, you can use an ADO.NET source or destination and select the .NET Framework Data Provider for ODBC. It is also possible to use the Script component to create an ad hoc ODBC source or destination for use in a single package. For more information, see **Creating an ODBC Destination with the Script Component**.

# Support for Meta Data Services (Repository) Packages

SQL Server 2000 Meta Data Services, typically known as the Repository, is a deprecated component. SQL Server 2005 and later versions do not install or use the Repository.

The DTS designer and the **dtsrun** utility continue to support DTS packages that were saved to the Repository.

Integration Services supports the Repository only in the Upgrade Advisor and the DTS Package Migration Wizard, and only if the Repository redistributable files are installed on the local computer. (These files are normally present when the computer has either SQL Server 2000 or the SQL Server 2000 tools installed.) When the Repository redistributable files are present, the Upgrade Advisor can scan, and the DTS Package Migration Wizard can migrate, DTS packages that were saved to the Repository. When the Repository redistributable files are not present, the Upgrade Advisor and the DTS Package Migration Wizard can only be used on DTS packages that were saved to SQL Server or to structured storage files.

The Integration Services Execute DTS 2000 Package task cannot run a DTS package that was saved to the Repository. However, the SQL Server 2000 DTS runtime can run such a package. Thus, to have an Integration Services package run a DTS package that is stored in the Repository, follow these steps:

- 1. Make sure that the Repository files are installed on the computer that is running the Integration Services package. To install the Repository files, install SQL Server 2000, the SQL Server 2000 tools, or the Repository redistributable files on the local computer
- 2. Create a new DTS package to act as a wrapper for the DTS package that is stored in the Repository. In this DTS wrapper package, use an Execute Package task to execute the package that is stored in the Repository. Save this wrapper package to SQL Server or as a structured storage file.
- 3. Use an Execute DTS 2000 Package task in the Integration Services package to run the new DTS wrapper package.

### <sup>©</sup> Support for DTS Packages in SQL Server Express

SQL Server Express does not include Integration Services nor does it support DTS packages. However, you can work with DTS packages on a server that is running SQL Server Express by using one of the following methods:

- To run existing DTS packages on a server that is running SQL Server Express, you need the DTS runtime on the server. Although SQL Server Express does not include the DTS runtime, you can install the DTS runtime on the server by either installing the SQL Server 2000 client tools or the DTS redistributable files.
- To modify existing DTS packages on a server that is running SQL Server Express, you can use the SQL Server 2000 tools. You can also modify the packages remotely in SQL Server Management Studio on a server that is running one of these versions of SQL Server: Workgroup, Standard, Enterprise, or Developer. SQL Server Express does not include SQL Server Management Studio.
- To migrate existing DTS packages to Integration Services, you need the DTS Package Migration Wizard that is installed with Integration Services. SQL Server Express does not include this wizard. To run the wizard, you can use one of these versions of SQL Server: Standard, Enterprise, or Developer.

### <sup>□</sup> See Also

### Concepts

Known DTS Package Migration Issues

**Other Resources** 

Mapping dtsrun Options to dtexec Options Setting DTS Application Properties 64-bit Considerations for Integration Services

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Replication Backward Compatibility



#### -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> >

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.

Торіс	Description
Deprecated Features in SQL Server 2008 Replication	Replication features that have been retained in Microsoft SQL Server 2008 for backward compatibility, but which will be removed in a future version of SQL Server.
<u>Discontinued</u> Functionality in SQL <u>Server 2008</u> <u>Replication</u>	Replication features that existed in earlier versions but were removed in SQL Server 2005. No features have been removed in SQL Server 2008.
<u>Behavior Changes in</u> SQL Server 2008 Replication	Replication features that have changed in SQL Server 2008.
Breaking Changes in SQL Server 2008 Replication	Replication feature changes that might require changes to applications.
<u>Using Multiple</u> <u>Versions of SQL Server</u>	Considerations for using more than one version of

2008 in a Replication	SQL Server in a replication topology.
Topology	
# <sup>□</sup> See Also

# Concepts

Considerations for Upgrading Replicated Databases

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help Deprecated Features in SQL Server 2008 Replication



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Replication Backward Compatibility</u> >

This topic describes the deprecated Replication features that are still available in SQL Server 2008. These features are scheduled to be removed in a future release of SQL Server. Deprecated features should not be used in new applications.

# □ Deprecated Features That Affect All Types of Replication

The following features, which apply to all types of replication, are deprecated in SQL Server 2005 or SQL Server 2008.

Feature	Description	
	This feature can be used when deploying a large number of pull subscriptions, which is common in merge replication. We recommend that you use the following approaches rather than attachable subscriptions:	
Attachable subscriptions	<ul> <li>For merge publications that are partitioned using parameterized filters, we recommend that you use partitioned snapshots, which simplify the initialization of a large number of subscriptions. For more information, see Snapshots for Merge Publications with Parameterized Filters.</li> <li>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.</li> </ul>	
	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 <b>Scripting Replication</b> .	
	The <b>sp_addsubscriber</b> stored procedure is deprecated.	
Subscriber registration	It is no longer required to explicitly register a Subscriber at the Publisher.	
	Existing code will continue to work, but SQL-DMO	

SQL Distributed	does not support new features in SQL Server 2005 and
Management Objects	SQL Server 2008. Use Replication Management
(SQL-DMO)	Objects (RMO) instead. For more information, see
	Replication Management Objects Concepts.
	The stored procedures <b>sp_repladdcolumn</b> and
	<b>sp_repldropcolumn</b> have been deprecated. Use
	schema change replication instead. For more
	information, see Making Schema Changes on
Schema changes using	Publication Databases.
<b>sp_repladdcolumn</b> and	The stored procedures cannot be used for adding or
sp_repldropcolumn	dropping columns with data types introduced in SOL
	Server 2005 and later versions: <b>xml</b> . <b>varchar(max)</b> .
	<b>nvarchar(max)</b> , <b>varbinary(max)</b> , or user-defined
	types (UDT), <b>datetime2</b> , <b>time</b> , <b>datetimeoffset</b> .
	<b>hierarchvid</b> , <b>geometry</b> , and <b>geography</b> types.
	Checksum validation should not be used. Use binary
	checksum validation. You can also use row count
	validation for all SQL Server Subscribers, including
Checksum validation	those subscribing to publications from Oracle
	Publishers. For more information, see <b>Validating</b>
	Replicated Data.
	Adding a publication to Active Directory using the
Adding publications to	<pre>@add_to_active_directory parameter of</pre>
Adding publications to	<b>sp_addpublication</b> or <b>sp_addmergepublication</b> , has
Active Directory	been deprecated. Subscribing to a publication by
	locating it in Active Directory has been discontinued.
	This parameter of the Distribution Agent and Merge
	Agent is deprecated because it is not compatible with
-UseInprocLoader	the XML data type. If you are not replicating XML
parameter	data, this parameter can be used. For more information,
	see <b>Replication Distribution Agent</b> and <b>Replication</b>
	Merge Agent.
PublisherAddress ,	
PublisherNetwork,	These parameters are used to specify the IP address
DistributorNetwork,	when connecting to the publisher or distributor.
	Because the replication subsystem uses server names to
DISTRIDUTORAGARESS	verify the connection, we recommend that you use allas

parameters in	at the client protocols to map the IP address to server
Distribution and Merge	name and use the server name in the agents.
Agents <sup>1</sup>	

<sup>1</sup> Deprecated in SQL Server 2008.

# <sup>□</sup> Deprecated Features of Transactional Replication

The following transactional replication features were deprecated in SQL Server 2005 or SQL Server 2008.

Feature	Description
	The <i>@retention</i> property of <i>sp_addpublication</i> has been
Subscription	deprecated. Subscriptions are still marked as inactive and must be
expiration for	reinitialized if they have not synchronized within the maximum
transactional	distribution retention period (the @max_distretention property
publications	of <b>sp_adddistributiondb</b> . For more information about retention
	periods, see Subscription Expiration and Deactivation.
	A subscription is a "no sync" subscription if a value of <b>none</b> is
"No sync"	specified for the <b>@sync_type</b> parameter of <b>sp_addsubscription</b>
subscriptions	or <b>sp_addpullsubscription</b> . If you want to specify that the
to	necessary schema and data are already present in the subscription
transactional	database, specify a value of <b>replication support only</b> for the
publications	parameter instead. For more information, see <b>Initializing a</b>
	Transactional Subscription Without a Snapshot.
ODBC	Use OLE-DB for non-SQL Server Subscribers instead. For
Subscribers	information about supported Subscribers, see <b>Non-SQL Server</b>
Outsensens	Subscribers.
	This feature is available through the stored procedure interface,
Transformable	but support for this feature in the user interface has been dropped.
subscriptions	Using the feature requires installation of SQL Server 2000 Data
1	Transformation Services (DTS). For more information, see
	Integration Services Backward Compatibility.
Updatable	
subscriptions	
including	
immediate	We recommend that you use peer-to-peer transactional
updating and	replication instead. For more information, see <b>Peer-to-Peer</b>
queued	Transactional Replication.
updating with	
snapshot and	
transactional	

publications <sup>1</sup>	
Distribution ActiveX control	This control allows you to embed the Distribution Agent in applications. Use RMO instead. For more information, see <b>How</b> <b>to: Synchronize a Pull Subscription (RMO Programming)</b> and <b>How to: Synchronize a Push Subscription (RMO</b> <b>Programming)</b> .
Replication Distributor Interface	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. SQL Server 2005 and SQL Server 2008 support publishing from Oracle databases without custom programming. For more information, see <b>Oracle Publishing Overview</b> .
Replicating to Oracle 8 subscribers and from Oracle 8 publishers1	For information about supported versions of Oracle, see <b>Oracle</b> <b>Publishing Overview</b> and <b>Oracle Subscribers</b> .

<sup>1</sup> Deprecated in SQL Server 2008.

# <sup>□</sup> Deprecated Features of Merge Replication

The following merge replication features were deprecated in SQL Server 2005 or SQL Server 2008.

Feature	Description
	The alternate synchronization partners feature allows you to specify an alternate Publisher with which a Subscriber can synchronize.
Alternate synchronization partners	In SQL Server 2005 and SQL Server 2008, we recommend that you use merge replication in conjunction with database mirroring, rather than alternate synchronization partners. For more information, see <b>Replication and</b> <b>Database Mirroring</b> .
"No sync" subscriptions to merge publications <sup>1</sup>	A subscription is a "no sync" subscription if a value of <b>none</b> is specified for the <b>@sync_type</b> parameter of <b>sp_addmergesubscription</b> or <b>sp_addmergepullsubscription</b> . This type of subscription is not recommended for merge replication.
Merge ActiveX control	This control allows you to embed the Merge Agent in applications. Use RMO instead. For more information, see <b>How to: Synchronize a</b> <b>Pull Subscription (RMO Programming)</b> or <b>How to: Synchronize a Push Subscription</b> <b>(RMO Programming)</b> .
	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.

Multicolumn UPDATE option	In versions of SQL Server prior to SQL Server 2005, we recommend that you specify a value of <b>false</b> for the <b>@fast_multicol_updateproc</b> article option to address cases in which a multicolumn update (one UPDATE statement) might be less efficient:
	<ul> <li>Most updates involve a small number of columns.</li> <li>Index maintenance on unchanged columns is high because those columns are reset when updates occur.</li> </ul>
	Due to performance improvements in SQL Server, this option is no longer required for these cases.
<b>-ParallelUploadDownload</b> parameter <sup>1</sup>	This parameter of the Merge Agent is used to perform simultaneous upload and download of changes in a merge replication session. This parameter provides a performance gain, but it is outweighed by the amount of metadata that must be transferred over the network.
allow partition realignment	This parameter is used to control the delete
property in	operations that must be sent to Subscribers if a
sn addmarganublication <sup>1</sup>	row moves out of the Subscriber's partition
-ExchangeType parameter <sup>1</sup>	This parameter is used to control whether the Merge Agent goes through the upload phase or the download phase or both. This defaults to 3 to perform both upload and download. We do not recommend upload-only because it would not replicate schema changes or initialization processes. Download-only functionality can be achieved by using <b>@subscriber_upload_options</b> for an article. For more information, see <b>sp_addmergearticle (Transact-SQL)</b> .
	This property is used to stop tracking deletes when deletes should be sent down to the Publisher or Subscriber. This can be

@delete_tracking property in sp_addmergearticle <sup>1</sup>	implemented by using the <b>DeleteHandler</b> in the <b>BusinessLogicModule</b> . For more information, see <b>Executing Business Logic</b> <b>During Merge Synchronization</b> .
Logical Records <sup>1</sup>	This feature is used to send a set of related rows in a single transaction. In most cases, this feature adds significant performance overhead to replication when it is used. For more information, see <b>Grouping Changes to</b> <b>Related Rows with Logical Records</b> .

<sup>1</sup> Deprecated in SQL Server 2008.

# $\Box$ See Also

# Concepts

**Replication Backward Compatibility** 

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help Discontinued Functionality in SQL Server 2008 Replication

# 

#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Replication Backward Compatibility</u> >

This topic describes Replication features that are no longer available in SQL Server 2008.

# Discontinued Functionality in SQL Server Replication

This section describes replication features that were discontinued in SQL Server 2005.

#### **Discontinued Features That Affect All Types of Replication**

The following replication features have been discontinued in SQL Server.

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 <b>How to: Create a Push Subscription (SQL Server</b> <b>Management Studio)</b> .
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 <b>Transferring</b> <b>Snapshots Through FTP</b> .
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 <b>Windows Synchronization Manager</b> .
Subscribing to a publication by locating it in Active	Adding a publication to Active Directory using the <b>@add_to_active_directory</b> parameter of <b>sp_addpublication</b> or <b>sp_addmergepublication</b> has been deprecated. Subscribing to a publication by locating it in Active Directory has been

Directory	discontinued.	
	This control allows you to embed the Snapshot Agent in	
Snapshot	applications. Use the new managed Snapshot Agent instead.	
ActiveX control	For more information, see <b>SnapshotGenerationAgent</b> and	
	How to: Create the Initial Snapshot (RMO Programming).	
	SQL Server 2000 supported running the Distribution Agent or	
Remote agent activation	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.	
	Jet is the underlying database used by Access, and replication	
	supported subscriptions to Jet databases in SQL Server 2000.	
Microsoft	These subscriptions are no longer supported.	
Access (Jet 4.0)		
Subscribers	We recommend using SQL Server Express instead. Access can	
	use a SQL Server database as a backend, and SQL Server	
	databases are not affected by this issue.	

# **Discontinued Transactional Replication Features**

The following transactional replication features have been discontinued.

subscriptions,	see Updatable Subscriptions for Transactional
Replication.	

# **Discontinued Merge Replication Features**

The following merge replication features have been discontinued.

Feature	Description
	SQL Server MSDE could serve as a Publisher for merge
	publications. SQL Server Express, cannot serve as a
Dubliching from SOI	Publisher. It can subscribe to merge, transactional, and
Publishing Holli SQL	snapshot publications. Merge replication, and
Server Express	transactional replication with updating subscriptions, both
	allow changes to be propagated from Subscribers back to
	the Publisher.
	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
IVBC ustom Basolvar	COM-based resolvers. For more information, see
interface	Executing Business Logic During Merge
interface	Synchronization.
	You can still write COM-based resolvers in C++. For
	more information, see <b>COM-Based Custom Resolvers</b> .

# $\Box$ See Also

# Concepts

**Replication Backward Compatibility** 

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help Breaking Changes in SQL Server 2008 Replication



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Replication Backward Compatibility</u> >

This topic describes breaking changes in SQL Server Replication. These changes might break applications, scripts, or functionalities that are based on earlier versions of SQL Server. You might encounter these issues when you upgrade. For more information, see <u>Using Upgrade Advisor to Prepare for Upgrades</u>.

# <sup>B</sup> Breaking Changes Made in SQL Server 2005

This section describes breaking changes in replication features that were made in SQL Server 2005.

#### **Breaking Changes That Affect All Types of Replication**

The following breaking changes apply to all types of replication.

Feature	Description
	The replication agent security model has changed from SQL Server 2000, For detailed information about the security
	model, see <b>Replication Agent Security Model</b> . If you are a
	member of the <b>sysadmin</b> fixed server role in SQL Server
	2005 and you run replication scripts created from SQL
Changes required	Server 2000 or SQL Server 7.0, the scripts execute properly.
for replication	If you are a member of the <b>dbo</b> fixed database role or
scripts	another role, the scripts fail and must be upgraded. For
Scripts	information about upgrading scripts, see <b>How to: Upgrade</b>
	Replication Scripts (Replication Transact-SQL
	<b>Programming)</b> . Although it is not required to upgrade
	scripts that are executed by members of the <b>sysadmin</b> role,
	it is recommended in order to take advantage of security
	enhancements.
	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 for replication agents	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 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 Considerations for Upgrading Replicated
	Databases and Replication Agent Security Model.
	All ActiveX controls are marked as unsafe for scripting and initialization.
ActiveX controls	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).
	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 <b>distributor_admin</b> account, see <b>Securing the Distributor</b> .
	To determine the type of connection being used
Password for the	• Execute <b>sp_helpdistpublisher</b> at the Distributor. If the value in the <b>trusted</b> column is <b>1</b> , you must change to a non-trusted connection.
<b>distributor_admin</b> account	To change to a non-trusted connection
	<ol> <li>Execute sp_changedistpublisher at the Distributor, specifying a value of 'trusted' for the parameter @property and a value of 'False' for the parameter @value.</li> <li>Mote:</li> </ol>

	Some versions of the SQL Server 2000 Books Online do not list 'trusted' as a valid value for <b>@property</b> . It is valid for all SQL Server 2000 releases.
	2. Execute <b>sp_changedistributor_password</b> at both the Publisher and the Distributor, specifying a strong password for the parameter <b>@password</b> .
	If you are upgrading to SQL Server Express, you must reconfigure replication synchronization because SQL Server Express does not include SQL Server Agent.
SOL Server	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
Express does not	information, see <b>Replicating Data to SQL Server Express</b> .
include SQL	If you want to continue to use SOL Server Agent to run
Server Agent	replication agent jobs, you must use push subscriptions or
	upgrade to a different version of SQL Server (all versions
	except SQL Server Express and SQL Server Compact 3.5
	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).
	replication supported subscriptions to let databases in SOI
	Server 2000. These subscriptions are no longer supported.
Microsoft Access	
(Jet 4.0)	It is recommended to use SQL Server Express instead.
Subscribers	Access can use a SQL Server database as a backend, and
	SQL Server databases are not affected by this issue. For
	more information, see <b>Replicating Data to SQL Server</b>
	Express.

# **Breaking Changes for Transactional Replication**

The following breaking changes apply to transactional replication.

Feature	Description
Initializing a transactional subscription from a	To initialize a subscription from a backup in SQL Server 2008, a user must be a member of the <b>dbcreator</b> server role. In SQL Server 2005, membership in the <b>db_owner</b> database role was sufficient.
backup <sup>1</sup>	For more information about how to initialize a subscription from a backup, see <b>Initializing a Transactional Subscription Without a Snapshot</b> .
Message Queuing option for queued updating subscription	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 Message Queuing to queue changes. The type of queue was specified with the @queue_type parameter of sp_addpublication, which allowed values of sql and msmq (Message Queuing). In SQL Server 2005, only a value of sql is allowed. Existing publications that use Message Queuing are modified during upgrade to use a SQL Server queue. If you have applications that depend on queued updating using Message Queuing, 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. Upgrade will remove the existing Message Queuing subscription queues if the Message Queuing service is running while SQL Server is being upgraded.  Important: In Windows 2000 and Windows XP, the Microsoft Distributed Transaction Coordinator (MSDTC) service must also be running, because Message Queuing service is not running, remove the queues manually after upgrade is complete. For more information about

	removing queues, see the Windows documentation.
Change to call format for updating subscriptions	By default, a set of stored procedures is used to apply changes to Subscribers in transactional replication. Each procedure has a <i>call</i> <i>format</i> . This format determines how parameters are passed to the procedure and the amount of data that is sent to the Subscriber. In SQL Server 2000, the default format is CALL. In SQL Server 2005 and SQL Server 2008, the default format is VCALL. This change only affects topologies in which the stored procedures have been customized. After upgrade, you must change the signature of the customized procedure to include additional parameters. Otherwise, the Distribution Agent will fail.

<sup>1</sup> This issue affects only SQL Server 2008 and later versions.

#### **Breaking Changes for Merge Replication**

The following breaking changes apply to merge replication.

Feature	Description
	SQL Server MSDE could serve as a Publisher for merge publications.
	SQL Server Express, the replacement for MSDE, cannot server as a
Publishing	Publisher. It can subscribe to merge, transactional, and snapshot
from SQL	publications. Merge replication and transactional replication with
Server	updating subscriptions both allow changes to be propagated from
Express	Subscribers back to the Publisher. For more information about
	replicating to SQL Server Express, see <b>Replicating Data to SQL</b>
	Server Express.
	In previous versions of SQL Server, changes made by the Merge
	Agent were performed on a row-by-row basis. In SQL Server 2005,
Batching	changes are batched to improve performance; therefore, more than
of	one row can be inserted, updated, or deleted within a single
changes	statement. If any published tables in the publication or subscription
Changes	databases have triggers, ensure that the triggers can handle multi-row
	inserts, updates, and deletes. For more information, see <b>Multirow</b>
	Considerations for DML Triggers.
	On upgrade to SQL Server 2005, conflict tables are re-created with

D	DBO as their owner. If any of the tables were owned by other users
Re-	in SQL Server 2000, your application might need to be modified.
creation	
of conflict	Merge replication creates a conflict table for each article in a
tables	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.
	For tables that use automatic identity range management, replication
	might assign new identity ranges during upgrade. If any tables have a
Now	larger identity range assigned to the Subscriber than to the Publisher,
idontity	replication assigns a range to the Publisher equal to that of the
ranges	Subscriber.
assigned	To determine ranges being used for each article, execute
	<b>sp_helpmergearticle</b> in the publication database and check the
	<pre>pub_identity_range and identity_range columns.</pre>

# $\Box$ See Also

# Concepts

**Replication Backward Compatibility** 

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help Behavior Changes in SQL Server 2008 Replication



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Replication Backward Compatibility</u> >

This topic describes behavior changes in SQL Server Replication. Behavior changes affect how features work or interact in SQL Server 2008 as compared to earlier versions of SQL Server.

# <sup>□</sup> Behavior Changes Made in SQL Server 2005

This section describes behavior changes made to replication features in SQL Server 2005.

#### **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. SQL Server now 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 <b>Security and Protection</b> <b>(Replication)</b> and <b>Replication Agent Security Model</b> . For information about how this change affects upgrade, see the "New Replication Agent Security Model" section of <u>Considerations for Upgrading Replicated Databases</u> and <u>Breaking Changes in SQL Server 2008 Replication</u> .
Windows Synchronization Manager	In versions of SQL Server prior to SQL Server 2005, 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 <b>How to: Synchronize a</b> <b>Subscription Using Windows Synchronization Manager</b> (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 <u>Using Upgrade Advisor to Prepare for Upgrades</u> .
	Schema options allow you to specify how attributes and objects
Schoma option	associated with tables, such as indexes and constraints, are
schemaes	replicated. The behavior of a number of schema options has
changes	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.

<sup>ℤ</sup>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 <b>0x8000</b> is also enabled, the primary key is created as a constraint with an index. If option <b>0x8000</b> 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 <b>0x10</b> and <b>0x40</b> are not enabled (these options control index creation for other cases).
0x4000	Creates a constraint or index. If option <b>0x8000</b> is also enabled, the unique constraint is created as a constraint with an index. If option <b>0x8000</b> 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 <b>0x10</b> and <b>0x40</b> are not enabled (these options control index creation for other cases).
0x8000	Creates primary key constraints and unique constraints if options <b>0x80</b> or <b>0x4000</b> are also specified. If neither of these options is specified,	The option has no effect.

option <b>Ux8000</b> has no effect.
-------------------------------------

#### **Behavior Changes for Transactional Replication**

The following changes affect transactional replication.

Feature	Description
Subscriber object ownership	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 provious 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 <b>sp_addarticle (Transact-SQL)</b> .
	The <b>@security_mode</b> parameter of <b>sp_link_publication</b> governs how the triggers for immediate updating subscriptions execute calls at the Publisher. In SQL Server 2005, the options for this parameter are:
Updatable subscriptions security mode	<ul> <li>0: Uses SQL Server Authentication with the login specified in the stored procedure as login and password.</li> <li>1: Uses the security context (SQL Server Authentication or Windows Integrated Authentication) of the user making the change at the Subscriber.</li> <li>2: Uses an existing, user-defined linked server login.</li> </ul>
	In previous versions of SQL Server, option <b>0</b> was used to specify a dynamic remote procedure call (RPC) from the Subscriber to the Publisher rather than a linked server.

#### **Behavior Changes for Merge Replication**

The following changes affect merge replication.

Feature	Description
	In previous versions of SQL Server, the compatibility

Publication compatibility level	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 f Merge Publications" section of <u>Using Multiple Versic</u> <u>SQL Server 2008 in a Replication Topology</u> .
Compensating actions	In previous versions of SQL Server, compensating ac were taken if errors (such as constraint violations) we encountered during synchronization. In some cases th 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. In SQL Server 2005, the <b>@compensate_for_errors</b> parameter of <b>sp_addmergearticle</b> controls whether compensating actions are taken. When set to <b>False</b> (t default), compensating actions are disabled; however errors are still logged and subsequent merges will con 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 appl and data will converge. When set to <b>True</b> , a change t cannot be applied at a node during synchronization le compensating actions that undo the change at all othe nodes.
	✓ Note: If the source table for an article is already published another publication, the value of @compensate_for_errors must be the same for bo articles. For pull subscriptions at Subscribers runnin 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 F
	In previous versions of SQL Server, merge replication

Conflict tables	created a single conflict table for each table article in publication, with a name in the form <b>conflict</b> _< <i>ArticleName</i> >. In SQL Server 2005, the information is contained in two tables: <b>MSmerge_conflicts_info</b> and a table with a name in
	form <b>MSmerge conflict</b> <publicationname> <articlen< td=""></articlen<></publicationname>
Retention-based metadata cleanup	<ul> <li>SQL Server 2005 uses retention-based metadata clear which was introduced in SQL Server 2000 Service Pa Metadata is periodically deleted from the following s tables:</li> <li>MSmerge_contents</li> <li>MSmerge_tombstone</li> <li>MSmerge_genhistory</li> <li>Any before image tables, if present. Before imag tables are present if the @keep_partition_chan synchronization optimization option is enabled ( publication (see the next section for more inforn about this option).</li> </ul>
The	The <b>@keep_partition_changes</b> parameter was set to by default in previous version of SQL Server, becaus results in more data being stored at the Publisher. It is
@keep_partition_changes	set to <b>True</b> if the publication compatibility level is 90
parameter	or higher, and the <b>@use_partition_groups</b> paramete
	to <b>False</b> . For more information about these options, s
	Parameterized Row Filters.

# $\Box$ See Also

# Concepts

**Replication Backward Compatibility** 

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

Using Multiple Versions of SQL Server 2008 in a Replication Topology



#### - -

<u>Upgrading to SQL Server 2008</u> > <u>Backward Compatibility</u> > <u>Replication Backward Compatibility</u> >

Replication supports replicating data to different versions of SQL Server. This topic provides information about the following areas:

- SQL Server versions supported
- Mapping SQL Server 2008 data types for earlier versions
- Restoring a replicated database from an earlier version
- Compatibility level for merge publications

For information about how to replicate data to SQL Server Express and SQL Server Compact 3.5, see **Replicating Data to SQL Server Express** and **Replicating Data to SQL Server Compact**. For information about the features that are supported by each edition of SQL Server, see <u>Features Supported by the Editions of SQL Server 2008</u>.

# Supported Versions of SQL Server

SQL Server 2000 and SQL Server 2005 can both participate in replication topologies with SQL Server 2008. For SQL Server 2000 the minimum version is Service Pack 3 (SP3). For SQL Server 2005 the minimum version is Service Pack 2 (SP2).

When you replicate between or among different versions of SQL Server, you are usually limited to the functionality of the earliest version used. For example, if you upgrade a Distributor to an instance of SQL Server 2008, but you have a Publisher that is running an instance of SQL Server 2005 and a Subscriber that is running an instance of SQL Server 2000, you are limited to the general functionality and replication functionality of SQL Server 2000.

#### <sup>I</sup>∕ Note:

Because the SQL Server on-disk storage format is the same in the 64-bit and 32-bit environments, a replication topology can combine server instances that run in a 32-bit environment and server instances that run in a 64-bit environment.

For all types of replication, the Distributor version must be no earlier than the Publisher version. (Frequently, the Distributor is the same instance as the Publisher.)

For transactional replication, a Subscriber to a transactional publication can be any version within two versions of the Publisher version. For example, a SQL Server 2000 Publisher can have SQL Server 2008 Subscribers, and a SQL Server 2008 Publisher can have SQL Server 2000 Subscribers.

For merge replication, a Subscriber to a merge publication can be any version no later than the Publisher version. For more information about compatibility for earlier versions, see "Compatibility Level for Merge Publications" later in this topic. For more information about replication features that are supported in the various editions of SQL Server, see Features Supported by the Editions of SQL Server 2008.

# Using a SQL Server 2005 or SQL Server 2008 Distributor with a Publisher Running SQL Server 2000

SQL Server 2005 and SQL Server 2008 can be used as a remote Distributor for Publishers that are running SQL Server 2000. To change agent properties in this scenario, execute the following stored procedures at the Distributor. These procedures let you change properties that were introduced 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 that are running SQL Server 2000, you can change the credentials under which agents make connections by using **sp\_changedistpublisher** and **sp\_changesubscriber**. However, if you upgrade the Distributor to SQL Server 2008, these procedures cannot be used to change the credentials that are used in existing agent jobs. The procedures do affect agent jobs that are created after the procedure is called. To change the credentials for existing agent jobs, call one of the four procedures listed previously.

# <sup>B</sup> Mapping New Data Types for Earlier Versions

SQL Server 2008 and SQL Server 2005 support several new data types. As shown in the following table, these new data types are mapped to compatible data types at the Subscriber if push subscriptions from a SQL Server 2005 or SQL Server 2008 Distributor are used. If new data types are replicated to Subscribers that are running earlier versions of SQL Server, you must verify that the data types are mapped appropriately:

- Mapping is performed by default for articles in merge publications, but not for articles in snapshot or transactional publications. For merge publications, the way in which types are mapped is determined by the publication compatibility level. For example, if a column is of type **geography** and the compatibility level is 90RTM, the type is mapped to **varbinary(max)**. If the compatibility level is 80RTM, the type is mapped to **image**.
- Mapping behavior is controlled by the @schema\_option parameter of sp\_addarticle and sp\_addmergearticle.
   For more information about how to set schema options, see How to:

Specify Schema Options (SQL Server Management Studio) and How to: Specify Schema Options (Replication Transact-SQL Programming).

SQL Server 2008 data type	SQL Server 2005 data type	SQL Server 2000 data type
Common language runtime user-defined type (UDT): 8000 bytes or less	UDT	image
UDT: more than 8000 bytes <sup>1</sup>	varbinary(max)	image
date <sup>2, 3</sup>	nvarchar(10)	nvarchar(10)
datetime2 <sup>2,3</sup>	nvarchar(27)	nvarchar(27)
datetimeoffset <sup>2, 3</sup>	nvarchar(34)	nvarchar(34)
FILESTREAM attribute <sup>1, 4</sup>	varbinary(max)	Not supported
geography and geometry <sup>1, 3</sup>	varbinary(max)	image
hierarchyid <sup>1, 5</sup>	varbinary(max)	image
nvarchar(max)	nvarchar(max)	ntext

time <sup>2, 3</sup>	nvarchar(16)	nvarchar(16)
varchar(max)	varchar(max)	text
varbinary(max)	varbinary(max)	image
xml	xml	ntext

<sup>1</sup> Mappings for UDT, FILESTREAM, **geography**, **geometry**, and **hierarchyid** types are not supported for transactional publications with updatable subscriptions. Only include these types if all updating Subscribers are running SQL Server 2008 or a later version.

<sup>2</sup> Replication does not check the format of data inserted at the Subscriber. Therefore, your application must ensure that inserted data is of the correct format for columns of type **date**, **datetime2**, **datetimeoffset**, and **time**. This is typically done with a constraint. If the data is not of the correct format, inserts at the Publisher will fail.

<sup>3</sup> SQL Server Compact Subscribers convert these types after they are replicated to the Subscriber. For information about data type mappings for SQL Server Compact, see the SQL Server Compact documentation.

If you map columns of type **geography** or **geometry** to **varbinary(max)** or **image**, you cannot replicate default constraints for these columns. This has the following consequences:

- If you already have a default constraint at the Publisher, drop the constraint or specify that it should not be replicated. To specify that it should not be replicated, use the article schema option for default constraints:
  - Select a value of False for the option Copy default value specifications. For more information, see How to: Specify Schema Options (SQL Server Management Studio).
  - Disable the schema option 0x800. For more information, see How to: Specify Schema Options (Replication Transact-SQL Programming).
- If you want to add a default constraint at the Publisher, first specify that schema changes should not be replicated. For more information, see **How to: Replicate Schema Changes (SQL Server Management Studio)** and **How to: Replicate Schema Changes (Replication Transact-SQL Programming)**.
<sup>4</sup> FILESTREAM is an attribute on a **varbinary(max)** column. For information about how to use FILESTREAM columns in replicated tables, see the "Replication" section of **Using FILESTREAM with Other SQL Server Features**. Columns that have the FILESTREAM attribute should not be included in publications that use a character mode snapshot.

<sup>5</sup> Support for columns of type **hierarchyid** depends on the type of replication and the versions of SQL Server that are used. For more information, see the "Using hierarchyid Columns in Replicated Tables" section of **hierarchyid** (**Transact-SQL**). For merge replication, **hierarchyid** is mapped to **image** when the publication compatibility level is 100RTM and a character mode snapshot is used.

# <sup>□</sup> Replicating XML data types

When replicating XML data types to SQL Server Compact 3.5, merge replication maps them to **Ntext**. The XML data on SQL Server 2008 has prefix bytes for UTF-16 encoding. These bytes are preserved when replicating from SQL Server to SQL Server Compact 3.5 by using merge replication. These prefix bytes are not understood by SQL Server Management Studio when viewing the **Ntext** column of the SQL Server Compact 3.5 database. Therefore, these bytes are displayed as garbage characters.

The XML Schema collection in SQL Server 2008 has been updated. This has an effect when replicating the XML columns bound to XML schemas from SQL Server 2008 to SQL Server 2005.

Timezones are not mandatory for date, time and datetime XML Schema values in SQL Server 2008. This means that if no timezone is specified on the SQL Server 2008 publisher XML column, it will not apply the change on SQL Server 2005 subscribers, because SQL Server 2005 requires that a timezone is specified.

Timezone information about the date, time, and datetime XML Schema typed values of SQL Server 2008 Publisher will be converted to the UTC-0 timezone in SQL Server 2005. This is represented by the timezone indicator Z.

SQL Server 2008 XML Schema date, time, and datetime types support larger precision. Therefore, these values get rounded off when replicating to SQL Server 2005.

When replicating XML Schema date or datetime values from SQL Server 2005 to SQL Server 2008, values with a negative year will not apply on SQL Server 2008 because it is not supported on SQL Server 2008.

In these situations, the **sp\_table\_validation** and **Validate** methods in Replication agents might fail. For more information, see the "Upgrading Typed XML from SQL Server 2005 to SQL Server 2008" section in **Typed XML Compared to Untyped XML**.

# <sup>©</sup> Publishing Compressed Data

SQL Server 2008 supports both row and page compression for both tables and indexes. For information about replication support for compressed data, see "How Compression Affects Replication" in **Creating Compressed Tables and Indexes**.

# Restoring a Replicated Database from an Earlier Version

You can keep replication settings when you restore a backup of a replicated database from an earlier version. If you restore the backups to a server and database that have 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 you restore the database, execute **sp\_vupgrade\_replication** to upgrade schema and system data to support replication at the current product level.

Although preserving replication after restoring from backup from earlier 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.

# <sup>©</sup> 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 80RTM (SQL Server 2000 that has no service packs installed) to 100RTM for SQL Server 2008. The compatibility level is specified by one of the following methods:

- By using the @publication\_compatibility\_level parameter of sp\_addmergepublication. For more information, see How to: Set the Compatibility Level for Merge Publications (Replication Transact-SQL Programming).
- On the **Subscriber Types** page of the New Publication Wizard. For more information about how to run 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 Compatibility Level for Merge Publications (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 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 that use ALTER <OBJECT> statements. For more information, see **Making Schema Changes on Publication Databases**.

The following features do not depend on the compatibility level; however, they do require the Merge Agent that is included with SQL Server 2005 and later versions. Subscribers that are running earlier versions of SQL Server function as if 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**.
- Snapshot features for publications that use parameterized filters. These features provide the following abilities:
  - 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 2008**

The following are some important behaviors of the publication compatibility level to consider:

- The publication compatibility level is not connected to the database compatibility level.
- If you create a publication by using **sp\_addmergepublication** 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 that you select on the **Subscriber Types** page of the wizard.
- In versions of SQL Server earlier than SQL Server 2005, the publication compatibility level automatically increased if you enabled a feature that required a higher level. Starting with SQL Server 2005, you must manually set the publication compatibility level to 90RTM or higher before you enable functionality that requires that compatibility level.
- 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 has a lower compatibility level (such as 80RTM) and you want to add another publication in the same database that has a level of 90RTM or higher, you must manually increase the level of the first publication before you add the new publication.
- If a database contains two or more publications that have lower compatibility levels and you want to add another publication in the same database that has a level of 90RTM or higher, you must drop all existing publications except one; increase the level of the remaining publication to 90RTM or higher; re-create the dropped publications with a level of 90RTM or higher; and then create the new publication with a level of 90RTM or higher.

# Required Components and Compatibility Levels for Web Synchronization

SQL Server 2008 supports Web Synchronization for Subscribers that are running SQL Server 2005, SQL Server 2008, and SQL Server Compact versions 3.0, 3.1, and 3.5. The following table lists the publication compatibility level and the server components that are required for each type of Subscriber.

Publisher version	Subscriber version	Required publication compatibility level	Required components on the IIS server
SQL Server 2008	SQL Server 2008	100RTM	SQL Server 2008 IIS components
SQL Server 2008	SQL Server Compact 3.0, 3.1, and 3.5	90RTM	SQL Server Compact 3.5 IIS components and SQL Server 2008 IIS components
SQL Server 2008	SQL Server 2005	90RTM	SQL Server 2008 IIS components
SQL Server 2005	SQL Server 2005	90RTM	SQL Server 2005 IIS components
SQL Server 2005	SQL Server Compact 3.0, 3.1, and 3.5	90RTM	SQL Server Compact 3.5 IIS components and SQL Server 2005 IIS components
SQL Server 2005	SQL Server 2008	Not applicable <sup>1</sup>	Not applicable <sup>1</sup>

<sup>1</sup> This configuration is not supported because the Publisher version must be equal to or greater than the Subscriber version.

### $\Box$ See Also

#### Concepts

Replication Backward Compatibility Considerations for Upgrading Replicated Databases

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Migrating to SQL Server 2008

# 

#### -

Migrating a SQL Server installation to SQL Server 2008 is useful if you encounter 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 includes manual and automated steps. The following topics describe the migration process.

# <sup>□</sup> In This Section

Торіс	Description
<u>How to: Migrate a</u> <u>Reporting Services</u> <u>Installation</u>	Provides information about how to migrate previous Reporting Services versions to SQL Server 2008.
Upgrading Integration	Provides information about how to migrate
<u>Services Packages</u>	Integration Services packages to SQL Server 2008.
Migrating Data	Provides information about how to migrate Data
Transformation Services	Transformation Services packages to SQL Server
Packages	2008.

## □ See Also

### Concepts

Backward Compatibility

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Upgrading Integration Services Packages



#### - -

Migrating to SQL Server 2008 >

When you upgrade an instance of SQL Server 2005 to SQL Server 2008, your existing SQL Server 2005 Integration Services (SSIS) packages are not automatically upgraded to the package format that SQL Server 2008 Integration Services (SSIS) uses. You will have to select an upgrade method and manually upgrade your SQL Server 2005 packages.

When you upgrade a SQL Server 2005 package, Integration Services migrates the scripts in any Script task and Script component to Microsoft Visual Studio Tools for Applications (VSTA). In SQL Server 2005, the scripts in Script tasks or Script components used Microsoft Visual Studio for Applications (VSA). For more information about changes that you might have to make to the scripts before migration and about script conversion failure, see <u>Migrating Scripts to</u> <u>VSTA</u>.

# <sup>□</sup> Selecting an Upgrade Method

You can use various methods to upgrade SQL Server 2005 packages. For some of these methods, the upgrade is only temporary. For others, the upgrade is permanent. The following table describes each of these methods and whether the upgrade is temporary or permanent.

Upgrade Method	Type of Upgrade
Use the <b>dtexec</b> utility (dtexec.exe) that is installed with SQL Server 2008 to run a SQL Server 2005 package.	The package upgrade and script migration are temporary.
For more information, see <b>dtexec Utility</b> .	The changes cannot be saved.
Add a SQL Server 2005	The package upgrade is permanent if you save the package; otherwise, it is temporary if you do not save the package.
package to an existing project or open a SQL Server 2005 package in SQL Server 2008 Integration Services.	The script migration is permanent if you add the package to an existing project. The script migration is also permanent if you open the package and save the conversion changes; otherwise, it is temporary if you do no save the conversion changes.
Use the SSIS Package Upgrade Wizard to create or open a SQL Server 2005 Integration Services project, and upgrade multiple packages in the project. For more information, see <u>How</u> to: <u>Upgrade Integration</u> <u>Services Packages Using the</u> <u>SSIS Package Upgrade Wizard</u> and <b>SSIS Package Upgrade</b>	The package upgrade and script migration are permanent.

Wizard F1 Help.

# <sup>©</sup> Understanding Package Upgrade Results

During the package upgrade process, most components and features in SQL Server 2005 packages convert seamlessly to their counterparts in SQL Server 2008. However, there are several component and features that either will not be upgraded or have upgrade results of which you should be aware. The following table identifies these components and features.

#### <sup>I</sup>Note:

To identify which packages have the issues listed in this table, run Upgrade Advisor. For more information, see <u>Using Upgrade Advisor to Prepare for</u> <u>Upgrades</u>.

Component or Feature	Upgrade Results	
Connection strings	<ul> <li>The names of certain providers have changed in SQL Server 2008 and require different values in the connection strings.</li> <li>To update the connection strings, use one of the following procedures:</li> <li>Use the SSIS Package Upgrade Wizard to upgrade the package, and select the Update connection strings to use new provider names option.</li> <li>In Business Intelligence Development Studio, in the Options dialog box, selec the Update connectionstrings to use new provider names option.</li> <li>In Business Intelligence Development Studio, open the package and manually change the text of the ConnectionString property.</li> </ul>	
Lookup transformation	The package upgrade process automatically upgrades the Lookup transformation to the SQL Server 2008 version. However, the SQL Server 2008 version of this component has some additional capabilities of which you might want to take advantage.	

	For more information, see <b>Lookup Transformation</b> .	
Script task and Script component	The package upgrade process automatically migrates scripts in the Script task and Script component from VSA to VSTA. For more information about changes that you might have to make to the scripts before migration and about script conversion failure, see <u>Migrating Scripts to VSTA</u> .	
Third-party and custom components	These components will not be upgraded. You must manually recompile these components to enable them to work with SQL Server 2008 Integration Services.	

### $\Box$ See Also

### Concepts

Migrating to SQL Server 2008

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Migrating Scripts to VSTA



#### - -

<u>Migrating to SQL Server 2008</u> > <u>Upgrading Integration Services Packages</u> >

When you upgrade SQL Server 2005 packages to SQL Server 2008, Integration Services migrates the scripts in any Script tasks or Script components to Microsoft Visual Studio Tools for Applications (VSTA). VSTA is the scripting environment that SQL Server 2008 Integration Services (SSIS) uses. In SQL Server 2005, the scripting environment for Integration Services is Microsoft Visual Studio for Applications (VSA).

If the scripts reference interfaces or use user-defined type values, you may need to modify those scripts before you upgrade the package. Otherwise, the package will fail to upgrade or the script will fail to validate, depending on the upgrade method that you use. The following table describes the code modifications.

Task	Description	
Modify references to interfaces	Replace references to IDTS <i>xxx</i> 90 interfaces with references to the corresponding IDTS <i>xxx</i> 100 interfaces. This task applies to Script tasks and Script components.	
Modify type objects	If user-defined type values in runtime variables are not defined in the assemblies mscorlib.dll or Microsoft.SqlServer.VSTAScriptTaskPrx.dll, change the types to inherit from System.MarshalByRefObject. This task applies only to Script tasks.	

For more information about how to migrate scripts and upgrade packages, see <u>Upgrading Integration Services Packages</u>.

# <sup>D</sup> Understanding Migration Failures

When you migrate the scripts, the migration can fail because of one of the following reasons:

- The entry point for the VSA script was renamed. The entry point specifies the method in the **ScriptMain** class in the VSTA project that the Integration Services runtime calls as the entry point into the Script task code. The **ScriptMain** class is the default class that the script templates generate.
- There is no entry point or there are multiple entry points in the VSA script.
- Assembly references could not be added.
- The **ScriptMain** class was modified to inherit from other classes in addition to the **ScriptObjectModelSSIS** class. Microsoft .NET Framework does not support multiple inheritance.

You cannot convert a VSA script that uses Visual Basic 2005 to a VSTA script that uses Microsoft Visual C# 2008. However, you can create a new VSTA script that uses Microsoft Visual C# 2008 For more information, see **Coding and Debugging the Script Task** and **Coding and Debugging the Script Component**.

### $\Box$ See Also

#### Other Resources Extending Packages with Scripting

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Upgrade Integration Services Packages Using the SSIS Package Upgrade Wizard

- -

<u>Migrating to SQL Server 2008</u> > <u>Upgrading Integration Services Packages</u> >

You can upgrade packages that were created in earlier versions of Integration Services to the Integration Services format that SQL Server 2008 uses. SQL Server provides the SSIS Package Upgrade Wizard to help in this process. Because you can configure the wizard to backup up your original packages, you can continue to use the original packages if you experience upgrade difficulties.

The SSIS Package Upgrade Wizard is installed when Integration Services is installed.

#### <sup>✓</sup>Note:

The SSIS Package Upgrade Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server.

For more information about how to upgrade Integration Services packages, see <u>Upgrading Integration Services Packages</u>.

#### <sup>I</sup>∕ Note:

For information about migrating packages that were created in SQL Server 2000 Data Transformation Services (DTS), see <u>Migrating Data</u> <u>Transformation Services Packages</u>.

The remainder of this topic describes how to run the wizard and back up the original packages.

# <sup>□</sup> Running the SSIS Package Upgrade Wizard

You can run the SSIS Package Upgrade Wizard from Business Intelligence Development Studio, from SQL Server Management Studio, or at the command prompt.

#### <sup>©</sup> To run the wizard from Business Intelligence Development Studio

- 1. In Business Intelligence Development Studio, create or open an Integration Services project.
- 2. In Solution Explorer, right-click the **SSIS Packages** node, and then click **Upgrade All Packages** to upgrade all the packages under this node.

#### <sup> </</sup> Note:

When you open an Integration Services project that contains SQL Server 2005 Integration Services (SSIS) packages, Integration Services automatically opens the SSIS Package Upgrade Wizard.

## <sup>□</sup> To run the wizard from SQL Server Management Studio

• In SQL Server Management Studio, connect to Integration Services, expand the **Stored Packages** node, and right-click the **File System** or **MSDB** node, and then click **Upgrade Packages**.

## <sup>□</sup> To run the wizard at the command prompt

• At the command prompt, run the SSISUpgrade.exe file from the

C:\Program Files\Microsoft SQL Server\100\DTS\Binn folder.

# <sup>Backing</sup> Up the Original Packages

To back up the original packages, both the original packages and the upgraded packages must be stored in the same folder in the file system. Depending on how you run the wizard, this storage location might be automatically selected:

- When you run the SSIS Package Upgrade Wizard from Business Intelligence Development Studio, the wizard automatically stores both the original packages and upgraded packages in the same folder in the file system. You can configure the wizard to back up the original packages.
- When you run the SSIS Package Upgrade Wizard from SQL Server Management Studio or at the command prompt, you can specify different storage locations for the original and upgraded packages. To back up the original packages, make sure to specify that both the original and upgraded packages are stored in the same folder in the file system. If you specify any other storage options, the wizard will not be able to back up the original packages.

When the wizard backs up the original packages, the wizard stores a copy of the original packages in an **SSISBackupFolder** folder. The wizard creates this **SSISBackupFolder** folder as a subfolder to the folder that contains the original packages and the upgraded packages.

#### <sup>©</sup> To back up the original packages in SQL Server Management Studio or at the command prompt

1. Save the original packages to a location on the file system.

#### <sup>I</sup>Note:

The backup option in the wizard only works with packages that have been stored to the file system.

2. In SQL Server Management Studio or at the command prompt, run the SSIS Package Upgrade Wizard.

- 3. On the **Select Source Location** page of the wizard, set the **Package source** property to **File System**.
- 4. On the **Select Destination Location** page of the wizard, select **Save to source location** to save the upgraded packages to the same location as the original packages.

#### <sup>ℤ</sup>Note:

The backup option in the wizard is available only when the upgraded packages are stored in the same folder as the original packages.

5. On the **Select Package Management Options** page of the wizard, select the **Backup original packages** option.

# <sup>□</sup> To back up the original packages in Business Intelligence Development Studio

1. Save the original packages to a location on the file system.

#### <sup>I</sup>Note:

The backup option in the wizard only works with packages that have been stored to the file system.

- 2. In Business Intelligence Development Studio, run the SSIS Package Upgrade Wizard.
- 3. On the **Select Package Management Options** page of the wizard, select the **Backup original packages** option.

© 2008 Microsoft Corporation. All rights reserved.

### SQL Server Setup Help How to: Migrate a Reporting Services Installation



#### - -

#### <u>Migrating to SQL Server 2008</u> >

This topic provides step-by-step instructions for migrating a SQL Server 2000 or SQL Server 2005 Reporting Services deployment to a new SQL Server 2008 Reporting Services instance. Migration is defined as moving application data files to a new SQL Server 2008 instance. You must migrate your installation 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:

- Back up database, application, and configuration files.
- Back up the encryption key.
- Install a new instance of SQL Server 2008. If you are using the same hardware, you can install SQL Server 2008 side-by-side your existing SQL Server 2000 or 2005 installation. Be aware that if you do this, you might need to install SQL Server 2008 as a named instance.
- Move the report server database and other application files from your SQL Server 2000 or 2005 installation to your new SQL Server 2008 installation.
- Move any custom application files to the new installation.
- Configure the report server.
- Edit RSReportServer.config to include any custom settings from your previous installation.
- Optionally, configure custom Access Control Lists (ACLs) for the new Reporting Services Windows service group.
- Test your installation.
- Remove unused applications and tools 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 <u>Creating a Report Server Database</u>.

# <sup>□</sup> Before You Start

Even though you are migrating rather than upgrading the installation, consider running Upgrade Advisor on your existing installation help identify any issues that could affect migration. This step is especially helpful if you are migrating a report server that you did not install or configure. By running Upgrade Advisor, you can find out about custom settings that might not be supported in a new SQL Server 2008 installation.

In addition, you should be aware of several important changes in SQL Server 2008 that will affect how you migrate your installation:

- IIS is no longer a prerequisite. If you are migrating a report server installation to a new computer, you do not need to add the Web server role. In addition, steps for configuring URLs and authentication are different from the previous release, as are techniques and tools for diagnosing and troubleshooting problems.
- Report Server Web service, Report Manager, and the Report Server Windows service are consolidated within a single Report Server service. All three applications run under the same account. All three applications read configuration settings from RSReportServer.config file, making RSWebApplication.config obsolete.
- Report Manager and SQL Server Management Studio are redesigned to remove overlapping features. Each tool supports a distinct set of tasks; the tools are no longer interchangeable.
- ISAPI filters are not supported in SQL Server 2008 Reporting Services. If you use ISAPI filters, you must redesign your reporting solution prior to migration.
- IP address restrictions are not supported in SQL Server 2008 Reporting Services. If you use IP address restrictions, you must redesign your reporting solution prior to migration or use a technology such as a firewall, router, or Network Address Translation (NAT) to configure addresses that are restricted from accessing the report server.
- Client Secure Sockets Layer (SSL) certificates are not supported in SQL Server 2008 Reporting Services. If you use client SSL certificates, you must redesign your reporting solution prior to migration.
- If you use an authentication type other than Windows Integrated

authentication, you must update the <AuthenticationTypes> element in the RSReportServer.config file with a supported authentication type. The supported authentication types are NTLM, Kerberos, Negotiate, and Basic. Anonymous, .NET Passport, and Digest authentication are not supported in SQL Server 2008 Reporting Services.

• If you use custom cascading style sheets in your reporting environment, they will not be migrated. You must manually move them following migration.

For more information about changes in SQL Server 2008 Reporting Services, see the Upgrade Advisor documentation and **What's New (Reporting Services)**.

# <sup>Backup</sup> Files and Data

Before you install a new instance of Reporting Services, be sure to back up all of the files in your current installation.

- 1. Back up the encryption key for the report server database. This step is critical to migration success. Further on in the migration process, you must restore it for the report server to regain access to encrypted data. To back up the key, use the Reporting Services Configuration tool (or **rskeymgmt** utility if you are migrating from SQL Server 2000).
- 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 back up the report server database in **Moving the Report Server Databases to Another Computer**.
- 3. Back up the report server configuration files. Files to back up include:
  - 1. Rsreportserver.config
  - 2. Rswebapplication.config
  - 3. Rssvrpolicy.config
  - 4. Rsmgrpolicy.config
  - 5. Reportingservicesservice.exe.config
  - 6. Web.config for both the Report Server and Report Manager ASP.NET applications.
  - 7. Machine.config for ASP.NET if you modified it for report server operations.

# <sup>□</sup> Install SQL Server 2008 Reporting Services

Install a new report server instance in files-only mode so that you can configure it to use non-default values. For command line installation, use the **FilesOnly** argument. In the Installation Wizard, select the **Install but do not configure option**.

Click one of the following links to view instructions on how to install a new instance of Reporting Services:

- How to: Install SQL Server 2008 (Setup)
- How to: Install SQL Server 2008 from the Command Prompt

# <sup>□</sup> Move the Report Server Database

The report server database contains published reports, models, shared data sources, schedules, resources, subscriptions, and folders. It also contains system and item properties, and permissions for accessing report server content.

If your migration includes using a different Database Engine instance, you must move the report server database to the new Database Engine instance. If you are using the same Database Engine instance, skip to "Configure the Report Server".

To move the report server database, do the following:

- 1. Choose the Database Engine instance to use. SQL Server 2008 Reporting Services requires that you use SQL Server 2005 or 2008 to host the report server database. SQL Server 2000 can no longer be used to host the report server database for SQL Server 2008 Reporting Services. If you used SQL Server 2000 or SQL Server 2005 running in 8.0 compatibility mode, you must upgrade the Database Engine or choose another Database Engine instance.
- 2. Start SQL Server Management Studio and connect to the Database Engine.
- 3. Create the **RSExecRole** in the system databases if the Database Engine has never hosted a report server database. For more information, see **How to: Create the RSExecRole**.
- 4. Follow the instructions in **Moving the Report Server Databases to Another Computer**.

Remember that both the report server database and the temporary database are interdependent and must be moved together. Do not copy the databases; copying does not transfer all of the security settings to the new installation. Do not move SQL Server Agent jobs for scheduled report server operations. The report server will recreate these jobs automatically.

# <sup>□</sup> Move Custom Assemblies or Extensions

If your installation includes custom report items, assemblies, or extensions, you must redeploy the custom components. If you are not using custom components, skip to "Configure the Report Server".

To redeploy the custom components, do the following:

- 1. Determine whether the assemblies are supported or need recompilation:
  - Custom authentication extensions created for the SQL Server 2000 release must be recompiled.
  - Custom rendering extensions for SQL Server 2008 Reporting Services must be rewritten using the Rendering Object Model (ROM).
  - HTML 3.2 and HTML OWC renderers are not supported in SQL Server 2008 Reporting Services.
  - Other custom assemblies should not require recompilation.
- 2. Move the assemblies to the new report server and Report Manager \bin folders. In SQL Server 2008, the report server binaries are located in \Program files\Microsoft SQL Server\MSRS10.MSSQLSERVER\Reporting Services\ReportServer\bin for the default SQL Server 2008 Reporting Services instance.
- 3. Modify the configuration files to add entries for your custom component. The entries will vary depending on the kind of assembly you are using. For instructions on where to place files and add configuration entries, see the following:
  - 1. **Deploying a Custom Assembly**
  - 2. How to: Deploy a Custom Report Item
  - 3. Deploying a Data Processing Extension
  - 4. Deploying a Delivery Extension
  - 5. Deploying a Rendering Extension
  - 6. Implementing a Security Extension

# <sup>□</sup> Configure the Report Server

Configure URLs for the Report Server Web service and Report Manager, and configure the connection to the report server database.

If you are migrating a scale-out deployment, take all of the report server nodes offline and migrate each server one at a time. Once the first report server is migrated and it successfully connects to the report server database, the report server database version is automatically upgraded to the SQL Server 2008 database version.

### <sup>I</sup>Note:

If any of the report servers in the scale-out deployment are online and have not been migrated, they might encounter an rsInvalidReportServerDatabase exception because they are using an older schema when connected to the upgraded.

- 1. Start the Reporting Services Configuration tool and connect to the Reporting Services instance you just installed. For more information, see <u>How to: Start Reporting Services Configuration</u>.
- 2. Configure URLs for the report server and Report Manager. For more information, see <u>How to: Configure a URL (Reporting Services</u> <u>Configuration</u>).
- 3. Configure the report server database, selecting the existing report server database from your previous installation. After successful configuration, the report server services will restart, and once a connection is made to the report server database, the database will be automatically upgraded to SQL Server 2008 Reporting Services. For more information about how to run the Change Database Wizard that you use to create or select a report server database, see <u>How to: Create a Report Server Database (Reporting Services Configuration</u>).
- 4. Restore the encryption keys. This step is necessary for enabling reversible encryption on pre-existing connection strings and credentials that are already in the report server database. For more information, see <u>Backing Up</u> and <u>Restoring Encryption Keys</u>.
- 5. If you installed report server on a new computer and you are using Windows Firewall, be sure that the TCP port on which the report server

listens is open. By default, this port is 80. For more information, see <u>How</u> <u>to: Configure a Firewall for Report Server Access</u>.

6. If you installed report server on Windows Vista or Windows Server 2008, create role assignments and add the report server site to Trusted Sites to support local administration. For more information, see <u>How to: Configure a Report Server for Local Administration on Windows Vista and Windows Server 2008</u>.
# Copy Custom Configuration Settings to RSReportServer.config File

If you modified the RSReportServer.config file or RSWebApplication.config file in the previous installation, you should make the same modifications in the new RSReportServer.config file. The following list summarizes some of the reasons why you might have modified the previous configuration file and provides links to additional information about how to configure the same settings in SQL Server 2008.

Customization	Information
Report Server E-mail delivery with custom settings	<u>How to: Configure a Report Server for E-mail</u> Delivery (Reporting Services Configuration)
Data processing extensions	Configuring Data Processing Extensions in Reporting Services
Device information settings	Customizing Rendering Extension Parameters in RSReportServer.Config
Report Manager on a remote instance	How to: Configure Report Manager

## <sup>©</sup> Windows Service Group and Security ACLs

In SQL Server 2008 Reporting Services, there is one service group, the Reporting Services Windows Service group, which is used to create security ACLs for all the registry keys, files, and folders that are installed with SQL Server 2008 Reporting Services. This Windows group name appears in the format SQLServerReportServerUser\$<*computer\_name*>\$*<instance\_name*>. This group takes the place of the two Windows service groups in SQL Server 2005 Reporting Services. If you have custom ACLs associated with either of the SQL Server 2005 Reporting Services Windows groups, you will need to apply those ACLs to the new group for your new report server instance in SQL Server 2008 Reporting Services.

## □ Verify Your Deployment

- 1. Test the report server and Report Manager virtual directories by opening a browser and typing in the URL address. For more information, see <u>How to:</u> <u>Verify a Reporting Services Installation</u>.
- 2. 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. The report server uses the SQL Server 2008 report object model when processing and rendering reports, but it does not replace SQL Server 2000 or 2005 constructs with new report definition language elements. To learn more about how existing reports run on a SQL Server 2008 report server, see <u>Upgrading Reports</u>.

## <sup>□</sup> Remove Unused Programs and Files

Once you have successfully migrated your report server to a SQL Server 2008 Reporting Services instance, you might want to perform the following steps to remove programs and files that are no longer necessary.

- 1. Uninstall the previous version of Reporting Services if you no longer need it. This step does not delete the following items, but you can manually remove them if you no longer need them:
  - The old Report Server database
  - RsExec role
  - Report Server service accounts
  - Application pool for the Report Server Web service
  - Virtual directories for Report Manager and the report server
  - Report server log files
- 2. Remove IIS if you no longer need it on this computer.
- 3. Delete RSActivate.exe (from SQL Server 2000 installations only).

## □ See Also

**Concepts** <u>Considerations for Upgrading Reporting Services</u>

Other Resources Report Server Database <u>Reporting Services Backward Compatibility</u> Reporting Services Configuration Tool

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Configuring SQL Server 2008

#### - -

After you have installed SQL Server, you can further configure SQL Server by using graphical and command-prompt utilities. The following table describes support for tools that are used to manage an instance of SQL Server:

Tool or utility	Description
SQL Server Management Studio	SQL Server Management Studio is used to edit and execute queries, and to start standard wizard tasks. For more information, see <b>Introducing SQL Server Management Studio</b> .
SQL Server Profiler	SQL Server Profiler provides a graphical user interface to monitor an instance of the SQL Server Database Engine or an instance of Analysis Services. For more information, see <b>Introducing SQL</b> <b>Server Profiler</b> .
Database Engine Tuning Advisor	The Database Engine Tuning Advisor helps you to create optimal sets of indexes, indexed views, and partitions. For more information, see <b>Using Database Engine Tuning Advisor</b> .
Business Intelligence Development Studio	Business Intelligence Development Studio is an integrated development environment for Analysis Services and Integration Services solutions. For more information, see <b>Introducing Business Intelligence Development Studio</b> .
Command Prompt Utilities	You can manage SQL Server objects from the command prompt. For more information, see <b>Command Prompt Utilities</b> .
SQL Server Configuration Manager	You can manage server and client network configuration settings. For more information, see <b>SQL Server Configuration Manager</b> .
Import and Export Data	Integration Services provides graphical tools and programmable objects to move, copy, and transform data. For more information, see <b>Overview (Integration Services)</b> .

SQL Server	Install, upgrade to, or change components in an instance of SQL
Setup	Server. For more information, see Installation How-to Topics.

## $\Box$ See Also

Concepts Configuring Reporting Services

Other Resources Administration (Database Engine)

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Validating a SQL Server Installation



#### - ▼

<u>Configuring SQL Server 2008</u> >

To verify a successful installation of SQL Server 2008, ensure that the services for the components you choose to install are running on your computer.

Use the following list of services to verify that the SQL Server service is running. The following table lists service display names and the services they provide.

## <sup>☑</sup> Note:

The actual service names differ slightly from their display names. Service names can be viewed by right-clicking a service and selecting **Properties**.

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 specified path.

Name	Service
SQL Server (MSSQLSERVER)	The default instance of the SQL Server Database Engine.
SQL Server (instancename)	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 (instancename)	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 Services (instancename)	A named instance of Analysis Services, where <i>instancename</i> is the name of the instance.
Reporting Services	The default instance of Reporting Services.
Reporting Services ( <i>instancename</i> )	A named instance of Reporting Services, where <i>instancename</i> is the name of the instance.

## $\Box$ See Also

### Tasks

How to: Read a SQL Server 2008 Setup Log File

### Concepts

How to: View SQL Server 2008 Setup Log Files

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Configuring Reporting Services



#### - -

#### Configuring SQL Server 2008 >

Use the Reporting Services Configuration tool to configure a SQL Server Reporting Services installation. If you installed a report server using the filesonly 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.

The Reporting Services Configuration tool 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.

# <sup>□</sup> In This Section

Term	Def
Configuring the Report	Provides recommendations and steps on how to modify
Server Service Account	the service account and password.
Configuring Report	Describes how to configure URLs used to access the
<u>Server URLs</u>	Report Server Web service and Report Manager.
Creating a Report	Describes how to create a report server database,
<u>Server Database</u>	required for storing server metadata and objects.
Configuring a Report	Describes how to modify the connection string used by
<u>Server Database</u>	the report server to connect to the report server
<u>Connection</u>	database.
Configuring the	Describes how to configure a user account to process
Unattended Execution	reports in unattended mode.
<u>Account</u>	- op on the manufacture and the
Configuring a Report	Describes how to configure a report server to support e-
<u>Server for E-Mail</u>	mail report distribution.
Delivery	1
Configuring Reporting	Provides information about configuring multiple report
Services for Scale-Out	server instances to use a shared report server database.
<u>Deployment</u>	
Configuring Reporting	Provides an overview of the tasks and requirements for
Services for SharePoint	adding a report server to an existing deployment of a
<u>S.0 Integration</u>	SharePoint product or technology.
<u>Configuring and</u>	Explains how to use and manage encryption keys that
<u>Managing Encryption</u>	are used when storing sensitive data.
<u>Neys</u> Deporting Services	
<u>Reporting Services</u>	Drovides stop by stop instruction for common tasks
Configuration from-to	Provides step-by-step instruction for common tasks.
Deporting Services	
Configuration F1 Holp	Provides help topics for the pages in the Reporting
Tonics	Services Configuration tool.
TOPICO	

# □ In SQL Server Books Online

The online help for the Reporting Services Configuration tool is a subset of the product documentation for SQL Server Reporting Services. Complete product documentation can be found in SQL Server Books Online.

When you use SQL Server Books Online, you have access to all the topics provided in this online help system, plus additional content about deploying, administering, and securing a Reporting Services installation. For more information, see **Getting Started with SQL Server Books Online**.

## □ See Also

### Concepts

**Considerations for Installing Reporting Services** 

### **Other Resources**

New Installation (Reporting Services) Upgrade (Reporting Services) Configuring a Report Server Installation

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Configuring the Report Server Service Account



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

Reporting Services is implemented as a single service that contains a Report Server Web service, Report Manager, and a background processing application that is used for scheduled report processing and subscription delivery. This topic explains how the service account is initially configured and how to modify the account or password using the Reporting Services Configuration tool. For more information about the applications that run in the service, see **Service Architecture (Reporting Services)** in SQL Server Books Online.

# <sup>□</sup> Initial Configuration

The Report Server service account is defined during Setup. You can run the service under a domain user account or a built-in such as **NetworkService** account. There is no default account; whatever account you specify in the <u>SQL</u> <u>Server Configuration - Service Accounts</u> page of the Installation Wizard becomes the initial account of the Report Server service.

## <sup>I</sup>Important:

Although the Report Server Web service and Report Manager are ASP.NET applications, they do not run under the ASP.NET account. The single service architecture runs both ASP.NET applications within the same Report Server process identity. This is an important change from previous releases, where both the Report Server Web service and Report Manager ran under the ASP.NET worker process identity specified in IIS.

# <sup>□</sup> Changing the Service Account

To view and reconfigure service account information, always use the Reporting Services Configuration tool. Service identity information is stored internally in multiple locations. Using the tool ensures that all references are updated accordingly whenever you change the account or password. The Reporting Services Configuration tool performs the following additional steps to ensure the report server remains available:

- Automatically adds the new account 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.
- Automatically updates the login permissions on the SQL Server Database Engine instance used to host the report server database. The new account will be added to the **RSExecRole**.

The database login for the old account will not be removed automatically. Be sure to remove accounts that are no longer in use. For more information, see **Administering a Report Server Database**.

Granting database permissions to new service account only occurs if you configured the report server database connection to use the service account in the first place. If you configured the report server database connection to use a domain user account or a SQL Server database login, the connection information is not affected by the service account update.

• Automatically updates the encryption key to include the profile information of the new account.

## <sup>☑</sup>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.

For instructions on how to set the account, see <u>How to: Configure a Service</u> <u>Account for Reporting Services</u>.

# <sup>□</sup> Choosing an Account

You can configure the Report Server service to run under any of these account types:

- Least-privileged Windows user account
- NetworkService
- LocalSystem
- LocalService

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 service to run under a domain user account so that you can avoid widespread damage if a shared account is compromised by a malicious user. It also makes it easier to audit the logon activity for this account. A trade-off to using a Windows user account is that if you are deploying Reporting Services in a network that uses Kerberos authentication, you must register the service with the user account. For more information, see <u>How to: Register a Service Principal Name (SPN) for a Report Server</u>.

The following guidelines and links in this section can help you decide on an approach that is best for your deployment.

- <u>Service Account (Reporting Services Configuration)</u>.
- <u>Setting Up Windows Service Accounts</u> in SQL Server Books Online.
- <u>The Services and Service Accounts Security Planning Guide</u> on MSDN.

# <sup>D</sup> Updating an Expired Password

If the Report Server service runs under a domain account and the password expires before you can update it in the Reporting Services Configuration tool, the service will not start until you specify a new password. If the service cannot be started, you cannot use the Reporting Services Configuration tool to connect to that server to update the account. In this case, you must use a combination of tools to bring the server back online.

To reset the password, do the following:

- 1. On the **Start** menu, point to **Control Panel**, point to **Administrator Tools**, and click **Services**.
- 2. Right-click **SQL Server Reporting Services**, select **Properties**.
- 3. Click **Log On**, and type the new password.
- 4. After you update the password, start the Reporting Services Configuration tool and update the password in the Service Account page. This additional step is necessary to update the account information that is stored internally by the report server.

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**.

# Configuring the Report Server Service for a SharePoint Integrated Report Server

If you are running a report server in SharePoint integrated mode, you must update the service account information that is stored in the SharePoint configuration database if either of the following conditions are true:

- Modifying the Reporting Services service account (for example, switching from NetworkService to a domain user account).
- Extending a SharePoint farm to include an additional SharePoint Web application. If the server farm is configured for report server integration, and newly added application is configured to run under a different user account than other applications in the farm, you must update the database access information.

After you reset the database access information, you should then restart the Windows SharePoint Services service to ensure that the old connection is no longer used.

- 1. In **Administrative Tools**, click **SharePoint 3.0 Central Administration**.
- 2. Click **Application Management**.
- 3. In the Reporting Services section, click **Grant Database Access**.
- 4. Click **OK**. The Enter Credentials dialog box appears.
- 5. Enter the credentials of a user who is a member of the Local Administrator's group on the computer that hosts the report server. The credentials will be used for a one-time connection to the report server computer for the purpose of retrieving service account information. The database login that is created for each service account will be updated in SharePoint databases.
- 6. To restart the service, click **Operations**.
- 7. In Topology and Services, click **Services on Server**.
- 8. For Windows SharePoint Services Web Application, click **Stop**.
- 9. Wait for the service stop.
- 10. Click Start.

For more information, see <u>How to: Configure Report Server Integration in</u> <u>SharePoint Central Administration</u>.

## □ See Also

#### Tasks

<u>How to: Configure a Service Account for Reporting Services</u> <u>How to: Start Reporting Services Configuration</u>

#### Reference

<u>Service Account (Reporting Services Configuration)</u> <u>Setting Up Windows Service Accounts</u>

#### Concepts

**Configuring Report Server URLs** 

#### **Other Resources**

Deployment (Reporting Services) Planning for Services, Accounts, and Connections Starting and Stopping the Report Server Service

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Configuring Report Server URLs



#### - ▼

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

In Reporting Services, URLs are used to access the Report Server Web service and Report Manager. Before you can use either application, you must configure at least one URL each for the Web service and Report Manager. Reporting Services provides default values for both application URLs that work well in most deployment scenarios, including side-by-side deployments with other Web services and applications.

- If you installed the default configuration, URLs were created automatically using the default values.
- If you are using the Reporting Services Configuration tool to create or modify the URLs, you can accept the default values for a URL or specify custom values. A test link of the URL appears on page when you define the URL so that you can immediately confirm that the settings you specified result in a valid connection. For step-by-step instructions on how to configure and test a URL, see <u>How to: Configure a URL</u> (Reporting Services Configuration).

# <sup>□</sup> Defining a Report Server URL

The URL precisely identifies the location of an instance of a report server application on your network. When you create a report server URL, you must specify the following parts.

Part	Description		
	A TCP/IP network uses an IP address to uniquely identify a device on		
	the network. There is a physical IP address for each network adapter		
Host	card installed in a computer. If the IP address resolves to a host header,		
name	you can specify the host header. If you are deploying the report server		
	on a corporate network, you can use the network name of the		
	computer.		
Dort	A TCP port is an endpoint on the device. The report server will listen		
ron	for requests on a designated port.		
	A port is often shared by multiple Web services or applications. For		
Virtual	this reason, a report server URL always includes a virtual directory		
directory	that corresponds to the application that gets the request. You must		
	specify unique virtual directory names for each Reporting Services		
	application that listens on the same IP address and port.		
	URLs in Reporting Services can be configured to use an existing SSL		
SSL	certificate that you previously installed on the computer. For more		
settings	information, see <b>Configuring a Report Server for Secure Sockets</b>		
	Layer (SSL) Connections.		

# <sup>□</sup> Default URLs

When you access a report server or Report Manager through its URL, the URL should include the host name and not the IP address. On a TCP/IP network, the IP address will resolve to a host name (or the network name of the computer). If you used the default values to configure URLs, you should be able to access the Report Server Web service using URLs that specify the computer name or localhost as the host name:

- http://<computername>/reportserver
- http://localhost/reportserver

The settings that make these URLs available appear in the following table. This table shows the default values that enable a report server connection though URLs that include a host name:

Part	Value	Explanation
IP address	All Assigned	The domain name service on your network resolves the host name on the URL to the computer's IP address. As
		define, a request that is sent to a specific host will reach its intended target.
Port	80	Port 80 is the default port for TCP/IP connections on a computer. Because the report server is listening on port 80, you can omit the port number from the URL. If you specify another port, you must specify it in the URL.
Virtual directory	ReportServer	Notice that both of the example URLs includes the virtual directory name. Unless you customize the URL definition, you must always specify the application's virtual directory name on the URL.

#### **Note:**

An underlying URL reservation enables any valid host name to be used on a URL. The Reporting Services Configuration tool creates a URL reservation in HTTP.SYS using syntax that allows variations of the host name to resolve to a particular report server instance. For more information about URL reservations, see <u>About URL Reservations and Registration (Reporting Services)</u>.

## <sup>©</sup> Server-side Permissions on a Report Server URL

Permissions on each URL endpoint are granted exclusively to the Report Server service account. Only this account has rights to accept requests that are directed to the Reporting Services URLs. Discretionary Access Control Lists (DACLs) are created and maintained for the account when you configure the service identity through Setup or the Reporting Services Configuration tool. If you change the service account, the Reporting Services Configuration tool will update all URL reservations that you created to pick up the new account information. For more information, see <u>URL Reservation Syntax (Reporting Services</u>).

# □ Authenticating Client Requests Sent to a Report Server URL

By default, the authentication type supported on the URL endpoints is Windows authentication. This is the default security extension. If you are implementing a custom or Forms authentication provider, you must modify the authentication settings on the report server. Optionally, you can also change the Windows authentication settings to match the authentication subsystem used in your network. For more information, see **Configuring Authentication in Reporting Services**.

# <sup>□</sup> In This Section

#### How to: Configure a URL (Reporting Services Configuration)

This topic provides instructions for setting and modifying a URL reservation in the Reporting Services Configuration tool.

#### About URL Reservations and Registration (Reporting Services)

URLs are used to access applications and reports. This topic explains the application URLs, the default URLs, and how URL reservations and registration work in Reporting Services.

#### URL Reservation Syntax (Reporting Services)

The default URL reservations that Reporting Services uses are valid for most scenarios. However, if you want to restrict access or extend the deployment to enable Internet or extranet access, you might have to customize the settings to fit your requirements. This topic describes the syntax of a URL reservation and provides recommendations for creating custom reservations for your deployment.

#### URLs in Configuration Files (Reporting Services)

The RSReportServer.config file contains multiple entries for URL reservations and the URLs used by Report Manager and report server e-mail delivery. This topic summarizes the URL configuration settings so that you can understand how they compare.

#### URL Reservations for Multi-Instance Report Server Deployments

When you install multiple instances of Reporting Services on a single computer, you increase the probability of encountering URL duplication when a URL is registered. To avoid these errors, follow the recommendations in this topic for creating instance-specific URL reservations.

## □ See Also

#### Tasks

How to: Configure a URL (Reporting Services Configuration)

#### Concepts

Web Service URL (Reporting Services Configuration)

#### **Other Resources**

RSReportServer Configuration File Viewing Reports with SharePoint 2.0 Web Parts Configuring a Report Server for Secure Sockets Layer (SSL) Connections Configuring a Report Server Installation

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help

About URL Reservations and Registration (Reporting Services)

# 

### - -

#### <u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring Report Server URLs</u> >

URLs for Reporting Services applications are defined as URL reservations in HTTP.SYS. A URL reservation defines the syntax of a URL endpoint to a Web application. URL reservations are defined for both the Report Server Web service and Report Manager when you configure the applications on the report server. URL reservations are created for you automatically when configure URLs through Setup or the Reporting Services Configuration tool:

- Setup will create URL reservations using default values. If Setup installs the default configuration, it will reserve two URLs; one of the Report Server Web service and another for Report Manager. You can use the Reporting Services Configuration tool to add more URLs or modify the default URLs that Setup creates.
- The Reporting Services Configuration tool will create a URL reservation based on the URL you specify in the **Web Service URL** or **Report Manager URL** pages in the tool.

Both Setup and the tool will also assign permissions on the URL to the Report Server service, check for duplicate instances, and add the URL reservation to HTTP.SYS. Never create or modify a Reporting Services URL reservation directly using HttpCfg.exe or other tool. If you skip a step or set an invalid value, you will encounter problems that might be difficult to diagnose or fix.

## <sup>I</sup> Note:

HTTP.SYS is an operating system component that listens for network requests and routes them to a request queue. In this release of Reporting Services, HTTP.SYS establishes and maintains the request queue for the Report Server Web service and Report Manager. Internet Information Services (IIS) is no longer used to host or access Reporting Services applications. For more information about HTTP.SYS functionality, see <u>HTTP Server API</u> on MSDN.

# □ In This Topic

This topic introduces the URLs, URL reservations and registration, default values, and deployment-specific recommendations that you should follow if you create or modify URLs for different server configurations. The following sections are in this topic:

- URLs in Reporting Services
- **URL Reservation and Registration**
- Default URLs
- <u>Authentication and Service Identity for Reporting Services URLs</u>
- URLs for Local Administration
- <u>URLs for Scale-out Deployment</u>
- URLs for SharePoint Integrated Mode

For more information about URL construction and usage in a Reporting Services installation or troubleshooting reservation or registration errors, see <u>Configuring</u> <u>Report Server URLs</u> and **Troubleshooting Configuration Errors**.

# □ URLs in Reporting Services

In a Reporting Services installation, you can access the following tools, applications, and items through URLs:

- Report Server Web service
- Report Manager
- Report Builder
- Reports that have been published to a report server

Other published URL-addressable items, such as models and shared data sources, should not be accessed through URLs as stand-alone items. The report server does not display those items in a meaningful format when viewed in a browser window.

## <sup>ℤ</sup>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.

# URL Reservation and Registration

A URL reservation defines the URLs that can be used to access a Reporting Services application. Reporting Services will reserve one or more URLs for the Report Server Web service and Report Manager in HTTP.SYS, and then register them when the service starts. URLs to Report Builder and reports are based on the Report Server Web service URL reservation. By appending parameters to the URL, you can open Report Builder or reports through the Web service. Reservations and registration is provided by HTTP.SYS. For more information, see <u>Namespace Reservations</u>, <u>Registration</u>, and <u>Routing</u> on MSDN.

*URL reservation* is a process by which a URL endpoint to a Web application is created and stored in HTTP.SYS. HTTP.SYS is the common repository of all URL reservations that are defined on a computer and defines a set of common rules that guarantee unique URL reservations.

*URL registration* occurs when the service starts. The request queue is created and HTTP.SYS begins routing requests to that queue. A URL endpoint must be registered before requests that are directed to that endpoint are added to the queue. When the Report Server service starts, it will register all URLs that it has reserved for all enabled applications. This means that the Web service must be enabled in order for registration to occur. If you set the **WebServiceAndHTTPAccessEnabled** property to **False** in the Surface Area Configuration for Reporting Services facet of Policy-Based Management, the URL for the Web service will not register when the service starts.

URLs are unregistered if you stop the service or recycle the Web service or Report Manager application domain. If you modify a URL reservation while the service is running, the report server will recycle the application domain immediately so that the old URL can be unregistered and the new one put into use.

A few simple examples illustrate the concept of a URL reservation and how it relates to URL addresses used for Reporting Services applications. A key point to notice is that the URL reservation has different syntax than the the URL you use to access the application:

URL Reservation in HTTP.SYS	URL	
http://+:80/reportserver		The URL wildcard ( into the re
	http:// <computername>/reportserver</computername>	host that r server con
	http:// <ipaddress>/reportserver</ipaddress>	that with t
	http://localhost/reportserver	number of access the
		This is the
		for a Repc
		server for
		This URL IP address restrictive reservatio the IP add
http://123.45.67.0:80/reportserver	http://123.45.67.0/reportserver	connect to this URL ו
		report serv http:// <coi or http://lc would fail</coi 

# <sup>□</sup> Default URLs

If you install Reporting Services in the default configuration, Setup will reserve URLs for the Report Server Web service and Report Manager. You can also accept these default values when you define URL reservations in the Reporting Services Configuration tool. Default URLs will include an instance name if you install SQL Server Express or if you install Reporting Services as a named instance.

## <sup>I</sup>Important:

The instance character is an underscore character (\_).

URL reservations include a port number. The following operating systems will allow multiple Web applications to share a port: Windows Server 2003, Windows Vista, and Windows Server 2008. If you are using Windows XP, make sure to use unique ports.

Instance Type	Application	Default URL	A
Default instance	Report Server Web service	http:// <servername>/reportserver</servername>	http:// <s< td=""></s<>
Default instance	Report Manager	http:// <servername>/reportserver</servername>	http://<د
Named instance	Report Server Web service	http:// <servername>/reportserver_<instancename></instancename></servername>	http:// <s< td=""></s<>
Named instance	Report Manager	http:// <servername>/reports_<instancename></instancename></servername>	http://<۶
SQL Server Express	Report Server Web service	http:// <servername>/reportserver_SQLExpress</servername>	http:// <s< td=""></s<>
SQL Server Express	Report Manager	http:// <servername>/reports_SQLExpress</servername>	http:// <s< td=""></s<>
# <sup>□</sup> Authentication and Service Identity for Reporting Services URLs

Reporting Services URL reservations specify the service account of the Report Server service. The account under which the service runs is used for all URLs that are created for the Reporting Services applications that run in the same instance. The service identity of the report server instance is stored in the RSReportServer.config file.

The service account has no default value. However, specifying a service account is required during Setup and is specified in **URLReservation** in RSReportServer.config even if you install the server in files-only mode. Valid values for the service account include a domain user account, **LocalSystem**, or **NetworkService**.

Anonymous access is disabled because the default security is **RSWindowsNegotiate**. For intranet access, report server URLs use network computer names. If you want to configure Reporting Services for Internet connections, you must use different settings. For more information, see **Planning for Extranet or Internet Deployment**. For more information about authentication, see **Configuring Authentication in Reporting Services**.

# URLs for Local Administration

You can use http://localhost/reportserver or http://localhost/reports if you specified a strong or weak wildcard for the URL reservation.

The http://localhost URL is interpreted as http://127.0.0.1. If you pegged the URL reservation to a computer name or single IP address, you cannot use localhost unless you create an additional reservation for 127.0.0.1 on the local computer. Similarly, if localhost or 127.0.0.1 is disabled on your computer, you cannot use that URL.

Windows Vista and Windows Server "Longhorn" include new security features to minimize the risk of accidentally running programs with elevated privileges. Additional steps are necessary to enable local administration on these operating systems. For more information, see <u>How to: Configure a Report Server for Local Administration on Windows Vista and Windows Server 2008</u>.

# □ URLs for Scale-out Deployment

Information for this section is not available in this release.

# URLs for Report Server in SharePoint Integrated Mode

If a stand-alone 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:

- URLs for reports and other items are addressed through the SharePoint Web application URL. For URL access to specific reports, always use a fully qualified URL that includes the site path, the document library, the item name, and a file name extension (such as .rdl for a report). You must specify fully qualified URLs when you reference shared data sources and models in reports, and when you specify a target server and folders for publish operations to a report server.
- The file name extension is used to distinguish between different types of report server items. Valid extensions include .rdl for report definitions, .smdl for report models, and .rsds for shared data sources that are created for a SharePoint site.
- Although SharePoint products and technologies have URL reservations defined for them, you can ignore the reservation when publishing to the server. For SharePoint Web applications, URL reservation is an internal operation.
- For single server deployments where an integrated report server and 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. Furthermore, if the report server is integrated with a SharePoint farm, localhost access to a reports server will not resolve for nodes in the deployment that are installed on remote computers.
- The URL reservation and endpoint for Report Manager cannot be configured for a report server that runs in SharePoint integrated mode. 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.

If you integrated a report server scale-out deployment to run within a larger

deployment of a SharePoint product or technology, load balance the report server nodes and define a single virtual server URL to the scale-out deployment. Report Server integration settings only allow you to specify a single report server URL. In the case of a scale-out deployment, the URL must be the access point for the server nodes in the scale-out deployment.

For more information about deployment modes and URL requirements for publishing items, see **Planning a Deployment Mode** and **Deploying Models and Shared Data Sources to a SharePoint Site**.

## □ See Also

#### Tasks

How to: Configure a URL (Reporting Services Configuration)

#### Concepts

<u>URL Reservation Syntax (Reporting Services)</u> <u>Default Configuration for a Native Mode Installation (Reporting Services)</u>

#### **Other Resources**

RSReportServer Configuration File Deploying Reporting Services and Internet Information Services Side-by-Side

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help URL Reservation Syntax (Reporting Services)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring Report Server URLs</u> >

This topic describes the parts of the URL string for the Report Server Web service and Report Manager. The URL string that is stored internally has a different structure from a URL that you type in the Address bar of a browser window. The URL reservation string appears in the Results window of the Reporting Services Configuration tool when you configure a URL and in the RSReportServer.config file. Knowing how the URL string is defined can be useful if you are troubleshooting URL reservation problems or querying HTTP.SYS to view the internal URL reservations that are defined on your server.

# <sup>□</sup> URL Syntax

A report server URL is stored in the **UrlString** element and the **VirtualDirectory** element. The reason for separating **UrlString** and **VirtualDirectory** into separate elements is that you can have multiple URL strings, but only one virtual directory name for each Reporting Services application.

In HTTP.SYS, the URL reservation includes both the **UrlString** and **VirtualDirectory**. The syntax for a URL reservation has the following parts:

<scheme>://<hostname>:<port>/<virtualdirectory>

The following table describes each property and which values are valid for each one.

Property	Valid values	Description
Scheme	http or https	Prefixes for non-SSL and SSL connections.
		Identifies the server on the network. (+) Strong wildcard is the default. HTTP.SYS will accept all requests on all network adaptors for a given port and
		The report server will accept any request on the port.
	<ul> <li>(+) Strong wildcard, equates to (All Assigned) value for the IP address.</li> <li>(*) Weak wildcard, equates to a lD address of (All address)</li> </ul>	(*) Weak wildcard. HTTP.SYS accepts all requests not handled by other URL reservations on all network adaptors for a given port and virtual directory combination.
	Unassigned).	

Hostname	Fully qualified domain name Machine name	Machine name is the NETBIOS name of the computer on the network.
	IP address (IPV4) IP address (IPV6)	Fully qualified domain name includes domain address and server name, as registered with a domain controller or public domain name server.
		IP address (IPV4) is the IP address of a network adaptor on the computer in IPV4 format: <i>nnn.nnn.nnn.nn</i> .
		IP address (IPV6) is the IP address of a network adaptor on the computer in IPV6 format: <header>:</header>
	80	Port 80 is the standard port for HTTP requests to and from a server.
Port	443	Port 443 is the standard report for SSL connections.
	<custom></custom>	You can use any port that is not already reserved by another application.
		Specifies the name of the application. This value is a string. By default, Reporting Services uses ReportServer and Reports as the application names for the Report Server Web service and Report Manager applications. You can use different names if you prefer.

		This value is required. It identifies the application.
Virtualdirectory	ReportServer[_InstanceName] Reports[_InstanceName] <custom></custom>	Specify only one virtual directory for each application instance. To create multiple URLs for the same application in the same instance, create multiple versions of the <b>UrlString</b> . To create unique virtual directory names for multiple application instances, consider including the instance name in the virtual directory name, using the underscore character () to append the instance name. <i>InstanceName</i> is optional, but recommended if you have multiple instances on the same computer. For more information about how to set URL reservations for named instances, see <u>URL Reservations</u> for <u>Multi-Instance Report Server</u> Deployments.
		not case-sensitive. You can use any string as long as it does not include URL separator characters or URL encoding.

## $\Box$ See Also

## Tasks

How to: Configure a URL (Reporting Services Configuration)

Concepts Configuring Report Server URLs

# Other Resources

**RSReportServer Configuration File** 

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help URLs in Configuration Files (Reporting Services)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring Report Server URLs</u> >

Reporting Services stores application settings in a RSReportServer.config file. Within this file, there are configuration settings for both URLs and URL reservations. These configuration settings have very different purposes and rules for modification. If you are accustomed to modifying configuration files to tune a deployment, this topic can help you understand how each URL setting is used.

# <sup>D</sup> URL Settings in RSReportServer.config File

Reporting Services stores URLs for application and report access, and to connect Web front-end components to a back-end report server.

## **URLs for Application Access**

URLs are used to access the Report Server Web service and Report Manager. To configure the URLs, you must use the Reporting Services Configuration tool. The tool creates the URL reservations for each application in HTTP.SYS and adds entries for the URLs in the **URLReservations** section of RSReportServer.config.

- To view descriptions of each element in the **URLReservations** section, see **RSReportServer Configuration File**.
- For more information about the syntax of just the **UrlString** element, see <u>URL Reservation Syntax (Reporting Services</u>).
- For instructions on how to configure URLs for application access, see <u>How</u> to: <u>Configure a URL (Reporting Services Configuration)</u>.

## **URLs for Report Access**

Reporting Services includes a report server e-mail delivery extension that you can use to send report links or attachments. A report link is constructed when the report is delivered. The report server e-mail delivery extension uses the **UrlRoot** setting in the configuration file to create the link. **UrlRoot** is also used to resolve links in a rendered report that is generated through unattended report processing.

**UrlRoot** is specified automatically in the RSReportServer.config file when you configure URLs for application access. If you modify this value in the configuration file, you must specify a valid URL address to a Report Server Web service that is connected to a report server database that contains the reports you want to deliver. You can only specify one **UrlRoot** for a single report server instance; only one **UrlRoot** entry can exist in the RSReportServer.config file for any given report server instance. If you have multiple URLs reserved for the Report Server Web service, you must choose one of the available values for **UrlRoot**.

In most cases, you do not need to modify **UrlRoot**. However, if the report server will be accessed through a fully qualified URL, and you did not configure a URL that uses a host header to the fully qualified site name, you must edit the RSReportServer.config manually to set the **UrlRoot** to the fully qualified report server URL that will be used to render the report (for example, https://www.adventure-works.com/mywebapp/reportserver).

## URLs Connecting Report Manager and Web Parts to the Report Server Web Service

Report Manager and the SharePoint 2.0 Web Parts for Reporting Services are Web front-end components that connect to a report server. URLs used to connect to a backend report server include the following:

- **ReportServerUrl** (used by Report Manager)
- **ReportServerExternalUrl** (used by Web Parts)

## <sup>I</sup>Note:

Previous versions of Reporting Services included the

**ReportServerVirtualDirectory** element. This value is obsolete in SQL Server 2008. If you upgraded an existing installation and are using a configuration file that contains this setting, the report server no longer reads this value.

The following table provides a summary of all the URLs that can be specified in a Reporting Services configuration file.

Setting	Usage

ReportServerUrl	<ul> <li>Optional. This element is not included in the RSRepo yourself.</li> <li>Set this element only if you are configuring one of the</li> <li>Report Manager provides Web front-end access on a different computer or a different instance or</li> <li>When you have multiple URLs to a report server specific URL.</li> <li>You have a specific report server URL through v connections to use. For example, you might enable Report Manager require that Report Manager connect to the report this case, you might configure <b>ReportServerUr</b></li> <li>For instructions on how to implement these scenarios Manager.</li> </ul>
ReportServerExternalUr	Optional. This element is not included in the RSRepo yourself. Set this element only if you are using the SharePoint able to retrieve a report and open it in a new browser Add <b><reportserverexternalurl< b="">&gt; underneath the <b><i< b=""></i<></b></reportserverexternalurl<></b>

The following example illustrates the syntax:
<reportserverexternalurl>http://myserver/repo</reportserverexternalurl>

## □ See Also

#### Tasks

How to: Configure a URL (Reporting Services Configuration)

Concepts Configuring Report Server URLs

#### **Other Resources**

**RSReportServer Configuration File** How to: Modify a Reporting Services Configuration File Viewing Reports with SharePoint 2.0 Web Parts

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help URL Reservations for Multi-Instance Report Server Deployments

# 

#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring Report Server URLs</u> >

If you install multiple instances of Reporting Services on the same computer, you must consider how you will define the URL reservations for each instance. Within each instance, the Report Server Web service and Report Manager must have at least one URL reservation each. The entire set of reservations must be unique in HTTP.SYS.

Duplicate URLs are detected during URL registration, which occurs when the service starts. If you create URL reservations that are not unique, the name conflict might not be detected until you start the service. For this reason, make sure that you follow naming conventions or rules to ensure all values are unique.

# Default Naming Conventions

Reporting Services can be installed within a SQL Server named instance. When you install or configure a report server within a named instance, the instance name is automatically included in the virtual directory in the default URL reservation that Reporting Services provides. The following table shows the URL reservations for a default instance and a named instance.

SQL Server Instance	Default URL Reservation
Default (MSSQLServer)	http://+:80/reportserver
Named (MynamedInstance)	http://+:80/reportserver_MyNamedInstance

For the named instance, the virtual directory includes the instance name. Both the default instance and the named instance listen on the same port, but the unique virtual directory names determine which report server gets the request.

Best practice recommendations are to use the virtual directory name to distinguish among the report server instance. It provides a clear correspondence between a URL and the target instance, and ensures that the application names are unique across the whole system.

# <sup>©</sup> Custom Naming Conventions

Although using the instance name is recommended, you can use the URL syntax and your own naming conventions to meet the unique name constraints for URL reservations. The following examples illustrate different approaches for creating unique URLs for each instance.

Report Server default instance (MSSQLSERVER)	ReportServer_MyNamedInstance	Uniqu
http://+:80/reportserver	http://+:8888/reportserver	Each instanc listens differei port.
http://www.contoso.com/reportserver	http://SRVR-46/reportserver	Each instanc respond differen server names (fully qualifie domair name, a machin name).

# <sup>□</sup> Uniqueness Requirements

The underlying technologies used by Reporting Services impose requirements around unique names. HTTP.SYS requires that all URLs within its repository be unique. You can vary the port, host name, or virtual directory name to create a unique URL. ASP.NET requires that application identities be unique within the same process. This requirement affects the virtual directory names. It specifies that you cannot duplicate a virtual directory name within the same report server instance.

## □ See Also

#### Tasks

How to: Configure a URL (Reporting Services Configuration)

## Concepts

<u>Configuring Report Server URLs</u> <u>Working with Multiple Versions and Instances of SQL Server</u>

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Creating a Report Server Database



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

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. With a default SQL Server instance, the databases are named **reportserver** and **reportservertempdb**. Collectively, the two databases are referred to as the "report server database" or "report server catalog".

## <sup>I</sup> 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.

# <sup>□</sup> 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 native mode default configuration** option or the **Install the SharePoint integrated mode default configuration** 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. You must create the report server database manually if you are using a remote SQL Server Database Engine to host the database. For more information, see <u>How to: Create a Report Server Database (Reporting Services Configuration)</u>.

# **Database Server Version Requirements**

SQL Server is used to host the report server databases:

- SQL Server 2005 or SQL Server 2008 can be used to host the databases. Do not use SQL Server 2000 or earlier versions of SQL Server.
- The SQL Server Database Engine instance can be a local or remote 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 <u>Configuring a Report Server</u> <u>Database Connection</u>.

## <sup>III</sup> 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.

# <sup>D</sup> Database Server Edition Requirements

When creating a report server database, be aware that not all editions of SQL Server 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	Use this edition of the Database Engine
Reporting Services	instance to host the database
SQL Server Express with	SQL Server Express with Advanced Services
Advanced Services	(local only).
Web	Web edition (local only)
Workgroup	Workgroup edition (local only)
Evaluation	Standard, Evaluation, Enterprise editions
	(local or remote)
Developer	Standard, Developer, Enterprise editions
	(local or remote)
Standard	Standard, Enterprise editions (local or remote)
Enterprise	Standard, Enterprise editions (local or remote)

## <sup>□</sup> See Also

#### Tasks

How to: Create a Report Server Database (Reporting Services Configuration)

#### **Other Resources**

**Report Server Database Administering a Report Server Database Developer's Guide (Reporting Services)** 

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Configuring a Report Server Database Connection



#### -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

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 <u>How to: Create a Report Server Database (Reporting Services Configuration</u>).

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 the service account for the connection (which uses 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 <u>Configuring the Report Server Service Account</u>.
- Configuring a report server scale-out deployment. Configuring a scaleout deployment requires that you create multiple connections to a report server database. For more information about how to perform this multistep operation, see <u>How to: Configure a Report Server Scale-Out</u> <u>Deployment (Reporting Services Configuration)</u>.

# How Reporting Services Connects to the Database Engine

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 service identity. For more information about working with encryption keys, see <u>Configuring and Managing Encryption Keys</u>.

The report server database is an internal component, accessed only by the report server. The credentials and connection information you specify for the report server database are used exclusively by the report server. Users who request reports do not require databases permissions or a database login for the report server database.

Reporting Services uses **System.Data.SqlClient** to connect to the Database Engine that hosts the report server database. If you are using a local instance of the Database Engine, the report server will establish the connection using shared memory. If you are using a remote database server for the report server database, you might have to enable remote connections depending on the edition you are using. If you are using the Developer edition, remote connections are turned off by default. If you are using the Enterprise edition, remote connections are enabled for TCP/IP by default.

To verify that the instance accepts remote connections, click **Start**, click **All Programs**, click **Microsoft SQL Server 2008**, click **Configuration Tools**, click **SQL Server Configuration Manager**, and then verify that the TCP/IP protocol is enabled for each service.

When you enable remote connections, the client and server protocols will also be enabled. To verify the protocols are enabled, click **Start**, click **All Programs**, click **Microsoft SQL Server 2008**, click **Configuration Tools**, click **SQL Server Configuration Manager**, click **SQL Server Network Configuration**, and then click **Protocols for MSSQLSERVER**. For more information, see **How**  to: Enable or Disable a Server Network Protocol (SQL Server Configuration Manager).

# <sup>Defining</sup> 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 Database Engine instance hosting the report server database. You can use a local or remote SQL Server 2005 or SQL Server 2008 instance.
- Name of the report server database. When creating a connection for the first time, you can create a new report server database or select an existing database. For more information, see <u>Creating a Report Server Database</u>.
- 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:

Copy Code

```
<Dsn></Dsn>
<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 service account. Because the report server is implemented as a single service, only the account under which the service runs requires database access.
- 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.

#### <sup>I</sup>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 do not affect connections to the report server database or to external data sources that provide content to reports.

If the instance of the Database Engine is configured for Windows authentication and is in the same domain or a trusted domain with the report server computer, you can configure the connection to use the service account or a domain user account that you manage as a connection property through the Reporting Services Configuration tool. If the database server is in a different domain or if you are using workgroup security, you must configure the connection to use a SQL Server database login. In this case, be sure to encrypt the connection. For more information, see **Encrypting Connections to SQL Server**.

## **Using Service Accounts and Integrated Security**

You can use Windows integrated security to connect through the Report Server

service account. The account is 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.

The service account is a trusted account that provides a low-maintenance approach to managing a report server database connection. Because the service account uses Windows integrated security to make the connection, the credentials do not have to be stored. However, if you subsequently change the service account password or identity (for example, switching from a built-in account to a domain account), be sure to use the Reporting Services Configuration tool to make the change. The tool automatically updates the database permissions to use the revised account information. For more information, see <u>Configuring the Report Server Service Account</u>.

If you configure the database connection to use the service account, the account 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 a Windows user account for the report server connection to the report server database. 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 help secure the transmission of data between the servers. 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 determine 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 the Report Server Databases to Another Computer** and **Migration (Reporting Services)**.

# <sup>□</sup> Configuring Multiple Reports Servers to Use the Same Report Server Database

You can configure multiple report servers to use the same report server database. This deployment configuration is called a scale-out deployment. This configuration is a prerequisite if you want to run multiple report servers in a server cluster. However, you can also use this configuration if you want to segment service applications or if you want to test the installation and settings of a new report server instance to compare it with an existing report server installation. For more information, see <u>Configuring Reporting Services for Scale-Out Deployment</u>.
## □ See Also

#### Concepts

<u>Creating a Report Server Database</u> <u>Reporting Services Configuration How-to Topics</u> <u>Configuring the Report Server Service Account</u>

#### **Other Resources**

Connecting to a Data Source (Reporting Services) Configuring a Report Server Installation RSReportServer Configuration File rsconfig Utility

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Configuring a Report Server for E-Mail Delivery



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

Reporting Services includes an e-mail delivery extension so that you can distribute reports through e-mail. Depending on how you define the e-mail subscription, a delivery might consist of a notification, link, attachment, or embedded report. The e-mail delivery extension works with your existing mail server technology. The mail server must be an SMTP server or forwarder. The report server connects to an SMTP server through Collaboration Data Objects (CDO) libraries (cdosys.dll) that are provided by the operating system.

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**.

## <sup>©</sup> Configuration Requirements

- Report server e-mail delivery is implemented on Collaboration Data Objects (CDO) and requires a local or remote Simple Mail Transfer Protocol (SMTP) server or SMTP forwarder. SMTP is not supported on all Windows operating systems. If you are using the Itanium-based edition of Windows Server 2008, SMTP is not supported. For more information about configuration options provided through CDO, see <u>Configuration CoClass</u> on MSDN.
- The Report Server service account must have permission on the SMTP server to send mail.
- The 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.

## <sup>I</sup>∕ Note:

The default e-mail delivery extension does not provide support for digitally signing or encrypting outgoing mail messages.

# <sup>□</sup> 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.

## <sup>©</sup> 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. For more information, see <u>How to:</u> <u>Configure a Report Server for E-mail Delivery (Reporting Services Configuration)</u>.
- (Optionally) Use a text editor to specify additional settings in the RSreportserver.config file. This file contains 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 **How to: Modify a Reporting Services Configuration File** in SQL Server Books Online.

#### <sup>I</sup>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.

## <sup>B</sup> 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>
     <prom>my-rs-email-account@Adventure-Works.com</prom>
     <EmbeddedRenderFormats>
          <RenderingExtension>MHTML</RenderingExtension>
     </EmbeddedRenderFormats>
     <PrivilegedUserRenderFormats></PrivilegedUserRenderFormats>
     <ExcludedRenderFormats>
          <RenderingExtension>HTMLOWC</RenderingExtension>
          <RenderingExtension>NULL</RenderingExtension>
     </ExcludedRenderFormats>
     <SendEmailToUserAlias>True</SendEmailToUserAlias>
     <DefaultHostName></DefaultHostName>
     <PermittedHosts>
          <HostName>Adventure-Works.com</HostName>
          <HostName>hotmail.com</HostName>
     </PermittedHosts>
</RSEmailDPConfiguration>
```

# <sup>©</sup> 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 workaround 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.

## <sup>©</sup> 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 these values 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**.

## <sup>©</sup> 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 <u>How to: Configure a Report Server for E-mail Delivery (Reporting Services Configuration)</u>.

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.

## □ See Also

#### Tasks

How to: Start Reporting Services Configuration How to: Configure a Report Server for E-mail Delivery (Reporting Services Configuration)

#### **Other Resources**

Subscription and Delivery (Reporting Services) E-Mail Delivery in Reporting Services Configuration Files (Reporting Services)

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Configuring the Unattended Execution Account



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

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, or connect to external report data sources that do not require or use authentication. For more information, see Specifying Credential and Connection Information for Report Data Sources 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 Images to a Report** in SQL Server Books Online.

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 service account are never used to connect to other computers.

## <sup>I</sup> 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.

## <sup>□</sup> How to Configure the Account

You must use a domain user account. To serve its intended purpose, this account should be different than the one used to run the Report Server service. Be sure to use an account that has minimum permissions (read-only access with network connection permissions is sufficient) and limited access to just those computers that provide data sources and resources to the report server. For more information, see <u>How to: Start Reporting Services Configuration</u>.

To specify 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.

- 1. Start the Reporting Services Configuration tool and connect to the report server instance you want to configure. For instructions, see <u>How to: Start</u> <u>Reporting Services Configuration</u>.
- 2. On the Execution Account page, select **Specify an execution account**.
- 3. Type the account and password, retype the password, and then click **Apply**.

## **Using RSCONFIG Utility**

Another way to set the account is to 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.

#### **How Account Information is Stored**

When you set the account, the following settings are specified as encrypted values in the RSreportserver.config file on a local or remote report server instance:

Copy Code

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 Use the Unattended Report Processing Account

To retrieve image files, the report server uses the account automatically and no specific action is required on your part. To use the account to connect to external data sources that provide data to reports, you must specify a **Credential Type** option in the data source properties page of the report data source or shared data source:

- In Report Manager or Management Studio, select the **Credentials are not required** option.
- On a SharePoint site for a report server that runs in SharePoint integrated mode, select the **Unattended report processing account** option.

The unattended report processing account is used primarily to connect to external servers, and not as a login to database servers. If you want to use the account credentials to log in to a database, you must specify credentials in the connection string. You can specify **Integrated Security=SSPI** if the database server supports Windows integrated security and the account used for unattended report processing has permission to read the database. Otherwise, you must enter the user name and password in the connection string, where it appears in clear text to any user who has permission to edit data source connection properties.

Although you are not prevented from using the unattended report processing account to retrieve data after the connection is made, doing so is not recommended. The account is supposed to be used for very specific functions. If you use it to retrieve data, you undermine the purpose for which it is intended.

# <sup>□</sup> How to Maintain 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**.

## □ How to Delete 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.

## □ See Also

**Other Resources** 

Planning for Services, Accounts, and Connections Configuring a Report Server Installation Data Sources Properties Page (Report Manager) RSReportServer Configuration File rsconfig Utility Specifying Credential and Connection Information for Report Data Sources

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Configuring and Managing Encryption Keys



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

Reporting Services uses encryption keys to secure credentials and connection information that is stored in a report server database. In Reporting Services, encryption is supported through 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 of the symmetric key, 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.

## <sup>□</sup> In This Section

Торіс	Description
<u>Initializing a Report Server</u>	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	Explains how to add and remove encryption keys to
Encryption Keys for Scale-	control which report servers are part of a scale-out
<u>Out Deployment</u>	deployment.

## □ See Also

Concepts
<u>Storing Encrypted Report Server Data</u>

**Other Resources** Reporting Services Configuration Tool rskeymgmt Utility Administering a Report Server Database

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help Initializing a Report Server



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring and Managing Encryption</u> <u>Keys</u> >

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 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 <u>Configuring and Managing Encryption Keys</u> and <u>Storing Encrypted Report</u> <u>Server Data</u>.

Encryption keys are based partly on the profile information of the Report Server service. If you change the user identity used to run the Report Server 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**.

## <sup>□</sup> 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 service to encrypt and decrypt data. The symmetric key is itself encrypted with an asymmetric key.

The following steps describe the initialization process:

- 1. At initial start up, the Report Server service reads the RSReportServer.config file to get the installation identifier and database connection information.
- 2. The Report Server service requests a public key from Cryptographic Services. Windows creates a private and public key and sends the public key to the Report Server service.
- 3. The Report Server service connects to the report server database and stores the installation identifier and public key values.
- 4. The Report Server service calls into Cryptographic Services again, this time to request a symmetric key. Windows creates the symmetric key.
- 5. The Report Server 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 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 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.

## <sup>□</sup> 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 <u>Configuring a Report Server</u> <u>Database Connection</u>.
- 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 <u>How to:</u> <u>Configure a Report Server Scale-Out Deployment (Reporting Services Configuration)</u>.

### <sup>I</sup>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 <u>Configuring and Managing Encryption Keys</u>.

## <sup>I</sup>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**.

## <sup>□</sup> 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

#### Concepts

Storing Encrypted Report Server Data

**Other Resources** 

Backup and Restore Operations for a Reporting Services Installation Troubleshooting Initialization and Encryption Key Errors Deployment (Reporting Services) Administration (Reporting Services)

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Backing Up and Restoring Encryption Keys



#### - ▼

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring and Managing Encryption</u> <u>Keys</u> >

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.

#### <sup>⊿</sup>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.

## <sup>□</sup> 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.

## <sup>□</sup> 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:

rskeymgmt -a -f a:\rsdbkey.snk -p<password>

## <sup>□</sup> See Also

**Concepts** <u>Configuring and Managing Encryption Keys</u>

Other Resources Reporting Services Configuration Tool rskeymgmt Utility Backup and Restore Operations for a Reporting Services Installation

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help Deleting and Re-creating Encryption Keys



#### - ▼

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring and Managing Encryption</u> <u>Keys</u> >

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.

## <sup>□</sup> 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 <u>Initializing a Report Server</u>.

#### How to re-create encryption keys (Reporting Services Configuration Tool)

- Disable the Report Server Web service and HTTP access by modifying the IsWebServiceEnabled property in the rsreportserver.config file. This step temporarily stops authentication requests from being sent to the report server without completely shutting down the server. You must have minimal service so that you can recreate the keys. If you are recreating encryption keys for a report server scale-out deployment, disable this property on all instances in the deployment.
  - Open Windows Explorer and navigate to *drive*:\Program Files\Microsoft SQL Server\*report\_server\_instance*\Reporting Services. Replace *drive* with your drive letter and *report\_server\_instance* with the folder name that corresponds to the report server instance for which you want to disable the Web service and HTTP access. For example, C:\Program Files\Microsoft SQL Server\MSRS10.MSSQLSERVER\Reporting Services.
- 2. Open the rsreportserver.config file.
- 3. For the **IsWebServiceEnabled** property, specify **False**, and then save your changes.
- 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. Re-enable the Web service and HTTP access by modifying the **IsWebServiceEnabled** property in the rsreportserver.config file. Do this for all instances if you are working with a scale out deployment.

## How to re-create encryption keys (rskeymgmt)

- 1. Disable the Report Server 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 respecify 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 (rskeymmgt)

1. Run **rskeymgmt.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:

rskeymgmt -d

Copy Code

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 for Report Data Sources**.
- 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

Configuring and Managing Encryption Keys Storing Encrypted Report Server Data

#### **Other Resources**

**Reporting Services Configuration Tool rskeymgmt Utility** 

#### SQL Server Setup Help

## Adding and Removing Encryption Keys for Scale-Out Deployment

## - ▼

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring and Managing Encryption</u> <u>Keys</u> >

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 <u>Configuring Reporting Services for Scale-Out Deployment and How to:</u> <u>Configure a Report Server Scale-Out Deployment (Reporting Services Configuration)</u>.

## 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)

 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>
```

These steps remove the report server from a scale-out deployment, but they do not uninstall the Reporting Services instance on the report server. After you remove the report server from the scale-out deployment, you can uninstall Reporting Services from the server if you no longer need Reporting Services on that server. For information, see <u>How to: Uninstall an Existing Instance of SQL Server 2008 (Setup)</u>.

## $\Box$ See Also

## Concepts

Configuring and Managing Encryption Keys Initializing a Report Server

#### **Other Resources**

**Reporting Services Configuration Tool rskeymgmt Utility** 

## SQL Server Setup Help Storing Encrypted Report Server Data



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Configuring and Managing Encryption</u> <u>Keys</u> >

Reporting Services stores encrypted values in the report server database and in configuration files. Most encrypted values are credentials that are used for accessing external data sources that provide data to reports. This topic describes which values are encrypted, the encryption functionality used in Reporting Services, and other kinds of stored confidential data that you should know about.

## <sup>□</sup> Encrypted Values

The following list describes the values that are stored in a Reporting Services installation.

- Connection information and credentials used by a report server to connect to a report server database that stores internal server data. These values are specified and encrypted during setup or report server configuration. You can update the connection information at any time using the Reporting Services Configuration tool or the **rsconfig** utility. Encryption of configuration settings is performed by using the machinelevel key of the local computer that is available to all users. Encrypted report server connection information is stored in the rsreportserver.config file (no other configuration file contains encrypted settings). For more information, see <u>Configuring a Report Server Database Connection</u>.
- Stored credentials that are used by a report server to connect to external data sources that provide data to a report.
   These values are defined when you configure data source information for a report, and then stored as encrypted values in a report server database. The report server uses a symmetric key to encrypt and decrypt this data. For more information about stored credentials, see Planning for Services, Accounts, and Connections and Specifying Credential and Connection Information for Report Data Sources.
- An unattended user account used by the report server to connect to other computers to retrieve external images files or external data that is used in a report.

This account is used when a connection to a remote computer is required and no other credentials are available to make the connection. This account is primarily used to support unattended report processing for reports that do not use credentials to access a data source. If you create reports based on data sources that do not require or use credentials when accessing data, you must configure this account for the report server to use.

This account is required under certain circumstances and can only be created through the Reporting Services Configuration tool or **rsconfig**. This value is also stored in the rsreportserver.config file. You must create this account manually. For more information about this account and how it is used, see <u>Configuring the Unattended Execution Account</u>.

• The symmetric key used for encryption.

This value is created during setup or server configuration, and then stored as an encrypted value in the report server database. The Report Server Windows service uses this key to encrypt and decrypt data that is stored in the report server database.

## <sup>B</sup> Encryption Functionality in Reporting Services

Reporting Services uses cryptographic functions that are part of the Windows operating system. Both symmetric and asymmetric encryption are used.

Data in the report server database is encrypted using a symmetric key. There is a single symmetric key for each report server database. This symmetric key is itself encrypted using the public key of an asymmetric key pair generated by Windows. The private key is held by the Report Server Windows service account.

In a report server scale-out deployment where multiple report server instances share the same report server database, a single symmetric key is used by all report server nodes. Each node must have a copy of the shared symmetric key. A copy of the symmetric key is created for each node automatically when the scale-out deployment is configured. Each node encrypts its copy of the symmetric key using the public key of a key pair specific to its Windows service account. To learn more about how the symmetric key is created for both single instance and scale-out deployments, see <u>Initializing a Report Server</u>.

## <sup>I</sup>Note:

When you change the Report Server Windows service account, the asymmetric keys can become invalid, which will disrupt server operations. To avoid this problem, always use the Reporting Services Configuration tool to modify service account settings. When you use the configuration tool, the keys are updated for you automatically. For more information, see <u>Configuring the Report Server Service Account</u>.

## <sup>□</sup> Other Sources of Confidential Data

A report server stores other data that is not encrypted, yet may contain sensitive information that you want to protect. Specifically, report history snapshots and report execution snapshots contain query results that may include data that is intended for authorized users. If you are using snapshot functionality for reports that contain confidential data, be aware that users who can open tables in a report server database may be able to view portions of a stored report by inspecting the contents of the table.

## <sup>I</sup>∕ Note:

Reporting Services does not support caching or report history for reports that use parameters based on the security identify of the user.

## <sup>□</sup> See Also

## Concepts

**Configuring and Managing Encryption Keys** 

## **Other Resources**

Administering a Report Server Database Configuration Files (Reporting Services)

SQL Server Setup Help Configuring Reporting Services for Scale-Out Deployment

## See Also

#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

Scale-out deployments are used to increase scalability of report servers to handle more concurrent users and larger report execution loads. It can also be used to dedicate specific servers to process interactive or scheduled reports. Scale-out deployments consist of:

- Two or more report server instances sharing a single report server database.
- Optionally, a network load-balanced (NLB) cluster to spread interactive user load across the report server instances.

When deploying Reporting Services on an NLB cluster, you need to ensure the NLB virtual server name is used in the configuration of report server URLs and that servers are configured to share the same view state.

Reporting Services does not participate in Microsoft Cluster Services clusters. However, you can create the report server database on a Database Engine instance that is part of a failover cluster.

To plan, install, and configure a scale-out deployment, follow these steps:

- Review **Planning for Scale-Out Deployment** for guidelines on when to use a scale-out deployment.
- Review <u>How to: Install SQL Server 2008 (Setup</u>) for instructions on how to install report server instances.
- Review <u>How to: Configure a Report Server Scale-Out Deployment</u> (<u>Reporting Services Configuration</u>) for instructions on how to share a report server database and join report servers to a scale-out.
- Optionally, if you are configuring the scale-out to run on a network loadbalanced cluster, review <u>How to: Configure a Report Server on a</u> <u>Network Load Balancing Cluster</u> for instructions on how to configure view state validation.

## □ See Also

## Concepts

<u>URLs in Configuration Files (Reporting Services)</u> <u>Considerations for Installing Reporting Services</u>

#### **Other Resources**

Planning a Deployment Topology Hosting a Report Server Database in a SQL Server Failover Cluster Configuration Files (Reporting Services) Reporting Services Components and Tools

## SQL Server Setup Help Configuring Reporting Services for SharePoint 3.0 Integration



## - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

You can configure a deployment of SQL Server Reporting Services 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. It contains the following sections:

- Overview of Configuration Steps on a Report Server
- Overview of Configuration Steps on a SharePoint Server
- Service Account Configuration
- How to Configure Server Integration

For more information about different server topologies, see **Deployment Topologies for Reporting Services in SharePoint Integrated Mode**. For a comprehensive list of topics about SharePoint integration in Books Online, see **How Do I Learn About SharePoint Integrated Mode**.

## <sup>©</sup> Overview of Configuration Steps on a Report Server

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:

- Create a report server database in SharePoint integrated mode. This 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.
- 2. For deployments that include a SharePoint server farm, install a Windows SharePoint Services Web front-end on the report server computer. The Web front-end joins the report server computer to a SharePoint farm. Installing the Web front-end on the report server computer is only necessary if you deploy the report server and the SharePoint technology instance on separate computers.
- 3. For deployments that include placing the SharePoint databases on the same computer as the report server, configure the Report Server service to run under a domain account. This step is only necessary if the report server and application databases are one computer, and the SharePoint Web application is on another computer.

## **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.

## About Report Server Scale-out Deployment and SharePoint Integrated Mode

If you are integrating a report server scale-out deployment with a SharePoint server farm, 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. Furthermore, you must create a single point of entry to the scale-out deployment (that is, a URL that resolves to a virtual IP for an NLB cluster on which the report server instances are installed).

## About Multiple Report Server Instances and SharePoint Integrated Mode

If you are installing multiple instances of Reporting Services on a single computer, but only configuring one of those instances for SharePoint integrated mode, be sure to do the following:

• Ensure that all instances of Reporting Services are the same version. SQL Server 2008 Reporting Services (SSRS) introduces an updated version of the Reporting Services Windows Management Instrumentation (WMI) provider. You cannot have SQL Server 2008 Reporting Services (SSRS) and pre-SQL Server 2008 Reporting Services (SSRS) versions on the same computer. Therefore, it is necessary that all report server instances run SQL Server 2008 Reporting Services (SSRS) or later, even if you do not require SQL Server 2008 Reporting Services (SSRS) features for all of the instances.

• If you run SQL Server 2008 Setup for each report server instance at different times, restart the WMI service after you install the second instance of SQL Server 2008 Reporting Services (SSRS).

# <sup>□</sup> Overview of Configuration Steps on a SharePoint Server

To configure integration on the server running the SharePoint product or technology, you must do the following:

1. 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.

- 2. Activate the report server feature if you did not install the add-in on a root site collection.
- 3. Use SharePoint Central Administration to configure report server integration. Configuration settings include specifying a single report server URL, setting an authentication type on requests directed to the report server, granting database access permissions to the Report Server service account.
- 4. Optionally, in SharePoint Central Administration, you can set report server defaults that enable logging, ad hoc reporting, and limits for report history and time-out values.

SharePoint products and technologies can be deployed as a farm or as a standalone 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.

## Service Account Configuration

A report server is implemented as a single Windows service that runs under a built-in account or a local or domain Windows user account. In SharePoint integrated mode, the service must connect to the SharePoint content databases with write and execute permissions. Because the service connects to the SharePoint content databases, the account that you use to run the service will have different requirements depending on whether remote connections are used and whether the report server URL proxy endpoint connection uses Windows integrated security or Trusted Account mode.

The following table provides service account recommendations for different deployment scenarios.

Deployment scenario	Sorvicos	SharePoint	Account recommendation for Report
	Services	databases	Server service
Single server	All services are installed on the same computer	On same computer	Any (NetworkService, domain user, local user, LocalSystem). Because the applications connect using a local database connection, you can configure the service to run under any account type.
Distributed deployment on multiple computers	SharePoint Web application and the report server installed on separate computers	On a third computer Or SharePoint server computer	Domain user account Or NetworkService, but only if you add NT_AUTHORITY\NetworkService to the WSS_WPG Windows group on the report server computer. For more information, see <b>Troubleshooting Configuration Errors</b> .
			Domain user account is required, regardless of how the report server URL proxy endpoint is configured.

Distributed deployment on multiple computers	SharePoint Web application and the report server installed on separate computers	On report server computer	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 service runs under a built-in account, it will be denied access accordingly. Configuring the service to run as domain user account ensures that the report server will not be denied database access because it is running under a prohibited account.
---	---	---------------------------------	---

<sup>
</</sup>
Note:

The report server database is not included in the discussion of service account requirements for database connections. This is because only the report server connects to the report server database.

## **About SharePoint Service Account Configurations**

Service account recommendations for Windows SharePoint Services and Office SharePoint Server state that for a server farm deployment you should run the application pool process as a domain user account, and not as Network Service. Although a domain user account is recommended, it is not required. However, it is important to realize that if you configure a SharePoint Web application to run as Network Service, and the report server is on a remote computer, you will encounter errors when accessing reports and report server features from a SharePoint site if the report server URL proxy endpoint connects in Trusted Account mode. For more information about server errors that occur when you deploy this configuration, see **Troubleshooting Configuration Errors**.

For more information about the service account connection to a SharePoint technology instance, see **Security Overview for Reporting Services in SharePoint Integrated Mode**. To read about the service account recommendations for a SharePoint deployment, see <u>Plan for administrative and</u> <u>service accounts</u> in the Windows SharePoint Services product documentation.

## <sup>□</sup> How to Configure Server Integration

Configuring the servers for integrated operations requires multiple tools and steps. Click the following links to read the instructions for configuring server integration in specific deployment scenarios:

- How to: Configure SharePoint Integration on a Stand-alone Server
- <u>How to: Configure SharePoint Integration on Multiple Servers</u>

## □ See Also

#### Tasks

How to: Configure Report Server Integration in SharePoint Central Administration How to: Switch Server Modes (Reporting Services Configuration)

#### Reference

Setting Up Windows Service Accounts

## Concepts

Backing Up and Restoring Encryption Keys How to: Create a Report Server Database for SharePoint Integrated Mode (Reporting Services Configuration)

#### **Other Resources**

**RSReportDesigner Configuration File** 

**Planning a Deployment Mode** 

**Deployment Topologies for Reporting Services in SharePoint Integrated Mode** 

Granting Permissions on Report Server Items on a SharePoint Site Security Overview for Reporting Services in SharePoint Integrated Mode Deploying Models and Shared Data Sources to a SharePoint Site

## SQL Server Setup Help Reporting Services Configuration How-to Topics



- -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

This section contains procedures for configuring a report server instance using the Reporting Services Configuration tool.

## <sup>□</sup> In This Section

The topics in this section are organized into categories so that you can more easily find the instructions you want. The first section contains topics for basic configuration tasks for a native mode report server. The second section contains advanced configuration topics. The third section contains topics for configuring a report server to run in SharePoint integrated mode.

## **Basic Configuration**

Торіс	Description
How to: Start Reporting	Provides steps for starting the Reporting Services
Services Configuration	Configuration tool.
How to: Configure a Service Account for Reporting Services	Explains how to specify account and password information for the Report Server service.
How to: Register a Service	Explains how to manually register an SPN for a
Principal Name (SPN) for	report server that runs under a domain user account
<u>a Report Server</u>	on a network that uses Kerberos authentication.
How to: Configure a URL	Explains how to establish one or more URLs used to
(Reporting Services	access the Report Server Web service and Report
Configuration)	Manager.
<u>How to: Create a Report</u> <u>Server Database</u> ( <u>Reporting Services</u> <u>Configuration)</u>	Provides steps for creating a report server database. This step is required for deploying a Reporting Services installation.

## **Advanced or Optional Configuration**

Торіс	Description
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 for E-mail Delivery (Reporting Services Configuration)	Provides steps for configuring a report server for e-mail distribution.
<u>How to: Configure a Firewall for</u> <u>Report Server Access</u>	Explains how to open ports used for inbound requests and outbound responses from a report server.
How to: Configure a Report Server for Local Administration on Windows Vista and Windows Server 2008	Describes additional steps required to connect to Report Manager or a report server using http://localhost.
How to: Configure a Report Server for Remote Administration	Explains how to configure a remote report server instance so that you can connect to and configure it from a different computer.
How to: Turn Reporting Services Features On or Off	Explains how to remove unused features in a Reporting Services installation.
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.

## SharePoint Integration

Торіс	Description
How to: Install a SharePoint	Explains how to add SharePoint program files
<u>Web Front-end on a Report</u>	used to join a report server computer to a
<u>Server Computer</u>	SharePoint server farm.
How to: Install or Uninstall the	Explains how to install report server
Deporting Services Add in	application pages, configuration pages, and
<u>Reporting Services Add-in</u>	URL on a SharePoint server.
How to: Activate the Report	Explains how to manually activate the report
Server Feature in SharePoint	server feature for a site collection if the root
Central Administration	site collection is not enabled.
How to: Create a Report Server	Explains how to greate a report somer database
Database for SharePoint	Explains now to create a report server database
Integrated Mode (Reporting	ior a report server that runs in SharePoint
Services Configuration)	integrated mode.
How to: Configure Report	Provides instructions for configuring

Server Integration in SharePoint	integration in SharePoint Central
Central Administration	Administration.
How to: Configure SharePoint	Provides step-by-step instructions for installing
Integration on a Stand-alone	all server components and application
<u>Server</u>	databases on a single computer.
How to: Configure SharePoint	Provides step-by-step instructions for installing
Integration on Multiple Servers	server components on multiple computers in a
	distributed server deployment.
How to: Switch Sorver Modes	Explains how to remove a report server
(Poporting Sorvices	instance from a deployment of a SharePoint
Configuration)	technology or instance so that you can run it as
Comgulation	a native mode report server.
How Do LL open About	Provides a list of topics specific to the
ShareDoint Integrated Mode	Reporting Services Add-in for SharePoint
	Technologies.

## □ See Also

## Tasks

How to: Detect Version Information (Reporting Services)

## Concepts

How to: Migrate a Reporting Services Installation

#### **Other Resources**

Designing Ad Hoc Reports: How-to Topics Configuring a Report Server Installation Reporting Services Configuration Tool

## SQL Server Setup Help How to: Start Reporting Services Configuration



#### - ▼

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

The Reporting Services Configuration tool is used to configure a Reporting Services installation. You can configure a local or remote report server instance as long as you are a member of the local administrator group on the report server computer. Before you can connect to and configure a remote report server instance, you must enable remote Windows Management Instrumentation (WMI) calls to pass through Windows Firewall. For more information, see <u>How</u> to: <u>Configure a Report Server for Remote Administration</u> in SQL Server Books Online.

The Reporting Services Configuration tool is version-specific. The Reporting Services Configuration tool that installs with SQL Server 2008 cannot be used to configure an earlier version of Reporting Services. If you are running older and newer versions of Reporting Services side-by-side on the same computer, you must use the Reporting Service Configuration tool that comes with each version to configure each instance.

## <sup>©</sup> To start Reporting Services Configuration

1. Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2008**, and then point to **Configuration Tools**.

If you want to configure a report server instance from a previous version of SQL Server, open the program folder for that version. For example, point to **Microsoft SQL Server 2005** instead of **Microsoft SQL Server 2008** to open the configuration tools for SQL Server 2005 server components.

2. Click **Reporting Services Configuration**. The Connect to a Report Server Instance dialog box appears so that you can select the report

server instance you want to configure.

- 3. In **Server Name**, specify the name of the computer on which the report server instance is installed. The name of the local computer appears by default, but you can type the name of a remote SQL Server instance if you want to connect to a report server that is installed on a remote computer.
- 4. If you specify a remote computer, click **Find** to establish a connection.
- 5. In **Report Server Instance**, select the SQL Server Reporting Services instance that you want to configure. Only SQL Server 2008 report server instances appear in the list. You cannot configure earlier versions of Reporting Services.
- 6. Click **Connect**.

## □ See Also

## Concepts

**Reporting Services Configuration How-to Topics** 

## **Other Resources**

**Configuring a Report Server Installation Reporting Services Configuration Tool** 

SQL Server Setup Help

How to: Activate the Report Server Feature in SharePoint Central Administration



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

Report server integration settings are configured in SharePoint Central Administration after you install the Microsoft SQL Server 2008 Reporting Services Add-in for SharePoint Technologies. To configure report server integration, use the Reporting Services pages in Central Administration's **Application Management** tab.

If the Reporting Services section is not visible in **Application Management**, the Reporting Services Add-in is either not installed or the report server integration settings are not activated.

To verify installation, view the list of installed applications in **Add and Remove Programs**. If the Reporting Services Add-in is installed, follow the instructions in this topic to activate the feature.

By default, the Reporting Services Add-in is activated for the root site collection of the local SharePoint Web application. If the root site is not enabled or if you do not have a site collection defined when the Reporting Services Add-in is installed, you must activate the report server integration settings manually.

## <sup>□</sup> To activate Reporting Services integration on a Site Collection

- 1. Click Start, click **Administrative Tools**, and then click **SharePoint 3.0 Central Administration**.
- 2. Click **Site Actions**.
- 3. Click **Site Settings**.
- 4. Click Site Collection Features.
- 5. Find **Report Server Integration Feature** in the list.
- 6. Click Activate.

# <sup>□</sup> Next Steps

After the feature is activated, you can continue with server integration. For instructions, see <u>How to: Configure Report Server Integration in SharePoint</u> <u>Central Administration</u>.

## □ See Also

### Concepts

How to: Install or Uninstall the Reporting Services Add-in How to: Install or Uninstall the Reporting Services Add-in Configuring Reporting Services for SharePoint 3.0 Integration

**Other Resources** 

Requirements for Running Reporting Services in SharePoint Integrated Mode Planning a Deployment Mode

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help How to: Configure a Firewall for Report Server Access



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

Report server applications and published reports are accessed through URLs that specify an IP address, port, and virtual directory. If Windows Firewall is turned on, the port that the report server is configured to use is most likely closed. Indications that a port might be closed are the appearance of a blank Web page after requesting a report, or a blank page when you attempt to open Report Manager from a remote client computer.

To open a port, you must use the Windows Firewall utility on the report server computer. Reporting Services will not open ports for you; you must perform this step manually.

By default, the report server listens for HTTP requests on port 80. As such, the following instructions include steps that specify that port. If you configured the report server URLs to use a different port, you must specify that port number when following the instructions below.

If you are accessing SQL Server relational databases on external computers, or if the report server database is on an external SQL Server instance, you must open port 1433 and 1434 on the external computer. For more information, see **How to: Configure a Windows Firewall for Database Engine Access**. For more information about the default Windows firewall settings, and a description of the TCP ports that affect the Database Engine, Analysis Services, Reporting Services, and Integration Services, see <u>Configuring the Windows Firewall to</u> <u>Allow SQL Server Access</u>.

# <sup>©</sup> Prerequisites

These instructions assume that you already configured the service account, created the report server database, and configured URLs for the Report Server Web service and Report Manager. For more information, see <u>Reporting Services</u> <u>Configuration How-to Topics</u>.

You should also have verified that the report server is accessible over a local Web browser connection to the local report server instance. This step establishes that you have a working installation. You should verify that the installation is configured correctly before you begin opening ports. To complete this step on Windows Vista or Windows Server 2008, you must have also added the report server site to Trusted Sites. For more information, see <u>How to: Configure a</u> <u>Report Server for Local Administration on Windows Vista and Windows Server 2008</u>.

# <sup>©</sup> Opening Ports in Windows Firewall

There are separate instructions for different versions of Windows Firewall. If you are using Windows Vista or Windows Server 2008, please use the instructions at the end of this topic.

# <sup>©</sup> To open port 80 on Windows Server 2003 SP 2 or Windows XP SP2

- 1. From the **Start** menu, click **Control Panel**, and then click **Windows Firewall**.
- 2. On the Exceptions tab, click **Add Port**.
- 3. In Name, type **ReportServer (TCP on port 80)**.
- 4. In Port number, type **80**.
- 5. Verify that **TCP** is selected.
- 6. Click **Change Scope**.
- 7. Click **My network (subnet) only**, and then click **OK**.
- 8. Click **OK** to close the dialog box.
- 9. Restart the computer.

### <sup>□</sup> To open port 80 on Windows Vista

- 1. From the **Start** menu, click **Control Panel**, click **Security**, and then click **Windows Firewall**.
- 2. Click Allow a program through Windows Firewall.

- 3. Click **Continue**.
- 4. On the Exceptions tab, click **Add Port**.
- 5. In Name, type **ReportServer (TCP on port 80)**.
- 6. In Port number, type **80**.
- 7. Verify that **TCP** is selected.
- 8. Click Change Scope.
- 9. Click **My network (subnet) only**, and then click **OK**.
- 10. Click **OK** to close the dialog box.
- 11. Restart the computer.

# <sup>□</sup> Next Steps

After you open the port and before you confirm whether remote users can access the report server on the port that you open, you must grant user access to the report server through role assignments on Home and at the site level. You can open a port correctly and still have report server connections fail if users do not have sufficient permissions. For more information, see **How to: Grant User Access to a Report Server (Report Manager)**.

You can also verify that the port is opened correctly by starting Report Manager on a different computer. For more information, see **How to: Start Report Manager**.

## □ See Also

#### Tasks

How to: Verify a Reporting Services Installation

#### Concepts

Configuring the Report Server Service Account Configuring Report Server URLs Creating a Report Server Database Configuring the Report Server Service Account Reporting Services Configuration How-to Topics

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Configure a Report Server for E-mail Delivery (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

Reporting Services provides an e-mail delivery extension that allows users to subscribe to reports that are subsequently delivered to electronic mailboxes. The Report Server E-mail delivery extension must be configured before you can use it. Configuring a report server for e-mail delivery requires the following:

- A Simple Mail Transfer Protocol (SMTP) server or gateway.
- An account that has permission to send e-mail from the SMTP server. 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.
- The Report Server service must have **Send As** permissions on the SMTP server.

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 tool 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.

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 <u>Configuring a Report Server for E-Mail Delivery</u>.

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.

### <sup>□</sup> 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.
- 2. Start the Reporting Services Configuration tool and connect to the report server instance.
- 3. 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.
- 4. In **Sender Address**, enter the name an account that has permission to send e-mail from the SMTP server.
- 5. Click **Apply**.

## <sup>□</sup> 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 email 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 <u>Configuring a Report Server for E-Mail Delivery</u> and **RSReportServer Configuration File**.

### <sup>©</sup> 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: Set the Default Delivery Extension for Report Manager File Share Delivery in Reporting Services E-Mail Delivery in Reporting Services Reporting Services Configuration Tool Subscription and Delivery (Reporting Services)

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Configure a Report Server for Local Administration on Windows Vista and Windows Server 2008



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

Deploying Reporting Services on Windows Vista and Windows Server 2008 requires additional configuration steps if you want to administer a report server instance locally. Both Windows Vista and Windows Server 2008 limit the overuse of elevated permissions by removing administrator permissions when accessing applications. Because the operating system removes permissions, members of the local Administrators group run most applications as if they using the Standard User account.

While this practice improves the overall security of your system, it prevents you from using the predefined, built-in role assignments that Reporting Services creates for local administrators. However, with additional configuration on your part, you can effectively manage report server content and operations using standard user permissions if you do the following:

- Add Reporting Services URLs to trusted sites. By default, Internet Explorer 7.0 on Windows Vista and Windows Server 2008 runs in Protected Mode, a feature that blocks browser requests from reaching high-level processes that run on the same computer. You can disable protected mode for the report server applications by adding them as Trusted Sites.
- Create role assignments that grant you, the report server administrator, permission to manage content and operations without having to use the **Run as administrator** feature on Internet Explorer. By creating role assignments for your Windows user account, you gain access to a report server with Content Manager and System Administrator permissions through explicit role assignments that replace the predefined, built-in role assignments that Reporting Services creates for local administrators.

This topic explains how to configure the report server for local administration

after setup is finished and the report server components are configured. If you have not yet installed or configured the report server, see <u>How to: Install SQL</u> <u>Server 2008 (Setup)</u> and <u>Reporting Services Configuration How-to Topics</u>. For more information about User Account Control (UAC) features and how to connect to the Database Engine with administrator permissions, see **How to: Connect to SQL Server from Windows Vista**.

## <sup>☑</sup> Note:

If you are deploying Reporting Services on Windows Vista or Windows Server 2008, be sure to open port 80 in Windows Firewall to allow access to the report server computer. For instructions, see <u>How to: Configure a</u> <u>Firewall for Report Server Access</u>.

# <sup>□</sup> To configure local report server administration on Windows Vista and Windows Server 2008

- 1. Open a browser window with Run as administrator permissions. From the **Start** menu, click **All Programs**, right-click **Internet Explorer**, and select **Run as administrator**.
- 2. Click **Allow** to continue.
- 3. In the URL address, enter the Report Manager URL. For instructions, see **How to: Start Report Manager**.
- 4. Click **Tools**.
- 5. Click **Internet Options**.
- 6. Click Security.
- 7. Click **Trusted Sites**.
- 8. Click Sites.
- 9. Add http://<your-server-name>.

- 10. Clear the check box **Require server certification (https:) for all sites in this zone** if you are not using HTTPS for the default site.
- 11. Click **Add**.
- 12. Click **OK**.
- 13. In Report Manager, on the Home page, click **Properties**.
- 14. Click New Role Assignment.
- 15. Type your Windows user account in this format: <domain>\<user>.
- 16. Select Content Manager.
- 17. Click **OK**.
- 18. Click **Site Settings** in the upper corner of the Home page.
- 19. Click **Configure Site-wide security**.
- 20. Click New Role Assignment.
- 21. Type your Windows user account in this format: <domain>\<user>.
- 22. Select System Administrator.
- 23. Click **OK**.
- 24. Close Report Manager.
- 25. Re-open Report Manager in Internet Explorer, without using **Run as administrator**.

# <sup>□</sup> Using Business Intelligence Development Studio on Windows Vista and Windows Server 2008

If you installed Business Intelligence Development Studio on Windows Vista or Windows Server 2008, you will encounter compatibility issues with the Visual Studio environment when you use the design tools. If you want to open an existing project or solution, or publish items from Report Designer or Model Designer to a report server, you must open Business Intelligence Development Studio with elevated permissions:

- 1. From the **Start** menu, click **All Programs**, click **SQL Server 2008**, rightclick **Business Intelligence Development Studio**, and then click **Run as administrator**.
- 2. Click **Continue**.
- 3. Click **Run Program**.

You should now be able to deploy reports and other items to a report server. If you do not have sufficient permissions, you will encounter issues such as the following:

- Access Denied errors when you try to open an existing project.
- Login failure errors when you configure a data source connection to a shared data source.
- When using the Report Server Wizard to create a report, you will get a login failure error when the Wizard attempts to connect to the external data source that provides data to the report.

SQL Server Management Studio users must use **Run as administrator** permissions to create logins for users who administer databases on the server. For more information about elevated permission requirements for Management Studio, see **How to: Connect to SQL Server from Windows Vista**.

## □ See Also

### Tasks

How to: Start Reporting Services Configuration

### Reference

Hardware and Software Requirements for Installing SQL Server 2008

#### Concepts

**Reporting Services Configuration How-to Topics** 

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Configure a Report Server Scale-Out Deployment (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

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:

- Use Setup to install each report server instance that will be joined to the scale-out deployment.
- Use the Reporting Services Configuration tool to connect each report server to the shared database. You can only connect to and configure one report server at a time.
- Use the Reporting Services Configuration tool to complete the scale-out by joining new report server instances to the first report server instance already connected to the report server database.

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 SQL Server 2008 report server instance, all other instances in the same deployment must also be SQL Server 2008. For more information, see <u>Configuring Reporting Services for Scale-Out Deployment</u>.

The following instructions explain how to configure a two-node report server scale-out deployment. Repeat the steps described in this topic to add additional report server nodes to the deployment.

If you are planning to host the scale-out deployment on a network load balanced (NLB) cluster, you should configure the NLB cluster before you configure the scale-out deployment. For more information, see <u>How to: Configure a Report</u>

Server on a Network Load Balancing Cluster.

# <sup>©</sup> To install a SQL Server instance to host the report server databases

- 1. Install a SQL Server instance on a computer that will host the report server databases. At a minimum, install SQL Server Database Engine and Reporting Services.
- 2. If necessary, enable the report server for remote connections. Some versions of SQL Server do not enable remote TCP/IP and Named Pipes connections by default. To confirm whether remote connections are allowed, use SQL Server Configuration Manager and view the network configuration settings 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 to connect to the named instance.

## <sup>□</sup> To install the first report server instance

- Install the first report server instance that is part of the deployment. When you install Reporting Services, choose the **Install but do not configure server** option on the Report Server Installation Options page.
- 2. Start the Reporting Services Configuration tool.
- 3. Configure the Report Server Web service URL, Report Manager URL, and the report server database. For more information, see **How to: Configure a Report Server**.
- 4. Verify that the report server is operational. For more information, see <u>How to: Verify a Reporting Services Installation</u>.

### <sup>□</sup> 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. When you install Reporting Services, choose the **Install but do not configure server** option on the Report Server Installation Options page.
- 2. Start the Reporting Services Configuration tool and connect to the new instance you just installed.
- 3. Connect the report server to the same database you used for the first report server instance:
  - 1. Click **Database** to open the Database page.
  - 2. Click **Change Database**.
  - 3. Click **Choose an existing report server database**.
  - 4. Type the server name of the SQL Server 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 the previous set of the instructions.
  - 5. Click **Test Connection**, and then click **Next**.
  - 6. In **Report Server Database**, select the database you created for the first report server, and then click **Next**. The default name is **ReportServer**. Do not select **ReportServerTempDB**; it is used only for storing temporary data when processing reports. If the database list is empty, repeat the previous four steps to establish a connection to the server.
  - 7. In the Credentials page, 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. Click **Next**.
  - 8. Click **Summary** and then click **Finish**.
- 4. Configure the Report Server Web service URL. Do not test the URL yet. It will not resolve until the report server is joined to the scale-out deployment.
- 5. Configure the Report Manager URL. Do not test the URL yet or try to

verify the deployment. The report server will be unavailable until the report server is joined to the scale-out deployment.

# <sup>©</sup> To join the second report server instance to the scale-out deployment

- 1. Open the Reporting Services Configuration tool, and reconnect to the first report server instance. The first report server is already initialized for reversible encryption operations, so it can be used to join additional report server instances to the scale-out deployment.
- 2. Click **Scale-out Deployment** to open the Scale-out Deployment page. You should see two entries, one for each report server instance that is connected to the report server database. The first report server instance should be joined. The second report server should be "Waiting to join". If you do not see similar entries for your deployment, verify you are connected to the first report server that is already configured and initialized to use the report server database.

Server	Instance	Status
SERVER01	MyNamedInstance	Waiting to Join
SERVER02	MSSQLSERVER	Joined

- 3. On the Scale-out Deployment page, select the report server instance that is waiting to join the deployment, and click **Add Server**.
- 4. You should now be able to verify that both report server instances are operational. To verify the second instance, you can use the Reporting Services Configuration tool to connect to the report server and click the Web Service URL or the Report Manager URL.

If you plan to run the report servers in a load-balanced server cluster, additional configuration is required. For more information, see <u>How to: Configure a Report</u> <u>Server on a Network Load Balancing Cluster</u>.

## □ See Also

#### Tasks

How to: Configure a Service Account for Reporting Services How to: Configure a URL (Reporting Services Configuration) How to: Create a Report Server Database (Reporting Services Configuration) How to: Start Reporting Services Configuration

#### Concepts

Configuring Report Server URLs Configuring a Report Server Database Connection Adding and Removing Encryption Keys for Scale-Out Deployment Reporting Services Configuration How-to Topics

#### **Other Resources**

**Planning for Scale-Out Deployment** 

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Configure a Report Server on a Network Load Balancing Cluster



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

If you are configuring a report server scale-out to run on a Network Load Balancing (NLB) cluster, you must do the following:

- Ensure that the NLB cluster is accessible through a virtual server name that maps to the virtual server IP address. A virtual server name is necessary so that you can configure a single point of entry to the NLB cluster. When you configure a URL for each report server instance, you will specify the virtual server name as the host.
- Configure view state validation to support interactive report viewing. Interactive reports are typically rendered numerous times during a single user session to visualize new or different data in response to user actions. By configuring view state validation, continuity is preserved within the user session regardless of which report server services the actual request.

Reporting Services does not provide functionality for load balancing a scale-out deployment or for defining a single point of access through a shared URL. You must implement a separate software or hardware NLB cluster solution to support a Reporting Services scale-out deployment.

You can install Reporting Services on nodes that are already part of an NLB cluster, or you can configure a scale-out deployment first and then install cluster software.

# <sup>□</sup> Steps for Report Server Deployment on an NLB Cluster

Use the following guidelines to install and configure your deployment:

Step	Description	More information
1	Before you install Reporting Services on server nodes in an NLB cluster, check the requirements for scale-out deployment.	Planning for Scale- Out Deployment
	Configure the NLB cluster and verify it is working correctly.	For more information, see the Windows Server product
2	Be sure to map a host header name to the virtual server IP of the NLB cluster. The host header name is used in the report server URL, and is easier to remember and type than an IP address.	documentation for the version of the Windows operating system that you run.
4	Install Reporting Services in files-only mode on nodes that are already part of a NLB cluster, and configure the report server instances for scale-out deployment. The scale-out that you configure might not respond to requests that are directed to the virtual server IP. Configuring the scale-out to use the virtual server IP occurs at a later step, after you configure view state validation.	How to: Configure a Report Server Scale- Out Deployment (Reporting Services Configuration)
5	Configure view state validation. For best results, perform this step after you configure the scale-out deployment, and before you configure the report server instances to use the virtual server IP. By configuring view state validation first, you can avoid exceptions about failed state validation when users attempt to access interactive reports.	<u>How to Configure</u> <u>View State Validation</u> in this topic.
	Configure <b>HostName</b> and <b>UrlRoot</b> to use the	How to Configure

6	virtual server IP of the NLB cluster.	HostName and
		UrlRoot in this topic.
7	Verify the servers are accessible through the host	Verify Report Server
	name you specified.	Access in this topic.

# <sup>□</sup> How to Configure View State Validation

To run a scale-out deployment on an NLB cluster, you must configure view state validation so that users can view interactive HTML reports. You must do this for the report server and for Report Manager.

View state validation is controlled by the ASP.NET. By default, view state validation is enabled and uses the identity of the Web service to perform the validation. However, in an NLB cluster scenario, there are multiple service instances and web service identities that run on different computers. Because the service identity varies for each node, you cannot rely on a single process identity to perform the validation.

To work around this issue, you can generate an arbitrary validation key to support view state validation, and then manually configure each report server node to use the same key. You can use any randomly generated hexadecimal sequence. The validation algorithm (such as SHA1) determines how long the hexadecimal sequence must be.

 Generate a validation key and decryption key by using the autogenerate functionality provided by the .NET Framework. In the end, you must have a single <machineKey> entry that you can paste into the Web.config file for each Report Manager instance in the scale-out deployment. The following example provides an illustration of the value you must obtain. Do not copy the example into your configuration files; the key values are not valid.

<machineKey validationKey="123455555" decryptionKey="678999999"

- Open the Web.config file for Report Manager, and in the <system.web> section paste the <machineKey> element that you generated. By default, the Report Manager Web.config file is located in \Program Files\Microsoft SQL Server\MSRS10.MSSQLSERVER\Reporting Services\ReportManager\Web.config.
- 3. Save the file.
- 4. Repeat the previous step for each report server in the scale-out deployment.
- 5. Verify that all Web.Config files in the \Reporting Services\Report Manager folders contain identical **<machineKey**> elements in the **<system.web**> section.

# <sup>□</sup> How to Configure HostName and UrlRoot

To configure a report server scale-out deployment on an NLB cluster, you must define a single virtual server name that provides a single point of access to the server cluster. Then register this virtual server name with the Domain Name Server (DNS) in your environment.

After you define the virtual server name, you can configure the **HostName** and **UrlRoot** properties in the RSReportServer.config file to include the virtual server name in the report server URL.

Configure the **HostName** property when you are using wildcard URL reservations in your reporting environment. When you specify the **HostName** property to be the virtual server name of the NLB server, network traffic for the reporting environment is directed to the NLB server. The NLB then distributes requests among the report server nodes.

Additionally, configure the **UrlRoot** property so that report links work in reports that have been exported to static reports, such as in an Excel or PDF format, or in reports that are generated by subscriptions, such as e-mail subscriptions.

If you integrate Reporting Services with Windows SharePoint Services 3.0 or Office SharePoint Server 2007, or you host your reports in a custom Web application, you might need to configure only the **UrlRoot** property. In this case, configure the **UrlRoot** property to be the URL of the SharePoint site or Web application. This will direct network traffic for the reporting environment to the application that handles the reports rather than to the report server or NLB cluster.

Do not modify **ReportServerUrl**. If you modify this URL, you will introduce an extra roundtrip through the virtual server each time an internal request is handled. For more information, see <u>URLs in Configuration Files (Reporting Services</u>). For more information about editing the configuration file, see **How to: Modify a Reporting Services Configuration File**.

- 1. Open RSReportServer.config in a text editor.
- 2. Find the **<services**> section, and add the following information to the configuration file, replacing the **HostName** value with the virtual server

name for your NLB server:

Copy Code

<HostName>virtual\_server</HostName>

- 3. Find **UrlRoot**. The element is unspecified in the configuration file, but the default value used is a URL in this format: http:// or https://<*computername*>/<*reportserver*>, where <*reportserver*> is the virtual directory name of the Report Server Web service.
- 4. Type a value for **UrlRoot** that includes the virtual name of the cluster in this format: http:// or https://<*virtual\_server*>/<*reportserver*>.
- 5. Save the file.
- 6. Repeat these steps in each RSReportServer.config file for each report server in the scale-out deployment.

# <sup>©</sup> Verify Report Server Access

Verify that you can access the scale-out deployment through the virtual server name (for example, https://MyVirtualServerName/reportserver and https://MyVirtualServerName/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** that shows which instance processed a particular request). For more information, see **Reporting Services Log Files**.

If you cannot connect to the report server, check the NLB to ensure that requests are sent to the report server and view the report server HTTP log to ensure that the server is receiving the requests.

#### **Troubleshooting Failed Requests**

If requests do not reach the report server instances, check the RSReportServer.config file to verify that the virtual server name is specified as the host name for the report server URLs:

- 1. Open the RSReportServer.config file in a text editor.
- 2. Find **<HostName>**, **<ReportServerUrl>**, and **<UrlRoot>**, and check the host name for each settings. If the value is not the host name you expect, replace it with the correct host name.

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

#### Tasks

How to: Start Reporting Services Configuration How to: Configure a URL (Reporting Services Configuration) How to: Configure a Report Server Scale-Out Deployment (Reporting Services Configuration)

#### Concepts

**Reporting Services Configuration How-to Topics** 

### **Other Resources**

Planning for Scale-Out Deployment RSReportServer Configuration File

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help How to: Configure a Report Server for Remote Administration

# 

#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

In Reporting Services, you can configure report server instances locally or remotely. To configure a remote report server instance, you can use the Reporting Services Configuration tool or write custom code that uses the Reporting Services Windows Management Instrumentation (WMI) provider. The Reporting Services Configuration tool provides a graphical interface to the WMI provider so that you can configure a report server without having to write code. When you start the tool, you can specify a remote server to connect to.

Before you can use the tool to configure a remote report server, you must follow the instructions in this topic to enable ports in Windows Firewall, enable remote connections, and enable remote WMI requests.

Proper configuration helps you avoid the following error:

The machine could not be found.

"The RPC server is unavailable. (Exception from HRESULT: 0x800706BA)".

# <sup>©</sup> Prerequisites

To modify firewall settings, you must be logged on locally and you must be a member of the local Administrators group. You cannot modify the Windows firewall settings of a remote computer over a remote connection.

If you want to enable remote administration for a non-administrator user, you must grant the account Distributed Component Object Model (DCOM) Remote Activation permissions. Instructions for configuring the server for non-administrator access are provided in this topic.

Some organizations have group policies that prevent remote server administration for certain operating systems or users. Before you begin modifying firewall settings, check with your network administrator to verify whether there are restrictions on remote administration.

For more information, see <u>Connecting Through Windows Firewall</u> in the Platform SDK documentation on MSDN.

# <sup>□</sup> Tasks

Tasks that enable remote report server configuration include the following:

- Enable ports in Windows Firewall to allow requests on ports used by the report server and by the SQL Server Database Engine instance.
- Enable remote connections to the instance of the Database Engine instance that hosts the report server database. A remote connection is necessary for configuring the report server database connection and managing the encryption keys.
- Enable remote WMI requests to pass through the Microsoft Windows firewall.
- If you are configuring a remote report server for administration by a nonadministrative user, you must set DCOM permissions to enable remote WMI access to a standard Windows user account. Because WMI uses DCOM as transport for remote calls, you must set the DCOM permissions so that users who are not logged on as the local administrator can configure the server.
- If you are configuring a remote report server for administration by a nonadministrative user, you must also set WMI permissions on the report server WMI namespace. By default, all members of the local Administrator group have access to the report server WMI namespace. If you want to grant access to non-administrators, you must set permissions.

Instructions on how to perform these tasks are provided in this topic.

# <sup>©</sup> To open ports in Windows Firewall

- 1. How to: Configure a Windows Firewall for Database Engine Access.
- 2. <u>How to: Configure a Firewall for Report Server Access</u>.

### <sup>□</sup> To configure remote connections to the report server database

- 1. Click **Start**, point to **Programs**, point to **Microsoft SQL Server 2008**, point to **Configuration Tools**, and click **SQL Server Configuration Manager**.
- 2. In the left pane, expand **SQL Server Network Configuration**, and then click **Protocols** for the instance of SQL Server.
- 3. In the details pane, enable the TCP/IP and Named Pipes protocols, and then restart the SQL Server service.

## <sup>□</sup> To enable remote administration in Windows Firewall

- 1. Log on as a local administrator to the computer for which you want to enable remote administration.
- 2. If the report server is running on Windows Vista, right-click **Command Prompt** and select **Run as administrator**. For other operating systems, open a command prompt window.
- 3. Run the following command:

netsh.exe firewall set service type=REMOTEADMIN mode=ENABLE s

You can specify different options for Scope. For more information, see the Windows Firewall product documentation.

4. Verify that remote administration is enabled. You can run the following command to show the status:

Copy Code

netsh.exe firewall show state

5. Reboot the computer.
# <sup>□</sup> To set DCOM permissions to enable remote WMI access for non-administrators

1. On the Start menu, point to **Administrative Tools**, click **Component Services**.

For Windows Vista, on the Start menu, click **All Programs**, click **Run**, and then enter **mmc comexp.msc**.

- 2. Open the Component Services folder.
- 3. Open the Computers folder.
- 4. Select My Computer.
- 5. On the **Action** menu, and select **Properties**.
- 6. Click **COM Security**.
- 7. In Launch and Activation Permissions, click Edit Limits.
- 8. If you do not see your name in **Launch Permission**, click **Add**.
- 9. Type the name of your user account, and then click **OK**.
- 10. In **Permissions for <User or Group>**, in the **Allow** column, select **Remote Launch** and **Remote Activation**, and then click **OK**.

# <sup>□</sup> To set permissions on the report server WMI namespace for non-administrators

- 1. On the Start menu, point to **Administrative Tools**, click **Computer Management**.
- 2. Open the Services and Applications folder.

- 3. Right-click WMI Control, and select Properties.
- 4. Click Security.
- 5. Open the Root folder.
- 6. Open the Microsoft folder.
- 7. Open the SQLServer folder.
- 8. Open the ReportServer folder.
- 9. Open instance folder. If you installed the default instance, the folder is MSSQLSERVER.
- 10. Open the v10 folder.
- 11. Select the Admin folder, and then click **Security**.
- 12. Click **Add**, and then type the user account you will use to manage the server.
- 13. In the Allow column, select **Enable Account**, **Remote Enable**, and **Read Security**, and then click **OK**.

## □ See Also

Tasks

How to: Start Reporting Services Configuration

**Other Resources** 

Reporting Services Configuration Tool Reporting Services WMI Provider Configuring a Report Server Installation

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help

How to: Configure a Service Account for Reporting Services



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

In a Reporting Services installation, the Report Server Web service, Report Manager, and the background processing application run within a single service. The account under which the service runs is defined during Setup when you specify the account in the Service Identity page, but you can use the Reporting Services Configuration tool if you want use a different account or update the password.

If you have a report server that is configured to use SharePoint Integrated mode and you change the service account by using the Reporting Services Configuration tool, you must also open SharePoint Central Administration and use the Reporting Services **Grant Database Access** page to re-apply the report server and instance settings. This step will grant the new service account access to the SharePoint databases, which is required for integrating Reporting Services with Windows SharePoint Services (WSS) or Microsoft Office SharePoint Server (MOSS). For more information about how to grant database access in SharePoint Central Administration, see <u>How to: Configure Report Server</u> <u>Integration in SharePoint Central Administration</u>.

Always use the Reporting Services Configuration tool to update the service account so that other settings that depend on the service identity can be updated concurrently.

## <sup>©</sup> To configure the Report Server service account

- 1. Start the Reporting Services Configuration tool and connect to the report server.
- 2. On the Service Account page, select the option that describes the type of account you want to use. For recommendations about which account

type to specify, see <u>Configuring the Report Server Service Account</u>.

3. If you selected a Windows user account, specify the new account and password. The account cannot be more than 20 characters.

If the report server is deployed in a network that supports Kerberos authentication, you must register the report server Service Principal Name (SPN) with the domain user account you just specified. For more information, see <u>How to: Register a Service Principal Name (SPN) for a Report Server</u>.

- 4. Click Apply.
- 5. When prompted to back up the symmetric key, type a file name and location for the symmetric key backup, type a password to lock and unlock the file, and then click **OK**.
- 6. If the report server uses the service account to connect to the report server database, the connection information will be updated to use the new account or password. Updating the connection information requires that you connect to the database. If the SQL Server **Database Connection** dialog box appears, enter credentials that have permission to connect to the database, and then click **OK**.
- 7. When prompted to restore the symmetric key, type the password you specified in step 5, and then click **OK**.
- 8. Review the status messages in the Results pane to verify all tasks completed successfully.

## <sup>©</sup> Troubleshooting Service Identity Update Errors

Changing the service identity initiates a series of events that include restarting the service, updating the password-protected encryption key, updating URL reservations, and updating the report server database connection information if you are using the service account to connect to the report server database. You can monitor the status of these events by viewing the notifications in the Results panel at the bottom of the page. If errors occur during this process, you can try to resolve them using the following techniques:

- If the symmetric key cannot be restored, you can try to restore it manually by using **Restore** in the Encryption Keys page. If that does not work, consider deleting the encrypted content. You will have to re-create data source connection information and subscriptions, but the rest of your content will still be available. For more information, see <u>Backing Up and Restoring Encryption Keys</u>.
- If the service will not start, restart it manually by using the Services console application in Administrator Tools.
- URL reservation errors can occur when you update the service account. Each URL reservation includes a security descriptor that includes a Discretionary Access Control List (DACL) that grants permission to the service account to accept requests on the URL. When you update the account, the URL must be recreated to update the DACL with the new account information. If the URL reservation cannot be recreated, and you know the account to be valid, try to restart the computer. If the error persists, try to use a different account.

## □ See Also

#### Tasks

How to: Start Reporting Services Configuration

#### Reference

Service Account (Reporting Services Configuration)

#### Concepts

<u>Configuring the Report Server Service Account</u> <u>Configuring a Report Server Database Connection</u> <u>Configuring and Managing Encryption Keys</u>

#### **Other Resources**

**Service Architecture (Reporting Services)** 

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help How to: Configure a URL (Reporting Services Configuration)

# 

#### -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

Before you can use Report Manager or the Report Server Web service, you must configure at least one URL for each application. Configuring the URLs is mandatory if you installed Reporting Services in "files-only" mode (that is, by selecting the **Install but do not configure the server** option on the Report Server Installation Options page in the Installation Wizard). If you installed Reporting Services in the default configuration, URLs are already configured for each application. If you have a report server that is configured to use SharePoint Integrated mode and you update the Report Server Web Service URL by using the Reporting Services Configuration tool, you must also update the URL in SharePoint Central Administration.

Use the Reporting Services Configuration tool to configure the URLs. All parts of the URL are defined in this tool. Unlike earlier releases, Internet Information Services (IIS) Web sites no longer provide access to Reporting Services applications in SQL Server 2008.

Reporting Services provides default values that work well in most deployment scenarios, including side-by-side deployments with other Web services and applications. Default URLs incorporate instance names, minimizing the risk of URL conflicts if you run multiple report server instances on the same computer.

This topic provides instructions for the following tasks:

- Create a URL for the Report Server Web service.
- Create a URL for Report Manager.
- Set advanced URL properties to define additional URLs.

For more information about how URLs are stored and maintained or interoperability issues, see <u>About URL Reservations and Registration (Reporting</u> <u>Services</u>) and **Deploying Reporting Services and Internet Information Services Side-by-Side**. To review examples of URLs often used in a Reporting Services installation, see <u>Examples of URLs</u> in this topic.

## <sup>©</sup> Prerequisites

Before you create or modify a URL, remember the following points:

- You must be a member of the local Administrators group on the report server computer.
- If IIS 6.0 or 7.0 is installed on the same computer, check the names of virtual directories on any Web site that uses port 80. If you see any virtual directories that use the default Reporting Services virtual directory names (that is, "Reports" and "ReportServer"), choose different virtual directory names for the Reporting Services URLs that you configure.
- You must use the Reporting Services Configuration tool to configure the URL. Do not use a system utility. Never modify URL reservations in the **URLReservations** section of the RSReportServer.config file directly. Using the Reporting Services Configuration tool is necessary to update both the underlying URL reservation that is stored internally and synchronize the URL settings stored in the RSReportServer.config file.
- Choose a time that has low report activity. Each time the URL reservation changes, you can expect that the application domains for Report Server Web service and Report Manager might be recycled.
- For an overview of URL construction and usage in Reporting Services, see <u>Configuring Report Server URLs</u>.

## <sup>©</sup> To configure a URL for the Report Server Web service

- 1. Start the Reporting Services Configuration tool and connect to a local report server instance.
- 2. Click Web Service URL.
- 3. Specify the virtual directory. The virtual directory name identifies which application receives the request. Because an IP address and port can be shared by multiple applications, the virtual directory name specifies which application receives the request.

This value must be unique to ensure that the request reaches its intended destination. This value is required. It is case-insensitive. There is a one-to-one correspondence between a virtual directory name and an instance of a Reporting Services application. If you create multiple URLs to the same application instance, you must use the same virtual directory name in all of the URLs you define for this application instance.

For the Report Server Web service, the default virtual directory name is **ReportServer**.

- 4. Specify the IP address that uniquely identifies the report server computer on the network. If you want to specify a host header or define additional URLs for the same application instance, you must click Advanced. For instructions on how to set advanced properties on the URL, see the instructions later in this topic. Otherwise, use the Web Service URL page to select from the following values:
  - **All Assigned** specifies that any of the IP addresses that are assigned to the computer can be used in a URL that points to a report server application. This value also encompasses friendly host names (such as computer names) that can be resolved by a domain name server to an IP address that is assigned to the computer. This is the default value for a Reporting Services URL.
  - **All Unassigned** specifies that the report server will receive any request that has not been handled by another application. We recommend that you avoid this option. If you select this option, it becomes possible for another application that has a stronger URL reservation to intercept requests intended for the report server.
  - **127.0.0.1** is the IPv4 address used to access to localhost. It supports local administration on the report server computer. If you select only this value, only users who are logged on locally to the report server computer will have access to the application.
  - **::1** is the loopback address in IPv6 format.
  - Specific IP addresses also appear in this list. IP addresses can be in IPv4 and IPv6 formats. *Nnn.nnn.nnn* is the 32-bit IPv4 address of a network adapter card on your computer. IPv6 addresses are 128-bit, with eight 4-byte fields separated by colons:
    <prefix>:nnn:nnn:nnn:nnn:nnn

If you have multiple cards or if your network supports both IPv4

and IPv6 addresses, you will see multiple IP addresses. If you select only one IP address, it will limit application access to the just the IP address (and any host name that a domain name server maps to that address). You cannot use localhost to access a report server, and you cannot use the IP addresses of other network adapter cards that are installed on the report server computer. Typically, if you select this value, it is because you are configuring multiple URL reservations that also specify explicit IP addresses or host names (for example, one for a network adapter card used for intranet connections and a second one used for extranet connections).

- 5. Specify the port. Port 80 is the default for Reporting Services on Windows Server 2003, Windows Vista, and Windows Server 2008 because it can be shared with other applications. If you are using 32-bit Windows XP SP2, the default port is 8080. If you want to use a custom port number, remember that you will have to always specify it in the URL used to access the report server. You can use the following techniques to find an available port:
  - From a command prompt, type the following command to return a list of TCP ports that are being used: netstat -a -n -p tcp
  - Review the Microsoft Support article, <u>Information about TCP/IP</u> <u>port assignments</u>, to read about TCP port assignments and the differences between Well Known Ports (0 through 1023), Registered Ports (1024 through 49151), and Dynamic or Private Ports (49152 through 65535).
  - If you are using Windows Firewall, you must open the port. For instructions, see <u>How to: Configure a Firewall for Report Server</u> <u>Access</u>.
- 6. If you have not done so already, verify that IIS (if it is installed) does not have virtual directory with the same name you plan to use.
- 7. If you installed an SSL certificate, you can select it now to bind the URL to the SSL certificate that is installed on your computer.
- 8. Optionally, if you select an SSL certificate, you can specify a custom port. The default is 443 but you can use any port that is available.

- 9. Click **Apply** to create the URL.
- 10. Test the URL by clicking the link in the **URLs** section of page. Note that the report server database must be created and configured before you can test the URL. For instructions, see <u>How to: Create a Report</u> <u>Server Database (Reporting Services Configuration)</u>.
- 11. Additionally, if your report server is configured to use SharePoint Integrated mode, configure the Report Server Web service URL in SharePoint Central Administration. For more information about how to update the Report Server Web service URL in SharePoint Central Administration, see <u>How to: Configure Report Server Integration in</u> <u>SharePoint Central Administration</u>.

## <sup>□</sup> To create a URL reservation for Report Manager

- 1. Start the Reporting Services Configuration tool and connect to the report server instance.
- 2. Click **Report Manager URL**.
- 3. Specify the virtual directory. Report Manager listens on the same IP address and port as the Report Server Web service. If you configured Report Manager to point to a different Report Server Web service, you must modify the Report Manager URL settings in the RSReportServer.config file. For instructions, see **How to: Configure Report Manager**.
- 4. If you installed an SSL certificate, you can select it to require that all requests to Report Manager are routed over HTTPS.

Optionally, if you select an SSL certificate, you can specify a custom port. The default is 443 but you can use any port that is available.

- 5. Click **Apply** to create the URL.
- 6. Test the URL by clicking the link in the **URLs** section of page.

## Setting Advanced Properties to Specify Additional URLs

You can reserve multiple URLs for the Report Server Web service or Report Manager by specifying different ports or host names (either an IP address or a host header name that a domain name server can resolve to an IP address assigned to the computer). By creating multiple URLs, you can set up different access paths to the same report server instance. For example, to enable intranet and extranet access to a report server, you might use the default URL for access across the intranet, and an additional fully qualified host name for extranet access:

- http://myserver01/reportserver
- http://www.adventure-works.com/reportserver

You cannot set multiple virtual directory names for the same application instance. Each Reporting Services application instance is mapped to a single virtual directory name. If you have multiple instances of Reporting Services on the same computer, the virtual directory name for an application should include the instance name to ensure that each request reaches its intended target.

## <sup>□</sup> To set advanced properties on a URL

- 1. On either the **Web Service URL** or **Report Manager URL** page, click **Advanced**.
- 2. Click **Add**.
- 3. Click IP Address or Host Header Name. If you specify a host header, be sure to specify a name that the DNS service can resolve. If you are specifying publicly available domain name, include the whole URL, including http://www.
- 4. Specify the port. If you specify a custom port, the URL to the application must always include the port number.

- 5. Click **OK**.
- 6. Test the URL by opening a browser window and entering the URL.

# <sup>□</sup> URLs for Multiple Report Server Instances on the Same Computer

If you are reserving URLs for multiple instances of Reporting Services, you should follow naming conventions so that you can avoid naming conflicts. For more information, see <u>URL Reservations for Multi-Instance Report Server</u> <u>Deployments</u>.

# Examples of URL Configurations

The following list shows some examples of what a report server URL might resemble:

- http://localhost/reportserver
- 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 that you want):

- http://localhost/reports
- http://localhost/reports\_SQLEXPRESS
- http://sales01/reports
- http://sales01:8080/reports
- https://sales.adventure-works.com/reports
- https://www.adventure-works.com:8080/reports

## $\Box$ See Also

Tasks

How to: Start Reporting Services Configuration

Concepts Configuring Report Server URLs

## Other Resources

**Configuring Authentication in Reporting Services** 

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Configure Report Server Integration in SharePoint Central Administration



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

If you are configuring a report server to run within a deployment of a SharePoint product or technology, 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:
- 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 connected to a report server database that was created for SharePoint integrated mode of operations.
- If you are configuring Reporting Services to run within a SharePoint server farm, you must also have a SharePoint Web front-end installed on the report server computer.

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 that are enabled in a SharePoint farm and that have an installation of the Reporting Services Add-in.

If you later modify the Report Server service accounts or extend a SharePoint farm to include additional applications, you must re-run the grant database access command to update the access rights in the SharePoint configuration database. For more information, see <u>Configuring the Report Server Service Account</u>.

For more information about the complete set of tasks, see <u>Configuring Reporting</u> <u>Services for SharePoint 3.0 Integration</u>.

## <sup>ℤ</sup>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.

# <sup>□</sup> 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.

In the Reporting Services section, you should see the following links: Manage integration settings, Grant database access, Set server defaults. If you do not see the section or the links, the Reporting Services Add-in is not installed on the computer or the feature is not activated. For more information, see <u>How to: Activate the Report Server Feature in</u> <u>SharePoint Central Administration</u>.

- 3. Click Manage integration settings.
- 4. In **Report Server Web Service URL**, specify the report server site that you want to use 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.

To get the URL, open the Reporting Services Configuration tool, connect to the report server, and click **Web Service URL**. Click the URL to verify it works. Copy the URL and paste it into **Report Server Web Service URL**.

The report server URL includes a server name and a virtual directory

that you specified in the Reporting Services Configuration tool. The following examples illustrate how a report server URL might be constructed:

- 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 runs on Microsoft Windows XP SP2).
- https://your-server-name:443/reportserver.

Do not specify http://localhost for a report server URL that is used throughout a SharePoint farm. A URL address that specifies http://localhost will not be valid for SharePoint Web front-ends that are installed on remote servers.

If you installed the report server and the SharePoint technology instance side-by-side, check for virtual directory name conflicts. For more information about interoperability issues, see **Deploying Reporting Services and Internet Information Services Side-by-Side**.

5. In **Authentication mode**, select either **Windows Authentication** or **Trusted Authentication** to specify whether the report server endpoint installed on the SharePoint server sends a header with a security token for an impersonated connection on the report server.

All requests that originate from a user action on a Reporting Services application page are forwarded by the report server endpoint on the SharePoint Web application to a report server for processing. Depending on the authentication mode, the request may or may not include a SharePoint user identity token in the header.

If the SharePoint Web application is configured for Windows Authentication and the Kerberos protocol is enabled, you can select **Windows Authentication** to connect through Windows integrated security. In this case, no user identity token is sent. Optionally, you can select **Trusted Authentication** to impersonate the trusted account and pass the SharePoint user information in the request header to the report server.

If the SharePoint Web application is configured for Forms

authentication, the request header will always include SharePoint user identity and any value you specify for Authentication mode is ignored. For more information about server connections, see **Security Overview for Reporting Services in SharePoint Integrated Mode**.

- 6. Click **OK**.
- 7. 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 are created for the Report Server 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. The Report Server service must be able to connect as trusted users with read and write permissions.
- 8. Specify the name of the report server. By default, this is the name of the computer.
- 9. 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 Connect to a Report Server Instance 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.
- 10. Click **OK**.

The Enter Credentials dialog box opens.

- 11. In the **Enter Credentials** dialog box, specify the User Name and Password to connect to the report server to retrieve the service account information. Be sure to enter credentials for an account that is a member of the local Administrators group on the report server computer.
- 12. In the Reporting Services section, click **Set server defaults**.
- 13. 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.

- 14. 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. The value that you specify must be sufficient to complete both data processing and report processing.
- 15. 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 **Report Server Service Trace Log**.
- 16. 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.
- 17. 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.
- 18. Click **OK**.

## □ See Also

#### Tasks

How to: Configure SharePoint Integration on a Stand-alone Server How to: Configure SharePoint Integration on Multiple Servers How to: Install a SharePoint Web Front-end on a Report Server Computer

#### Concepts

How to: Install or Uninstall the Reporting Services Add-in How to: Create a Report Server Database for SharePoint Integrated Mode (Reporting Services Configuration)

#### **Other Resources**

Reporting Services and SharePoint Technology Integration Managing Report History Viewing and Managing Report Server Items from a SharePoint Site Setting Time-out Values for Report Processing

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help

How to: Configure SharePoint Integration on a Stand-alone Server



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

One of the deployment scenarios for running Reporting Services in SharePoint integrated mode is to install all server components and application databases on a single, stand-alone server. This configuration is useful if you want to learn how the technologies work together, or you are developing a custom application and want the simplicity of having all services on a single computer. The following illustration shows the components you must install.



## <sup>©</sup> Prerequisites

Before you begin installing components, be sure to do the following:

- Review the introduction to deployment configurations so that you are familiar with the components that you are about to install. For more information, see **Deployment Topologies for Reporting Services in SharePoint Integrated Mode**.
- Review the configuration overview topic for best-practice recommendations on service account configuration and to learn more about the configuration steps. For more information, see <u>Configuring Reporting</u> <u>Services for SharePoint 3.0 Integration</u>.
- Review the hardware and software requirements that apply to all deployment scenarios that include Reporting Services in SharePoint integrated mode. For more information, see **Requirements for Running Reporting Services in SharePoint Integrated Mode**.
- Review the remaining sections in this topic to learn about additional requirements that apply to a stand-alone server configuration.

## Database Considerations for Standalone Server Configuration

Both Reporting Services and SharePoint products and technologies use SQL Server relational databases to store application data. Reporting Services requires an instance of the Database Engine in a compatible edition. SharePoint products and technologies can use an existing database server if you provide one. If you do not, the Setup program will install SQL Server Embedded Edition as an internal component and use it for the application databases. Because you are already installing the Database Engine for Reporting Services, you might consider using that same instance for the SharePoint application databases.

However, if you allow Setup to install and use SQL Server Embedded Edition, you can significantly reduce the number of installation and configuration steps. If you have sufficient disk space, you should use the Embedded Edition that SharePoint provides.

If you do not want multiple instances of SQL Server on your computer, you can use the Database Engine for the SharePoint configuration and content databases. To do so, you must choose the **Advanced** installation option when you install a SharePoint product or technology.

## <sup>I</sup>Note:

The report server instance cannot use the Embedded Edition for its database. However, the Embedded Edition instance that is installed by the SharePoint product or technology can exist side-by-side other Database Engine editions that you might also install. For more information about edition requirements for the report server database, see <u>Creating a Report Server Database</u>.

## <sup>□</sup> Installation Order

You can install the component technologies in any order, but for best results you should install Reporting Services first and verify that you have a valid installation before you install the SharePoint product or technology.

To save steps, you should install both Reporting Services and the Database Engine at the same time. Because you are installing the Database Engine and Reporting Services together, you will most likely have the option of installing Reporting Services in the default configuration, where virtual directories, the service account, and report server databases are configured automatically during installation.

The following list summarizes the deployment tasks you must perform:

- 1. Install Reporting Services and Database Engine. If possible, choose the **Install the SharePoint mode default configuration** option when installing Reporting Services. Otherwise, you can select the **Install but do not configure** option and configure the server as a post-setup task.
- 2. Configure the report server if it is not configured already, and verify you have a valid installation.
- 3. Download and install the Microsoft .NET Framework 3.0 redistributable package.
- 4. Install Windows SharePoint Services or Office SharePoint Server 2007. Choose the **Basic** installation option.
- 5. Run the SharePoint Products and Technologies Configuration Wizard and then open the SharePoint site to verify you have a valid installation.
- 6. Download and install the Reporting Services Add-in for SharePoint Technologies.
- 7. Configure the report server integration features in Central Administration.

### <sup>©</sup> To install Reporting Services and the Database Engine

• Install Reporting Services and the Database Engine on the same computer. For more information, see <u>How to: Install SQL Server 2008</u>

<u>(Setup)</u>.

• Configure the report server, but when you create the report server database, select the option for SharePoint Integrated mode. For more information, see **How to: Configure a Report Server**.

After Setup is finished, you can verify that installation succeeded by connecting to the report server. You can test the connection by entering http://localhost/reportserver in the URL address of a browser window.

# <sup>□</sup> To download and install the .NET Framework 3.0 redistributable package

- 1. Download the <u>.NET Framework 3.0 redistributable package</u> from the Microsoft Download Center.
- 2. Run Setup. Setup runs in the background and might take awhile to finish. To check its progress, click the Microsoft .NET Framework 3.0 Installation icon in the notification area of the taskbar.

# <sup>©</sup> To install and configure Windows SharePoint Services or Office SharePoint Server

- 1. Run Setup.exe from the installation media.
- 2. Enter the 25-character product key.
- 3. Accept the software license agreement.
- 4. Click **Basic**.

This option provides the fewest installation steps. It immediately installs the software using all default values. It also automates the configuration steps for setting up Central Administration and the default SharePoint site.

If you want to use an existing Database Engine instance, you must click **Advanced** instead. When you choose this option, Setup provides the **Complete** installation option that you can use to specify which database server to use. Be aware that choosing the **Advanced** installation option increases the complexity of the configuration steps that you must perform later. For more information about how to specify and configure an **Advanced** installation, follow the procedure "To install a SharePoint product or technology" in <u>How to: Configure SharePoint Integration on Multiple Servers</u>. When you finish the **Advanced** installation and configuration steps in the multi-server deployment topic, return to this topic, change the port assignment and continue with the remaining tasks.

- 5. In the **Finished** page, verify that the check box is selected for **Run the SharePoint Products and Technologies Configuration Wizard**.
- 6. Click Close.

After Setup closes, the SharePoint Products and Technologies Configuration Wizard runs. When you select the **Basic** installation option, configuration tasks run in the background and you can click **Finish** when the configuration tasks are complete.

7. Verify that installation succeeded. By default, a SharePoint site will open in a browser window after the configuration tasks are complete. This step verifies the installation and configuration process.

By default, SharePoint Products and Technologies Configuration Wizard assigns port 80 to the SharePoint Web application, which will cause the report server to become unavailable. As a result, you must now modify report server configuration settings to use a different port and you must start the default Web site.

# <sup>©</sup> Verify the report server database is configured for SharePoint integrated mode

- 1. Start the Reporting Services Configuration tool and connect to the report server instance. For instructions, see <u>How to: Start Reporting</u> <u>Services Configuration</u>.
- 2. In the Report Server Status page, verify that **Report Server Mode** is set to **SharePoint integrated**.
- 3. If the report server is not configured for SharePoint integrated mode, click **Database** and create the report server database for SharePoint integrated mode. For instructions, see<u>How to: Create a Report Server Database (Reporting Services Configuration)</u> and <u>How to: Create a Report Server Database for SharePoint Integrated Mode (Reporting Services Configuration)</u>.

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 <u>Configuring the Unattended Execution</u> <u>Account and Configuring a Report Server for E-Mail Delivery</u>.

### <sup>©</sup> To download and install the Reporting Services Add-in for SharePoint Technologies

1. Click <u>Microsoft SQL Server</u> to open the download page on the Microsoft Web site.

#### <sup>I</sup>Important:

You must be a SharePoint Web farm administrator and Site Collection Administrator to install the Reporting Services Add-in.

- 2. Choose which language to install.
- 3. In **Files to Download**, click **Download** to install either the 32-bit or 64bit version of the Reporting Services Add-in.

- 4. Click **Save** to save the file to a location on your computer.
- 5. Open Windows Explorer and browse to the folder where the file is located.
- 6. Double-click **SharePointRS.msi** to begin installation. For more information, see <u>How to: Install or Uninstall the Reporting Services</u> <u>Add-in</u>.
- 7. Get the latest version of the readme file for the Reporting Services Addin to review late-breaking changes and known issues. On the download page on the Microsoft Web site, in **Files to Download**, click **Download** to open the **readme\_rsaddin.htm** file, and then save the file to your computer.

### 

1. Click **Start**, click **Administrative Tools**, and then click **SharePoint 3.0 Central Administration**.

### <sup>I</sup> Important:

To complete the following steps, you must be a member of the SharePoint Farm Administrator group and a Site Collection administrator.

2. Click **Application Management**. You should see a new section named Reporting Services. This section has links to pages used to specify integration settings. If you do not see this section, you must activate the report server integration feature. For more information, see <u>How to:</u> <u>Activate the Report Server Feature in SharePoint Central</u> <u>Administration</u>.

At a minimum, you must set the report server URL, select an authentication mode, and grant report server access to the SharePoint databases. For more information, see <u>How to: Configure Report Server</u> <u>Integration in SharePoint Central Administration</u>.

3. Close Central Administration.

## <sup>□</sup> Next Steps

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 resemble http://example-server-name:8080/reportserver. This URL is to the proxy 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.

After you install all the components and configure the settings for connecting server components, you must assign permissions, add content types, and add report server items to a SharePoint library.

#### To finish the deployment tasks for server integration

- 1. Open the SharePoint Web application. By default, the URL is http://<your-server-name>.
- 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 more information, see Granting Permissions on Report Server Items on a SharePoint Site or How to: Set Permissions for Report Server Items on a SharePoint Site (Reporting Services in SharePoint Integrated Mode).
- 3. Add Reporting Services content types if you want to create new shared data sources, report models, and Report Builder reports. For more information, 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 more information, 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 more information, 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 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. 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 Integrated Mode**.

## □ See Also

#### Tasks

How to: Configure SharePoint Integration on Multiple Servers

### Concepts

Configuring Reporting Services for SharePoint 3.0 Integration

#### **Other Resources**

Requirements for Running Reporting Services in SharePoint Integrated Mode Deployment Topologies for Reporting Services in SharePoint Integrated Mode Planning a Deployment Mode Reporting Services and SharePoint Technology Integration

© 2008 Microsoft Corporation. All rights reserved.
#### SQL Server Setup Help

How to: Configure SharePoint Integration on Multiple Servers



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

This topic provides step-by-step instructions for installing and configuring Reporting Services in SharePoint integrated mode in a multi-server deployment topology. This configuration is recommended if you have a large number of users who access reports from a SharePoint site. The following illustration shows the components you must install.



## <sup>©</sup> Prerequisites

Before you begin installing components, be sure to do the following:

- Review the introduction to deployment configurations so that you are familiar with the components that you are about to install. For more information, click **Deployment Topologies for Reporting Services in SharePoint Integrated Mode**.
- Review the configuration overview topic for best practice recommendations on service account configuration and to learn more about the configuration steps. For more information, click <u>Configuring Reporting Services for SharePoint 3.0 Integration</u>.
- Review the hardware and software requirements that apply to all deployment scenarios that include Reporting Services in SharePoint integrated mode. For more information, click **Requirements for Running Reporting Services in SharePoint Integrated Mode**.
- Review the remaining items in this section to learn about additional requirements that apply to a stand-alone server configuration.

## Server Requirements for a Multi-server Deployment

You must have at least two or three servers. Hosting the application databases on a third server that is used exclusively by the Database Engine is common technique for improving performance if you expect a relatively large number of users.

If you are using two servers, each one should have the following components:

- On one server, install SQL Server 2008 Reporting Services (SSRS) and Database Engine. This server will be used to host the report server along with the report server database and SharePoint application databases.
- On the second server, install the Microsoft .NET Framework 3.0 redistributable package, an instance of a SharePoint product or technology, and the Reporting Services Add-in. This server will be the primary server in the farm. Central Administration will be installed on this server. Users will access report server content from the site collection on this computer.

#### Account Requirements for a Multi-server Deployment

You must have a domain user account to connect the instance of the SharePoint product or technology to the SharePoint\_config database. Server farms require that you use domain accounts for services and database connections. If possible, create a new account that can be used for just this application. Avoid using your own credentials or those of any user who will be accessing a SharePoint site as a regular user (you will get access denied errors).

You must create a SQL Server database login for the domain account. The login needs **DBCreator** permissions on the server.

You must configure the SharePoint application pool process account to run as a domain user.

You must configure the Report Server service to run as a domain user account.

#### <sup>I</sup>Important:

For distributed server deployments where the report server, the SharePoint Web application, and the server databases are on three 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 encounter connection errors unless you follow the guidelines about service account configuration described in the previous section. For more information, see <u>How to: Configure a Service Account for Reporting Services</u>.

Administrator credentials are required to run all Setup programs, to grant permissions to be the primary site collection administration, and to grant database access when you configure report server integration.

## <sup>□</sup> Installation Order

You can install the component technologies in any order. For best results, confirm that each server product or technology is installed correctly before you integrate them

The following list provides a summary of the deployment tasks:

- 1. Install Reporting Services and Database Engine. If you are installing server software on a production server or if you are using a three-server deployment configuration, plan on choosing the **Install but do not configure the server** option so that you can control when and how the report server is configured.
- 2. Configure the report server, and verify you have a valid installation.
- 3. Download and install the .NET Framework 3.0 redistributable package.
- 4. Install Windows SharePoint Services or Office SharePoint Server 2007. Plan on choosing the **Advanced** installation option so that you can specify a remote database server during configuration.
- 5. Run the SharePoint Products and Technologies Configuration Wizard and select the database server.
- 6. Configure the site collection features in Central Administration and open the SharePoint site to verify you have a valid installation.
- 7. Download and install the Reporting Services Add-in for SharePoint Technologies.
- 8. Configure the report server integration features in Central Administration.

#### <sup>©</sup> To install Reporting Services and the Database Engine

• Install Reporting Services and the Database Engine on the same computer or on different computers. For more information, see <u>Considerations for Installing Reporting Services</u>.

After Setup is finished, configure the report server for native mode operation to verify that the installation succeeded. Doing so allows you to use Report Manager to verify the installation.

For instructions on how to start the Reporting Services Configuration tool and configure the report server for operation, see <u>Reporting Services Configuration</u> <u>How-to Topics</u>.

## <sup>©</sup> To download and install the .NET Framework 3.0 redistributable package

- 1. On a different server that you will use for the instance of the SharePoint product or technology, download the <u>.NET Framework 3.0</u> redistributable package from the Microsoft Download Center.
- 2. Run Setup. Setup runs in the background and might take awhile to finish. To check its progress, click the Microsoft .NET Framework 3.0 Installation icon in the notification area of the taskbar.

## <sup>□</sup> To install a SharePoint product or technology

- 1. Run Setup.exe from the installation media for Windows SharePoint Services 3.0 or Microsoft Office SharePoint Server .
- 2. Enter the 25-character product key.
- 3. Accept the software license agreement.
- 4. Click Advanced.
  - 1. On Server Type, click Web Front End
  - 2. The Data Location tab specifies file location for search index files. On **Data Location**, use the default value or select a computer that has excess disk space.
  - 3. Click Install Now.
- Configure the SharePoint product or technology. The SharePoint Product and Technology Configuration Wizard should start after

Setup is finished. If it does not, start it from Administrative Tools.

- 1. Click **Next** on the Welcome screen.
- 2. Click **Yes** to restart the services if required.
- 3. On the **Connect to a Server Farm** page, click **No, I want to create a new server farm**, and then click **Next**.
- 4. On the **Specify Configuration Database Settings** page, enter the database server name. If you installed the Database Engine as a default instance, the database server name is the computer name.
- 5. In **User Name** and **Password**, type the credentials of a domain user account that has permission to log on to the computer, a SQL Server database login, and **DBCreator** permissions on the server. If possible, specify an account that is used only to run this application. Do not type the credentials of a user who will later access the SharePoint site; otherwise, that user might receive access denied errors later when accessing content on the site. Click **Next**.
- 6. On the **Configure SharePoint Central Administration** page, click **Next** to accept all the defaults. If you know that impersonation and delegation features are enabled for Windows Authentication in your domain, you can select **Negotiate (Kerberos)** instead. Otherwise, select the default value and click **Next**.
- 7. On **Completing the SharePoint Products and Technologies Configuration Wizard**, click **Next** to start the configuration using the settings that you specified.
- 8. Click Finish.

## <sup>©</sup> To configure the SharePoint Web application

1. Click **Start**, click **Administrative Tools**, and then click **SharePoint 3.0 Central Administration**. Follow the checklist in the Administrator's Tasks. For more information about each task, click the link **Read the deployment steps** and then open the **Quick Start Guide**. The tasks in the following steps are required.

- 2. Create a Web application:
  - 1. Click **Create SharePoint Sites**.
  - 2. Click **Create new Web application**.
  - 3. Select **Create a new IIS web site (SharePoint 80)** using the default values. This includes using port 80, no host headers, default path, NTLM, and no Anonymous access. Be sure to specify the application pool process account. Choose a domain user account.
  - 4. Click **OK**.
- 3. Click **Create Site Collection** on the **Application Created** page.
  - 1. Enter a title.
  - 2. Use all default values. When specifying Primary Site Collection Administrator, specify an account that has local administrator permissions.
  - 3. Click **OK**.
- 4. Open a Command Prompt window and type **iisrest** /**noforce**.
- 5. Click the URL to verify the site.
- 6. (Optional) In Central Administration, enable diagnostic logging.

#### <sup>©</sup> To configure the report server for SharePoint integrated mode

- 1. Install and configure a SharePoint Web front end on the report server computer. For instructions, see <u>How to: Install a SharePoint Web Front-end on a Report Server Computer</u>.
  - 1. A browser window will open to Central Administration on the primary server, allowing you to verify that the server farm configuration succeeded.
- 2. Configure the report server for integration:
  - 1. Start the Reporting Services Configuration tool and connect to the

report server instance. For instructions, see <u>How to: Start</u> <u>Reporting Services Configuration</u>.

- 2. In the navigation pane, click **Service Account**. Verify that it is using a domain user account. If it is not, configure it to use one.
- 3. In the navigation pane, click **Database**.

<sup>II</sup>Note:

If you are using a remote Database Engine instance, see <u>How to:</u> <u>Create a Report Server Database for SharePoint Integrated Mode</u> <u>(Reporting Services Configuration)</u> for more information.

- 4. Click **Change Database**.
- 5. Select **Create a new database**.
- 6. Click Next.
- 7. Specify the instance of the Database Engine that will host the database.
- 8. Click **Test Connection** to verify that you can connect to it.
- 9. Click Next.
- 10. In **Database Name**, type a name for the new database (for example, **reportserverSharepointMode**).
- 11. Select **SharePoint Integrated Mode**.
- 12. Click Next.
- 13. Specify credentials that the report server uses to connect to the report server database.
- 14. Click Next.
- 15. Click Next.
- 16. Click Finish.
- 17. Configure the unattended execution account and any other settings that you want to change. The values for the Report Server virtual directory should still be valid. However, Report Manager will no longer be available now that you have configured the report server to use a SharePoint integrated mode database (the tab for the Report Manager virtual directory should be shaded).

## <sup>©</sup> To download and install the Reporting Services Add-in

1. Click <u>Microsoft SQL Server 200</u> to open the download page on the Microsoft Web site.

#### <sup>I</sup>Important:

You must be a SharePoint Web farm administrator and Site Collection Administrator to install the Reporting Services Add-in.

- 2. Choose which language to install.
- 3. In **Files to Download**, click **Download** to install either the 32-bit or 64bit version of the Reporting Services Add-in.
- 4. Click **Save** to save the file to a location on your computer.
- 5. Open Windows Explorer and browse to the folder where the file is located.
- 6. Double-click **SharePointRS.msi** to begin installation. For instructions, see <u>How to: Install or Uninstall the Reporting Services Add-in</u>.
- 7. Get the latest version of the readme file for the Reporting Services Addin to review late-breaking changes and known issues. On the download page on the Microsoft Web site, in **Files to Download**, click **Download** to open the **readme\_rsaddin.htm** file, and then save the file to your computer.

#### <sup>□</sup> To configure server integration in Central Administration

1. Click **Start**, click **Administrative Tools**, and then click **SharePoint 3.0 Central Administration**.

#### <sup>I</sup> Important:

To complete the following steps, you must be a member of the SharePoint Farm Administrator group and a Site Collection administrator.

2. Click **Application Management**. You should see a new section named Reporting Services. This section has links to pages used to specify integration settings. If you do not see this section, you must activate the report server integration feature. For more information, see <u>How to:</u> <u>Activate the Report Server Feature in SharePoint Central</u> <u>Administration</u>.

At a minimum, you must set the report server URL, select an authentication mode, and grant report server access to the SharePoint databases. For more information, see <u>How to: Configure Report Server</u> <u>Integration in SharePoint Central Administration</u>.

3. Close Central Administration.

You will get an error if the Reporting Services services are running under builtin accounts. If you already configured the services to use domain user accounts, restart the services to clear out old connection information (assuming that there are existing connections in the current session). For more information about service account configuration, see <u>Configuring the Report Server Service</u> <u>Account</u>.

## <sup>□</sup> Next Steps

After you install all the components and configure the settings for connecting server components, you must assign permissions, add content types, and add report server items to a SharePoint library.

#### <sup>©</sup> To finish the deployment tasks for server integration

- 1. Open the SharePoint Web application. By default, the URL is http://<your-server-name>.
- 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 more information, see Granting Permissions on Report Server Items on a SharePoint Site or How to: Set Permissions for Report Server Items on a SharePoint Site (Reporting Services in SharePoint Integrated Mode).
- 3. Add Reporting Services content types if you want to create new shared data sources, report models, and Report Builder reports. For more information, 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 more information, see **How to: Create and Manage Shared Data Sources** (Reporting Services in SharePoint Integrated Mode).
- Create shared schedules to use with subscriptions or unattended report processing. For more information, 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 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. 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. For a complete list of supported and unsupported features, see **Features Supported by Reporting Services in SharePoint Integrated Mode**.

## □ See Also

#### Tasks

How to: Configure SharePoint Integration on a Stand-alone Server

#### Concepts

Configuring Reporting Services for SharePoint 3.0 Integration Considerations for Installing Reporting Services Configuring the Unattended Execution Account Configuring a Report Server for E-Mail Delivery

#### **Other Resources**

Requirements for Running Reporting Services in SharePoint Integrated Mode Deployment Topologies for Reporting Services in SharePoint Integrated Mode Planning a Deployment Mode Reporting Services and SharePoint Technology Integration

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help How to: Create a Report Server Database (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

Reporting Services uses a SQL Server database for internal storage. The database is a required component; it is used to store published reports, models, shared data sources, session data, resources, and server metadata.

To create a report server database or to change the connection string or credentials, use the options in the Database page in the Reporting Services Configuration tool.

# □ When to Create or Configure the Report Server Databases

You must create and configure the report server database if you installed the report server in files-only mode.

If you installed Reporting Services in either the default configuration for native mode or SharePoint integrated mode, the report server database was created and configured automatically when the report server instance was installed. You can use the Reporting Services Configuration tool to view or modify the settings that Setup configured for you.

## <sup>□</sup> Before You Start

Creating or configuring a report server database is a multi-step process. Before you create the report server database, consider how you want to specify the following items:

Select a database server

Reporting Services requires a SQL Server 2005 or 2008 Database Engine instance to host the report server database. You can use a local or remote server.

Check edition requirements

Verify that the edition of the Database Engine is compatible with the report server edition you are using. For more information about edition compatibility, see <u>Creating a Report Server Database</u>.

Enable TCP/IP connections

Enable TCP/IP connections for the Database Engine. Some editions of the Database Engine do not enable TCP/IP by default. Instructions are provided in this topic.

Open port for SQL Server

For a remote server, if you are using firewall software, you must open the port that the Database Engine listens on.

Decide on a mode

A report server database support native mode or SharePoint integrated mode. **Native mode** is the default. Use this mode when you want to deploy Reporting Services as a stand-alone application server.

**SharePoint integrated mode** is used when you plan to deploy the report server together with an instance of a SharePoint product or technology. The database will be created when you run the wizard, but you will not be able to add content or view items until you add the report server to a SharePoint

farm. For more information, see **Planning a Deployment Mode**.

Decide on report server credentials

Decide how the report server will connect to the report server databases. Credential types include domain user account, SQL Server database user account, or the Report Server service account.

These credentials are encrypted and stored in the RSReportServer.config file. The report server uses these credentials for ongoing connections to the report server database. If you want to use a Windows user account or a database user account, be sure to specify one that already exists. Although the Reporting Services Configuration tool will create a login and set the necessary permissions, it will not create an account for you. For more information, see <u>Configuring a Report Server Database Connection</u>.

Decide on a report server language

Choose a language to specify for the report server. Predefined role names, descriptions, and the My Reports folders do not appear in different languages when users connect to the server using different language versions of a browser. For more information about how language resources are used in a report server deployment, see **Solution Design Considerations for Multi-Lingual or Global Deployments (Reporting Services)**.

Check credentials to create and provision the database

Verify that you have account credentials that have permission to create databases on the Database Engine instance. These credentials are used for a one-time connection to create the report server database and **RSExecRole**. If a login does not already exist, a database user login will be created for the account used by the report server to connect to the database. You can connect under the Microsoft Windows account you are logged in as, or you can enter a SQL Server database login.

## $^{\Box}$ To enable access to a remote report server database

- 1. If you are using a remote Database Engine instance, log on to the database server to verify or enable TCP/IP connections.
- 2. Point to **Start**, point to **All Programs**, point to **Microsoft SQL Server**, point to **Configuration Tools**, and click **SQL Server Configuration Manager**.
- 3. Open SQL Server Network Configuration.
- 4. Select the instance.
- 5. Right-click **TCP/IP** and click **Enabled**.
- 6. Restart the service.
- 7. Open your firewall software and open the port that SQL Server listens on. For the default instance, this is typically port 1433 for TCP/IP connections. For more information, see **How to: Configure a Windows Firewall for Database Engine Access**.

#### <sup>□</sup> To create a local 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, see <u>How to: Start Reporting Services Configuration</u>.
- 2. On the Database page, click **Change Database**.
- 3. Click **Create a new database**, and then click **Next**.
- 4. Connect to the instance of the Database Engine that you will use to create and host the report server database:
  - 1. Type the SQL Server Database Engine instance that you want to use. The wizard will display a local Database Engine that runs as the default instance if it is available. Otherwise, you must type the server and instance to use. Named instances are specified in this

format: <servername>\<instancename>.

- 2. Enter the credentials used for a one-time connection to the Database Engine for the purpose of creating the report server databases. For more information about how these credentials are used, see <u>Before You Start</u> in this topic.
- 3. Click **Test Connection** to validate the connection to the server.
- 4. Click **Next**.
- 5. Specify properties used to create the database. For more information about how these properties are used, see <u>Before You Start</u> in this topic:
  - 1. Type the name of the report server database. A temporary database is created along with the primary database. Consider using a descriptive name to help you remember how the database is used. Note that the name you specify will be used for the lifetime of the database. You cannot rename a report server database after it is created.
  - 2. Select the language in which you want role definitions and My Reports to appear.
  - 3. Select the server mode. Always select **Native Mode** unless you are planning to deploy the report server with a SharePoint product or technology.
  - 4. Click Next.
- 6. Specify the credentials used by the report server to connect to the report server database.
  - 1. Specify the authentication type:

Select **Database Credentials** to connect using a SQL Server database login that is already defined. Using database credentials is recommended if the report server is on a computer that is in a different domain, a non-trusted domain, or behind a firewall. Select **Windows Credentials** if you have a least-privileged domain user account that has permission to log on to the computer and the database server.

Select **Service Credentials** if you want the report server to connect using its service account. With this option, the server connects using integrated security; credentials are not encrypted or stored.

2. Click Next.

- 7. Review the information on the Summary page to verify the settings are correct, and then click **Next**.
- 8. Verify the connection by clicking a URL on the Report Server URL page or Report Manager URL page. The URLs must be defined in order for this test to work. If the report server database connection is valid, you will see either the report server folder hierarchy or Report Manager in a browser window. For more information, see <u>How to: Verify a</u> <u>Reporting Services Installation</u>.

## □ See Also

#### Tasks

How to: Start Reporting Services Configuration

#### Reference

Setting Up Windows Service Accounts Database (Reporting Services Configuration)

#### Concepts

Configuring a Report Server Database Connection Reporting Services Configuration How-to Topics

#### **Other Resources**

Administering a Report Server Database Report Server Database Configuring a Report Server Installation Reporting Services Configuration Tool

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help

How to: Create a Report Server Database for SharePoint Integrated Mode (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

Use the Reporting Services Configuration tool 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**.

Key points to remember about a report server database that supports SharePoint integrations include the following:

• Use the Change Database Wizard in the Database page of the Reporting Services Configuration tool to create the report server database in SharePoint integrated mode. For instructions, see <u>How to: Create a</u> <u>Report Server Database (Reporting Services Configuration)</u>.

- Use the Reporting Services Configuration tool to maintain or update the connection information to the report server database.
- Use SQL Server Management Studio and SQL Server tools to back up and administer a report server database. The database backup features in SharePoint 3.0 Central Administration do not apply to a report server database.
- Creating the report server database is only one step in a multi-step process that configures a report server for integration operations. 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 <u>How to: Configure Report</u> <u>Server Integration in SharePoint Central Administration</u>.

## □ See Also

#### Tasks

How to: Switch Server Modes (Reporting Services Configuration)

#### Concepts

<u>Configuring Reporting Services for SharePoint 3.0 Integration</u> <u>Configuring a Report Server Database Connection</u>

#### **Other Resources**

Planning a Deployment Mode Administering a Report Server Database

© 2008 Microsoft Corporation. All rights reserved.

SQL Server Setup Help How to: Enable Remote Errors (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

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.

#### <sup>I</sup>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 SQL Server Management Studio or script to modify the settings.

## <sup>□</sup> Enable remote errors through SQL Server Management Studio

- 1. Start Management Studio and connect to a report server instance.
- 2. Right-click the report server node, and select **Properties**.
- 3. Click **Advanced** to open the properties page.
- 4. In **EnableRemoteErrors**, select **True**.
- 5. Click **OK**.

## <sup>©</sup> 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 Coc** rs -i EnableRemoteErrors.rss -s http://servername/ReportServ€

## □ See Also

**Other Resources** 

Report Server System Properties How to: Connect to a Report Server in Management Studio Report Server System Properties Server Properties (Advanced Page) - Reporting Services

© 2008 Microsoft Corporation. All rights reserved.

#### SQL Server Setup Help

How to: Install a SharePoint Web Front-end on a Report Server Computer



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

If you are configuring a report server to run in SharePoint integrated mode in a SharePoint farm that consists of multiple servers, you must install the Web Front End application of a SharePoint product or technology on the report server computer so that you can join the report server computer to the farm.

A SharePoint farm consists of multiple servers that all use the same version of the SharePoint product or technology. If the server farm uses Windows SharePoint Services 3.0, you must install Windows SharePoint Services 3.0 on the report server computer. If the server farm uses Microsoft Office SharePoint Portal Server farm, you must install an Office SharePoint Portal Server.

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 server farm.

The following steps are required only if you are deploying a report server in a SharePoint server farm. If all server applications run on a single computer, installing the Web front end on the report server is not required.

After you install the Web front end and join the report server computer to the farm, you can use SharePoint 3.0 Central Administration to turn off any Web front end features that you do not require.

## $^{\square}$ To install the Web front end

1. On the server that runs Reporting Services, run Setup for the SharePoint product or technology used for the farm.

- 2. On Installation Types, click **Advanced**.
- 3. On Server Type, select **Web Front End**.
- 4. The Data Location tab specifies file location for search index files. On **Data Location**, use the default value or choose a computer that has excess disk space.
- 5. Click **Install Now**.

After Setup is finished, you must configure the Web front end to join it to the server farm.

#### <sup>□</sup> To configure the Web front end

- Run the SharePoint Products and Technologies Configuration Wizard to join the Web front end to the remote SharePoint farm. The SharePoint Product and Technology Configuration Wizard should start after Setup is finished. If it does not, start it from Administrative Tools.
- 2. Click **Next** on the Welcome screen.
- 3. Click **Yes** to restart the services if required.
- 4. In the Connect to a Server Farm page, select **Yes** to connect to an existing server farm. Click **Next**.
- 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.
- 6. Enter a user name and password that the Web front end will use to connect to the configuration database. Click **Next**.
- 7. In the Completing the SharePoint Products and Technologies Wizard page, click **Advanced Settings**.

- 8. In Advanced Settings, verify that **Do not use this machine to host the Web site** is selected.
- 9. 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 <u>Configuring Reporting</u> <u>Services for SharePoint 3.0 Integration</u>.

For more information about configuring a multi-server deployment that integrates Reporting Services and SharePoint products and technologies, see <u>How to: Configure SharePoint Integration on Multiple Servers</u>.

## □ See Also

#### Tasks

How to: Configure SharePoint Integration on Multiple Servers

#### Concepts

How to: Install or Uninstall the Reporting Services Add-in Configuring Reporting Services for SharePoint 3.0 Integration

#### **Other Resources**

Reporting Services and SharePoint Technology Integration Planning a Deployment Mode Requirements for Running Reporting Services in SharePoint Integrated Mode

© 2008 Microsoft Corporation. All rights reserved.

## SQL Server Setup Help How to: Install or Uninstall the Reporting Services Add-in



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

The Microsoft SQL Server 2008 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 URL proxy endpoint, and application pages so that you can create, view, and manage reports, report models, and other report server content on a SharePoint site. The Reporting Services Add-in for SharePoint Technologies is a required component for a report server that runs in SharePoint integrated mode.

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.

## <sup>©</sup> 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 <u>Configuring Reporting Services for SharePoint 3.0 Integration</u>. 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. Log on as a SharePoint farm administrator to install the Reporting Services Add-in.
- 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.
- 3. Download the Reporting Services Add-in to a server that has an instance of the SharePoint product or technology.
  - 1. Go to the <u>SQL Server 2008 Reporting Services Add-in for SharePoint</u> <u>Technologies</u> page on the <u>Microsoft Download Center</u>.
  - 2. Select the language version you want to install. The Report Viewer Web Part (ReportViewer.dwp) will be available only in the language that you choose. The application pages and messages will be available in all supported languages, regardless of the language you select on the Download page. For more information about supported languages, see **Solution Design Considerations for Multi-Lingual or Global Deployments (Reporting Services)**.
  - 3. In the **Files in this Download** section, click **Download** to install the **SharepointRS.msi**.
- 4. If you are integrating Reporting Services with a SharePoint farm that has multiple Web front end applications, copy or download the **SharepointRS.msi** to each computer in the farm that has a Web server front-end. Do this only for Web front ends that will be used to access report server content.

# □ How to Install Reporting Services Add-in Using the Installation Wizard

Run **SharepointRS.msi** 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. You can use the Installation Wizard to install the Reporting Services Add-in, or run Setup from the command line and specify command line arguments for a files-only mode installation.

Setup does not configure the server for integration with Reporting Services. You must configure integration after Setup is complete. Setup uses the existing folder structure of the current deployment and creates a new folder for the Reporting Services Add-in application files.

- 1. Navigate to the folder that contains **SharePointRS.msi**.
- 2. Double-click **SharePointRS.msi** to run the Installation Wizard. The wizard displays a Welcome page, the license agreement, and a registration information page.
- 3. Click **Install** to run Setup.
- 4. After Setup is finished, go to Administrator Tools and click **SharePoint 3.0 Central Administration**.
- 5. 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 <u>How to: Configure Report Server Integration in SharePoint Central Administration</u>.

## <sup>□</sup> How to Install Reporting Services Add-in in Files-Only Mode

You can install the Reporting Services Add-in in files-only mode if you encountered errors during a normal Setup. In files-only mode, Setup will run the Installation Wizard to copy the application files to your computer. After the files are copied, you must run custom actions to complete the installation.

- 1. Run SharePointRS.msi SKIPCA=1.
- 2. Find the rsCustomAction.exe file on the file system. This file is copied to your computer by the Setup program. The file will be located in the %Temp% directory.

To get the path information for this file, click **Start**, click **Run**, and then type **%temp%**. The file should be located in \Documents and Settings\ <*your name*>\LOCALS~1\Temp.

- 3. Open a command prompt window. To do this, click **Start**, click **Run**, and type **cmd**.
- 4. Navigate to the folder that contains the rsCustomAction executable.
- 5. Enter the following command:

Copy Code

rsCustomAction.exe /i

This configuration step will take several minutes to complete. The W3SVC service will be restarted during this process.

- 6. After Setup is finished, go to Administrator Tools and click **SharePoint 3.0 Central Administration**.
- 7. 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 <u>How to: Configure Report Server Integration in SharePoint Central Administration</u>.

## <sup>□</sup> How to Remove the Reporting Services Add-in

You can uninstall the Reporting Services Add-in for SharePoint Technologies by running the Setup program (SharepointRS.msi)) in uninstall mode. Uninstalling the add-in will remove server integration features that are used to process reports and models on a report server.

To uninstall the Reporting Services Add-in, you must have a Windows SharePoint Services or Office SharePoint Server 2007 installation still running. If you uninstall the SharePoint product or technology first, you must reinstall it to uninstall the Reporting Services Add-in.

- 1. On the SharePoint site, 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.

#### Click **Remove**.

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 service account used to access the SharePoint configuration and content databases. You must delete any logins for the Report Server service account 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.

# <sup>□</sup> How to Run SharePointRS.msi from the Command Line

You can run Setup from the command line to repair or uninstall the Reporting Services Add-in.

- Run **SharePointRS.msi -r** to repair an existing installation.
- Run **SharePointRS.msi -u** to uninstall an existing installation of the Addin.

# <sup>©</sup> 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

<u>How to: Switch Server Modes (Reporting Services Configuration)</u> <u>How to: Install a SharePoint Web Front-end on a Report Server Computer</u>

## Concepts

Configuring Reporting Services for SharePoint 3.0 Integration

## **Other Resources**

Viewing and Managing Report Server Items from a SharePoint Site Planning a Deployment Mode Requirements for Running Reporting Services in SharePoint Integrated Mode Security Overview for Reporting Services in SharePoint Integrated Mode Granting Permissions on Report Server Items on a SharePoint Site

SQL Server Setup Help

How to: Register a Service Principal Name (SPN) for a Report Server



### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

If you are deploying Reporting Services in a network that uses the Kerberos protocol for mutual authentication, you must create a Service Principal Name (SPN) for the Report Server service if you configure it to run as a domain user account. Additionally, when you configure a domain user as the service account, the report server must be configured to use NTLM authentication.

# <sup>□</sup> About SPNs

An SPN is a unique identifier for a service on a network that uses Kerberos authentication. It consists of a service class, a host name, and a port. On a network that uses Kerberos authentication, an SPN for the server must be registered under either a built-in computer account (such as NetworkService or LocalSystem) or user account. SPNs are registered for built-in accounts automatically. However, when you run a service under a domain user account, you must manually register the SPN for the account you want to use.

To create an SPN, you can use the **SetSPN** command line utility. For more information, see <u>Setspn Overview</u> on the Microsoft Windows Server TechCenter.

You must be a domain administrator to run the utility on the domain controller.

## <sup>□</sup> Syntax

The command syntax for using SetSPN utility to create an SPN for the report server resembles the following:

Setspn -a http/<computername>.<domainname>:<port> <domain-user-accou

**SetSPN** is available with Windows Server 2003 Service Pack 1 support tool pack, which you must download separately. The **-a** argument is used to register a service principal name with a particular account.

**HTTP** is the service class. The Report Server Web service runs in HTTP.SYS. A by-product of creating an SPN for HTTP is that all Web applications on the same computer that run in HTTP.SYS (including applications hosted in IIS) will be granted tickets based on the domain user account. If those services run under a different account, the authentication requests will fail. To avoid this problem, be sure to configure all HTTP applications to run under the same account, or consider creating host headers for each application and then creating separate SPNs for each host header. When you configure host headers, DNS changes are required regardless of the Reporting Services configuration.

The values that you specify for *<computername>*, *<domainname>*, and *<port>* identify the unique network address of the computer that hosts the report server. This can be a local host name or a fully qualified domain name (FQDN). If you only have one domain and are using port 80, you can omit *<domainname>* and *<port>* from your command line. *<domain-user-account>* is the user account under which the Report Server service runs and for which the SPN must be registered.

# <sup>□</sup> Register an SPN for Domain User Account

# <sup>□</sup> To register an SPN for a Report Server service running as a domain user

1.

Install Reporting Services and configure the Report Server service to run as a domain user account. Note that users will not be able to connect to the report server until you complete the following steps.

• Download and install <u>Windows Server 2003 Service Pack 1 Support Tools</u> from the Download Center.

- Log on to the domain controller as domain administrator.
- Open a Command Prompt window.

• Copy the following command, replacing placeholder values with actual values that are valid for your network:

```
Setspn -a http/<computer-name>.<domain-name>:<port> <domain-user-act
```

• Run the command.

• Open the RsReportServer.config file and locate the <AuthenticationTypes> section. Add <RSWindowsNegotiate/> as the first entry in this section to enable NTLM.

## □ See Also

## Tasks

How to: Configure a Service Account for Reporting Services

## Concepts

Configuring the Report Server Service Account Reporting Services Configuration How-to Topics Considerations for Installing Reporting Services

#### **Other Resources**

**Planning for Services, Accounts, and Connections** 

#### SQL Server Setup Help

How to: Switch Server Modes (Reporting Services Configuration)

# 

#### - ▼

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

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.

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 **Use the schema that supports 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 <u>Configuring Reporting Services for SharePoint 3.0 Integration</u>.

## <sup>□</sup> To switch to native mode

- 1. Click **Start**, point to **Programs**, point to **Microsoft SQL Server 2008**, 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 Reporting Services instance that you want to configure.
- 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: Configure Report Server Integration in SharePoint Central Administration

## Concepts

Configuring Reporting Services for SharePoint 3.0 Integration How to: Install or Uninstall the Reporting Services Add-in Configuring a Report Server Database Connection How to: Create a Report Server Database for SharePoint Integrated Mode (Reporting Services Configuration)

## **Other Resources**

#### **Planning a Deployment Mode**

## SQL Server Setup Help How to: Turn Reporting Services Features On or Off



### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration</u> <u>How-to Topics</u> >

You can turn off report server features that you do not use as part of a lockdown strategy for reducing the attack surface of a production report server. In most cases, you will want to run Reporting Services features concurrently to use all of the functionality provided in Reporting Services. However, depending on your deployment model, you can disable the features that you do not require. For example, you can enable only the background processing 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.

The procedures in this topic show you how to turn off Reporting Services features. Features can be configured in different ways, such as by editing the RsReportServer.config file directly or by using the **Surface Area Configuration for Reporting Services** facet of Policy-Based Management in SQL Server Management Studio. Use the links to locate the procedure or procedures that explain how to turn a feature on or off:

- <u>Report Server Web service</u>
- <u>Scheduled events and processing</u>
- <u>Report Manager</u>
- <u>Report Builder</u>
- Windows Integrated security for report data sources

## <sup>□</sup> Report Server Web Service

# <sup>□</sup> To turn on or off the Report Server Web service by editing configuration

1.

Open the RsReportServer.config file in a text editor. For more information, see **How to: Modify a Reporting Services Configuration File**.

• To turn on the Report Server Web service, set **IsWebServiceEnabled** to **true**:

<IsWebServiceEnabled>true</IsWebServiceEnabled>

• To turn off the Report Server Web service, set **IsWebServiceEnabled** to **false**:

Copy Code

Copy Code

<IsWebServiceEnabled>false</IsWebServiceEnabled>

• Save your changes and then close the file.

# <sup>©</sup> To turn on or off the Report Server Web service by using SQL Server Management Studio

- 1. Open SQL Server Management Studio and connect to the Reporting Services instance that you want to configure.
- 2. In Object Explorer, right-click the Reporting Services node, point to **Policies**, and click **Facets**.
- 3. In the Facet list, select Surface Area Configuration for Reporting

Services.

- 4. Under Facet Properties:
  - To turn on the Report Server Web service, set **WebServiceAndHTTPAccessEnabled** to **True**.
  - To turn off the Report Server Web service, set **WebServiceAndHTTPAccessEnabled** to **False**.
- 5. Click **OK**.

# <sup>©</sup> Scheduled Events and Delivery

# $^{\mbox{\tiny \Box}}$ To turn on or off scheduled events and delivery by editing configuration

1.

Open the RsReportServer.config file in a text editor. For more information, see **How to: Modify a Reporting Services Configuration File**.

• To turn on scheduled report processing and delivery, set **IsSchedulingService**, **IsNotificationService**, and **IsEventService** to **true**:

```
Copy Code
```

```
<IsSchedulingService>true<\IsSchedulingService>
<IsNotificationService>true<\IsNotificationService>
<IsEventService>true<\IsEventService>
```

• To turn off scheduled report processing and delivery, set **IsSchedulingService**, **IsNotificationService**, and **IsEventService** to **false**:

Copy Code

```
<IsSchedulingService>false<\IsSchedulingService>
<IsNotificationService>false<\IsNotificationService>
<IsEventService>false<\IsEventService>
```

• Save your changes and then close the file.

## <sup>I</sup>Note:

You cannot turn off background processing completely because it provides database maintenance functionality that is required for server operations.

# <sup>□</sup> To turn on or off scheduled events and delivery by using SQL Server Management Studio

- 1. Open SQL Server Management Studio and connect to the Reporting Services instance that you want to configure.
- 2. In Object Explorer, right-click the Reporting Services node, point to **Policies**, and click **Facets**.
- 3. In the **Facet** list, select **Surface Area Configuration for Reporting Services**.
- 4. Under Facet Properties:
  - To turn on scheduled events and delivery, set ScheduleEventsAndReportDeliveryEnabled to True.
  - To turn off scheduled events and delivery, set
     ScheduleEventsAndReportDeliveryEnabled to False.
- 5. Click **OK**.

## <sup>I</sup>Note:

You cannot turn off background processing completely because it provides database maintenance functionality that is required for server operations.

# <sup>□</sup> Report Manager

## <sup>□</sup> To turn on or off Report Manager by editing configuration

1.

Open the RsReportServer.config file in a text editor. For instructions, see **How to: Modify a Reporting Services Configuration File**.

• To turn on Report Manager, set **IsReportManagerEnabled** to **true**:

Copy Code

<IsReportManagerEnabled>true<\IsReportManagerEnabled>

• To turn off Report Manager, set **IsReportManagerEnabled** to **false**:

Copy Code

<IsReportManagerEnabled>**false**<\IsReportManagerEnabled>

• Save your changes and then close the file.

## <sup>□</sup> To turn on or off Report Manager by using SQL Server Management Studio

- 1. Open SQL Server Management Studio and connect to the Reporting Services instance that you want to configure.
- 2. In **Object Explorer**, right-click the Reporting Services node, point to **Policies**, and click **Facets**.
- 3. In the **Facet** list, select **Surface Area Configuration for Reporting Services**.

## 4. Under **Facet Properties**:

- To turn on Report Manager, set **ReportManagerEnabled** to **True**.
- To turn off Report Manager, set **ReportManagerEnabled** to **False**.
- 5. Click **OK**.

# <sup>□</sup> Report Builder

## <sup>□</sup> To turn on or off Report Builder by using SQL Server Management Studio

1.

Open SQL Server Management Studio and connect to the Reporting Services instance that you want to configure.

• In Object Explorer, right-click the Reporting Services node, and click **Properties**.

- In the **Server Properties** dialog box, under **Select a page**, click **Security**.
  - To turn on Report Builder, select the **Enable ad hoc report executions** option.
  - To turn off Report Builder, unselect the **Enable ad hoc report executions** option.
- Click **OK**.

# <sup>□</sup> Windows Integrated Security

# <sup>□</sup> To turn on or off Windows Integrated security by using SQL Server Management Studio

1.

Open SQL Server Management Studio and connect to the Reporting Services instance that you want to configure.

• In Object Explorer, right-click the Reporting Services node, and click **Properties**.

- In the **Server Properties** dialog box, under **Select a page**, click **Security**.
  - To turn on Windows Integrated security, select the **Enable Windows Integrated security for report data sources** option.
  - To turn off Windows integrated security, unselect the **Enable Windows Integrated security for report data sources** option.
- Click **OK**.

## □ See Also

### Tasks

How to: Start Reporting Services Configuration

#### **Other Resources**

RSReportServer Configuration File Component Architecture Reporting Services Components and Tools Service Architecture (Reporting Services) Introducing SQL Server Management Studio Administering Servers by Using Policy-Based Management

## SQL Server Setup Help Reporting Services Configuration F1 Help Topics

-

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> >

This section provides F1 help for the Reporting Services Configuration tool.



# <sup>□</sup> In This Section

- <u>Connect to a Report Server</u>
- <u>Report Server Status</u>
- <u>Service Account</u>
- Web Service URL
- Advanced Multiple Web Site Configuration
- <u>Database</u>
- <u>Change Database Wizard</u>
- Change Credentials Wizard
- <u>Report Manager URL</u>
- <u>E-mail Settings</u>
- Execution Account
- Encryption Keys
- Backup Encryption Key
- <u>Restore Encryption Key</u>
- <u>Scale-out Deployment</u>

## $\Box$ See Also

### Tasks

How to: Start Reporting Services Configuration

## **Other Resources**

**Reporting Services Configuration Tool Configuring a Report Server Installation** 

#### SQL Server Setup Help

Connect to a Report Server (Reporting Services Configuration)

# See Also

#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

Use this dialog box to connect to a local or remote SQL Server 2008 Reporting Services report server instance. You cannot use this tool to connect to earlier versions of Reporting Services report servers. You can only connect to one instance at a time.

- To connect to a local report server instance, use the default values and click **Connect**. The Reporting Services Configuration tool provides the local server name and detects the default instance. In most cases, you can click **Connect** without having to change the values. If you installed more than one instance, you must select the one that you want to use.
- To connect to a remote report server instance, type the server name, click **Find**, select the instance, and then click **Connect**.

To open this dialog box, start the Reporting Services Configuration tool. This dialog box appears immediately when you start the tool. For more information, see <u>How to: Start Reporting Services Configuration</u>.

# • Options

#### Server Name

Enter the network name of the computer on which SQL Server 2008 Reporting Services is installed. Type just the computer name; do not include a prefix or slashes.

#### Find

Find the computer specified in **Server Name**.

#### **Report Server Instance**

Select which instance to connect to if multiple report server instances are installed. Only valid instances are available for selection. If you are running older versions of Reporting Services side-by-side a SQL Server instance, those instances will not appear in the list.

#### Connect

Connect to the server and instance you specified.

## □ See Also

**Tasks**How to: Start Reporting Services Configuration

Concepts

Working with Multiple Versions and Instances of SQL Server

## **Other Resources**

**Reporting Services Configuration Tool** 

## SQL Server Setup Help Report Server Status (Reporting Services Configuration)



### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

Use this page to view information about the report server instance to which you are currently connected. This page is the start page for report server configuration. Additional pages are available to configure URLs, the service account, the report server database, report server e-mail delivery, scale-out deployment, and encryption keys.

To open this page, start the Reporting Services Configuration tool and connect to the report server instance. For more information, see <u>How to: Start Reporting</u> <u>Services Configuration</u>.

If you connect to the report server and all of the page links are grayed out, verify that the Report Server service is started. You can use the Services console application in Administrator Tools to check service status.

## • Options

#### **SQL Server Instance**

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 is MSSQLSERVER. A named instance will be a value that you specified during setup. For more information about instances, see <u>Working with Multiple Versions and Instances of SQL Server</u>.

## <sup>I</sup> Note:

In SQL Server Express with Advanced Services with Advanced Services, the default instance is SQLExpress.

#### **Instance ID**

Corresponds to a folder on the file system that stores program files for the SQL Server instance to which you are connected. The **Instance ID** value is assigned by Setup in the format *component.instance*, where *component* is a value that signifies a SQL Server component and *instance* is an instance name. The default instance name is MSSQLSERVER. For example, if you install default instances of the Database Engine, Analysis Services, and Reporting Services components, the corresponding folder names are MSSQL10.MSSQLSERVER, MSAS10.MSSQLSERVER, and MSRS10.MSSQLSERVER, respectively. If you install a second instance of a component that you already installed, such as the Database Engine, and you name the instance Contoso, the **Instance ID** is MSSQL10.Contoso.

#### Edition

Displays edition information. Reporting Services is available in SQL Server Express with Advanced Services, Workgroup, Developer, Evaluation, Standard, and Enterprise editions.

#### **Product Version**

Displays the version of Reporting Services that you installed.

#### **Report Server Database**

Displays the name of the report server database that stores application data for the current report server instance.

#### **Report Server Mode**

Shows whether the report server is deployed in native mode or in SharePoint integrated mode. This value will be set to *SharePoint integrated mode* if the report server database was created using a schema that supports that mode of operation. For more information about deployment modes, see **Planning a Deployment Mode**. For more information about all of the steps required to set up integrated operations, see <u>Configuring Reporting Services</u> for SharePoint 3.0 Integration.

#### **Server Status**

Shows whether the Report Server service is running.

#### Start

Starts the Report Server service. Restarting the service is necessary after some configuration changes (for example, when reconfiguring a report server after a computer name change). If you reconfigure the URL reservations, the service will restart automatically. The restart is required to pick up the changes.

#### Stop

Stops the Report Server service. Stopping the service causes the report server to stop working. For more information, see **Starting and Stopping the Report Server Service**.

## □ See Also

## Tasks

How to: Start Reporting Services Configuration

## Concepts

<u>Reporting Services Configuration F1 Help Topics</u> <u>Initializing a Report Server</u>

### **Other Resources**

Service Architecture (Reporting Services) Configuring a Report Server Installation Starting and Stopping the Report Server Service

## SQL Server Setup Help Service Account (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

Use the Service Account page to specify the account under which the Report Server service runs. This account is initially configured during Setup. You can modify it if you want to change the account or password. The Report Server Web service, Report Manager, and the background processing application all run under the service identity you specify on this page.

The account you specify for the Report Server service requires permission to access the registry, report server program files, and the report server database. All permissions are configured for the account automatically when you use the Reporting Services Configuration tool to set the account. If you use the service account to connect to the report server database, the tool creates a database login for the account and configures database permissions by assigning the account to the **RSExecRole** on the SQL Server instance that hosts the report server database. The report server database is the only data store that a report server writes to. The service account does not require permissions to any other data stores.

To open this page, start the Reporting Services Configuration tool and select the link in the navigation pane. For more information, see <u>How to: Start Reporting</u> <u>Services Configuration</u>.

## <sup>I</sup> Important:

Whenever you need to update the account or password, it is strongly recommended that you use the Reporting Services Configuration tool. Using the tool to update the account ensures that other internal settings that depend on the service identity are automatically updated at the same time.

# • Options

#### Use a built-in account

Select **Network Service**, **Local System**, or **Local Service** from the list. Only **Network Service** is recommended; however, you can configure the account to use any account that is available.

#### Use another account

Select this option to specify a Windows user account. You can enter a local Windows user account or domain user account. Specify a domain account in this format: *<domain>*\*<user>*. Specify a local Windows user account in this format: *<computer name>*\*<user>*. You can only select an existing account; you cannot create new accounts in Reporting Services Configuration.

The maximum character limit on the account is 20 characters.

If your network uses Kerberos authentication and you configure the report server to run under a domain user account, you must register the service with the user account. For more information, see <u>How to: Register a Service</u> <u>Principal Name (SPN) for a Report Server</u>.

If you switch the account type (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 encryption key. The backup copy will be restored automatically when you select the new account.

## <sup>I</sup>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 available to the report server. For more information about these actions, see <u>Encryption Keys</u> (<u>Reporting Services Configuration</u>).

Additionally, if you have a report server that is configured to run in SharePoint

Integrated mode and you change the service account by using the Reporting Services Configuration tool, you must also open SharePoint Central Administration and use the Reporting Services **Grant Database Access** page to re-apply the report server and instance settings. This step will grant the new service account access to the SharePoint databases, which is required for integrating Reporting Services with Windows SharePoint Services (WSS) or Microsoft Office SharePoint Server (MOSS). For more information about how to grant database access in SharePoint Central Administration, see <u>How to:</u> <u>Configure Report Server Integration in SharePoint Central Administration</u>.
# <sup>□</sup> Choosing an Account

For best results, specify an account that has network connection permissions, with access to network domain controllers and corporate SMTP servers or gateways. The following table summarizes the accounts and provides recommendations for using them.

Account	Explanation
Domain user accounts	If you have a Windows domain user account that has the minimum permissions required for report server operations, you should use it.
	Server service from other applications. Running multiple applications under a shared account, such as Network Service, increases the risk of a malicious user taking control of the report server because a security breach for any one application can easily extend to all applications that run under the same account.
	A domain user account is required if you are configuring the report server for constrained delegation, or for a specific configuration in SharePoint integrated mode where the report server, report server database, and the SharePoint configuration and content databases are on one computer, and the SharePoint Web application is on a second computer.
	Note that if you use a domain user account, you will have to change the password periodically if your organization enforces a password expiration policy. You might also need to register the service with the user account. For more information, see <u>How to: Register a Service</u> <u>Principal Name (SPN) for a Report Server</u> .
	Avoid using a local Windows user account. Local accounts typically do not have sufficient permission to access resources on other computers. For more information about how using a local account limits report server functionality, see <u>Considerations for Using Local</u> <u>Accounts</u> in this topic.
	<b>Network Service</b> is a built-in least-privilege account that has network

Network Service	logon permissions. This account is recommended if you do not have a domain user account available or if you want to avoid any service disruptions that might occur as a result of password expiration policies.
	If you select <b>Network Service</b> , try to minimize the number of other services that run under the same account. A security breach for any one application will compromise the security of all other applications that run under the same account.
Local Service	<b>Local Service</b> is a built-in account that is like an authenticated local Windows user account. Services that run as the <b>Local Service</b> account access network resources as a null session with no credentials. This account is not appropriate for intranet deployment scenarios where the report server must connect to a remote report server database or a network domain controller to authenticate a user prior to opening a report or processing a subscription.
Local System	<b>Local System</b> is a highly privileged account that is not required for running a report server. Avoid this account for report server installations. Choose a domain account or <b>Network Service</b> instead.

# <sup>©</sup> Considerations for Using Local Accounts

The primary consideration for using local accounts is whether the report server requires access to remote database servers, mail servers, and domain controllers. If you configure the report server to run as a local Windows user account, Local Service, or Local System, you introduce considerations that must be factored into how you set other configuration settings, and on subscription creation and delivery:

- Running the service under a local account will limit your options later if you configure a connection to a remote report server database. Specifically, if you are using a remote report server database, you will have 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 service will try to connect to a domain controller to authenticate the user when the subscription is processed. If the 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 service under a local account, it cannot access a remote domain controller to resolve the target e-mail account.

### Tasks

How to: Configure a Service Account for Reporting Services

### Concepts

<u>Configuring the Report Server Service Account</u> <u>Reporting Services Configuration F1 Help Topics</u>

### **Other Resources**

**Report Server Starting and Stopping the Report Server Service** 

## SQL Server Setup Help Web Service URL (Reporting Services Configuration)



### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

Use the Web Service URL page to configure or modify the URL used to access the report server. A *URL reservation* will be created based on the URL you specify. The URL reservation defines the syntax and rules for all URLs that can be subsequently used to access the Report Server Web service. It specifies the prefix, host, port, and virtual directory for the Report Server Web service. Depending on how you specify the host, multiple URLs might be possible for a single reservation. The default value for the host specifies a strong wildcard. A strong wildcard allows you to specify in a URL any host name that can be resolved to the computer that hosts the report server. For more information about URL configuration and reservations, see <u>How to: Configure a URL (Reporting</u> <u>Services Configuration</u>) and <u>Configuring Report Server URLs</u>.

To open this page, start the Reporting Services Configuration tool and click **Web Service URL** in the navigation pane. For more information, see <u>How to: Start</u> <u>Reporting Services Configuration</u>.

This page provides values that are commonly used in report server URLs. If you want to create additional URLs, use host headers, or specify the IP address in a particular format, click **Advanced**.

A link to the Web service will appear on this page after you click **Apply**. If you click this link before the report server database is created, you can expect to see a "Page Not Found" error. This error will no longer appear once the database is configured. For more information, see <u>How to: Create a Report Server Database</u> (Reporting Services Configuration).

If you reinstalled Reporting Services and find that you get errors when attempting to use the default IP address value of All Assigned and port 80, you can usually resolve the error by re-creating the URL after restarting the service. For instructions, see **Troubleshooting Configuration Errors**.

# • Options

#### **Virtual Directory**

Specifies the virtual directory name for the Report Server Web service. You can only have one virtual name for each Report Server Web service instance on the same computer.

#### **IP Address**

Identifies the report server computer on a TCP/IP network. Valid values include:

- All Assigned specifies that any of the IP addresses that are assigned to the computer can be used in a URL that points to a report server application. This value also encompasses friendly host names (such as computer names) that can be resolved by a domain name server to an IP address that is assigned to the computer. This is the default value for a Reporting Services URL.
- **All Unassigned** specifies that the report server will accept any request that does not have an exact match for the IP address or host name. Do not use this value if another Web application is already using it. If you do so, you will disrupt service for the other application.
- **127.0.0.1** is used to access to localhost. It supports local administration on the report server computer. If you select only this value, only users who are logged on locally to the report server computer will have access the application.
- *Nnn.nnn.nnn* is the IPv4 address of a network adapter card on your computer. If your network uses IPv6 addressing, the IP address will be a 128-bit value of 8 4-byte fields similar to the following format: <hr/>
  <header>:nnnn:nnnn:nnnn</hr>

If you have multiple cards, you will see an IP address for each one. If you select only this value, it will limit application access to the just the IP address (and any host name that a domain name server maps to that address). You cannot use localhost to access a report server, and you cannot use the IP addresses of other network adapter cards that are installed on the report server computer.

#### **TCP Port**

Specifies the port that report server monitors for HTTP requests for URLs that include the report server virtual directory name.

#### **SSL Certificate**

Binds a certificate to the IP address you specified. The certificate must be installed and configured on the computer. Reporting Services does not provide features for managing certificates. The certificate must be issued to a host name or a computer name that resolves to the IP address. For example, to use a certificate that was issued to http://salesreports, the IP address you specified must resolve to a server named salesreports.

If you use a certificate, you must 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.

#### SSL Port

Specifies the port for SSL connections.

#### URLs

Displays the URLs defined for the current report server instance.

#### Advanced

Click to create additional URLs for the current application instance.

### Concepts

Reporting Services Configuration F1 Help Topics Configuring Report Server URLs

### **Other Resources**

Deploying Reporting Services and Internet Information Services Side-by-Side Planning for Extranet or Internet Deployment

### SQL Server Setup Help Advanced Multiple Web Site Configuration (Reporting Services Configuration)



#### - -

<u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1 Help Topics</u> > <u>Web Service URL</u> (<u>Reporting Services Configuration</u>) >

Use this dialog box to create and manage the URLs used to access a report server or Report Manager. The **Advanced Multiple Web Site Configuration** dialog box is used to create additional URLs, custom URLs that include a host header name, or specify an IP address in the IPv4 or IPv6 format.

Creating multiple URLs is useful if you want to configure different ways to access a report server. For example, report server access over intranet and extranet connection typically requires that you have different URLs for each type of connection.

To open the **Advanced Multiple Web Site Configuration** dialog box, click **Advanced** on the **Web Service URL** or the **Report Manager URL** page in the Reporting Services Configuration tool. After the **Advanced Multiple Web Site Configuration** dialog box is open, you can click **Add** or **Edit** to define new URLs or modify or delete existing URLs.

Click **OK** to save your changes. If you add or remove URLs, but then close the dialog box without first clicking **OK**, your changes will be lost.

# <sup>©</sup> Options

### **IP Address**

Identifies the report server computer on a TCP/IP network. Valid values include:

- All Assigned specifies that any of the IP addresses that are assigned to the computer can be used in a URL that points to a report server application. This value also encompasses friendly host names (such as computer names) that can be resolved by a domain name server to an IP address that is assigned to the computer. This is the default value for a Reporting Services URL.
- All Unassigned specifies that the report server will accept any request that does not have an exact match for the IP address or host name. Do not use this value if another Web application is already using it. If you do so, you will disrupt service for the other application.
- **127.0.0.1** is used to access to localhost. It supports local administration on the report server computer. If you select only this value, only users who are logged on locally to the report server computer will have access the application.
- *Nnn.nnn.nnn* is the IPv4 address of a network adapter card on your computer. If your network uses IPv6 addressing, the IP address will be a 128-bit value of 8 4-byte fields similar to the following format: <header>:nnnn:nnn:nnnn.

If you have multiple cards, you will see an IP address for each one. If you select only this value, it will limit application access to the just the IP address (and any host name that a domain name server maps to that address). You cannot use localhost to access a report server, and you cannot use the IP addresses of other network adapter cards that are installed on the report server computer.

#### Port

Specifies the port that report server monitors for requests. Port 80 is the default port. If you use port 80, you do not have to include it in the URL. If you use any other port number, you must always include it in the URL (for example, http://localhost:8181/reports).

#### **Host Header**

If you already have a host header defined on a domain name server that resolves to your computer, you can specify that host header in a URL that you configure for report server access.

A host header is a unique name that allows multiple Web sites to share a single IP address and port. Host header names are easier to remember and type than IP address and port numbers. An example of a host header name might be www.adventure-works.com.

#### **SSL Port**

Specifies the port for SSL connections. The default port for SSL is 443.

#### **SSL Certificate**

Specifies the certificate name of an SSL certificate that you installed on this computer. If the certificate maps to a wildcard, you can use it for a report server connection.

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.

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.

#### **Issued To**

Shows the name of the computer for which the certificate was created.

#### Add

Define an additional URL.

### Edit

Modify any part of the URL syntax.

### Remove

Clear a URL entry from the list.

# $\Box$ See Also

## Tasks

How to: Start Reporting Services Configuration How to: Configure a URL (Reporting Services Configuration)

### Concepts

Configuring Report Server URLs

## SQL Server Setup Help Database (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

Use the Database page to create and configure the report server databases that provide internal storage for one or more report server instances. If you are configuring a report server to use a remote report server database, you must use the Reporting Services Configuration tool to create the database.

Creating a report server database and configuring the connection is a multi-step process. To guide you through the steps, this page provides Wizards for each type of task. Permissions and logins are created or updated for you. You can monitor the status of each step in the Progress page. If an error occurs, you can click the error for information on how to resolve it.

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 <u>How to: Create a Report Server Database (Reporting Services Configuration)</u> and <u>How to: Create a Report Server Database for SharePoint Integrated Mode (Reporting Services Configuration)</u>.

To open this page, start the Reporting Services Configuration tool and click **Database** in the navigation pane. For more information, see <u>How to: Start</u> <u>Reporting Services Configuration</u>.

# • Options

#### SQL Server Name

In Current Report Server Database, **SQL Server Name** specifies the name of the SQL Server 2005 or SQL Server 2008 Database Engine that runs the report server database. You can use a default or named instance on a local or remote computer.

#### **Database Name**

Specifies the name of the report server database that stores server data.

#### **Report Server Mode**

Indicates whether the report server database supports native mode or SharePoint integrated mode. For more information, see **Planning a Deployment Mode**.

#### **Change Database**

Start a wizard that guides you through all of the steps required for creating or selecting a report server database.

#### **Credential Type**

Specifies credentials that the report server uses to connect to the report server database. Credential types you can specify include the service account, a Windows domain user, Windows local user, or SQL Server database login. For more information about selecting credentials, see <u>Configuring a Report Server Database Connection</u>.

#### **User 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.

## **Change Credentials**

Start a wizard that guides you through all of the steps required for selecting a different account or updating the password on the account that is used to connect to the report server database.

### Tasks

How to: Create a Report Server Database (Reporting Services Configuration)

### Concepts

Reporting Services Configuration F1 Help TopicsCreating a Report Server DatabaseUpgrading a Report Server DatabaseConfiguring a Report Server Database ConnectionConfiguring Reporting Services for Scale-Out Deployment

### **Other Resources**

**Report Server Database Administering a Report Server Database** 

# SQL Server Setup Help Change Database Wizard (Reporting Services Configuration)

# See Also

### - -

<u>Configuring Reporting Services > Reporting Services Configuration F1 Help Topics > Database</u> (<u>Reporting Services Configuration</u>) >

The Reporting Services Configuration tool provides the Change Database Wizard to guide you through the steps of creating a new report server database or selecting an existing report server database to use with the current report server instance.

If you select a report server database from an earlier version, it will be upgraded to match the version of the report server instance to which it is connected. When the service starts, it checks the database version and upgrades it to the current schema automatically.

To start the wizard, click **Change Database** on the Database page in the Reporting Services Configuration tool. For instructions on how to start the Reporting Services Configuration tool, see <u>How to: Start Reporting Services</u> <u>Configuration</u>.

# • Options

#### Action

Select the task you want to perform. You can create a new database in native or SharePoint integrated mode. Or, you can select an existing report server database to use with the current report server instance.

#### **Database Server**

Specify the name of the SQL Server Database Engine instance that hosts the report server database. You can use a SQL Server 2005 or SQL Server 2008 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>*.

To connect to the Database Engine instance, you must use credentials that have permission to log on to the server and update database information. The Reporting Services Configuration tool uses your current Windows credentials, but if you do not have a login or database permissions, you must specify a SQL Server database login. You cannot specify different Windows credentials. If you want to connect as a different Windows user, login as that user and then start the Reporting Services Configuration tool.

Connecting to a remote instance requires that you first enable that instance for remote connections. Some versions and editions of SQL Server do not enable remote connections by default. To confirm whether remote connections are allowed, use the SQL Server Configuration Manager tool and confirm that the TCP/IP and Named Pipes protocols are enabled. 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.

#### Database

Specifies the name of the report server database that stores server data. You can specify an existing database or create a new database.

Properties that are used to create a new database appear in the Wizard when you select **Create new database** on the Actions page. 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. For more information, see **Report Server Database**.

You can also choose an existing report server database. The Reporting Services Configuration tool does not filter out invalid databases. Valid databases are based on the report server database schema (you cannot select a database that is missing the necessary tables, views, or stored procedures). If you choose a database that was created for an earlier version of Reporting Services, the database will be upgraded to the current format.

#### Language

This value is set only when you are creating a new report server database.

With this value, you specify the language in which role definitions and descriptions are created. Reporting Services provides a role-based authorization model that includes a set of predefined roles. These roles are created once in the language you specify. Role names and descriptions never appear in other languages, even if you connect to the report server using a browser that has culture or language settings that are supported by the server. The language that you specify also determines the language used to create the name of the My Reports folder and the Users folders that are part of the My Reports feature. For more information about language support in Reporting Services, see **Solution Design Considerations for Multi-Lingual or Global Deployments (Reporting Services)**.

#### Server Mode

A report server database supports either native mode or SharePoint integrated mode. The modes are mutually exclusive.

If you are creating a new report server database, you must specify a mode. The mode you select determines the structure of the report server database and sets the **SharePointIntegrated** report server system property to **true** or **false**.

If you are selecting a different report server database, the mode of the

current database is displayed so that you know how the current database is used. You can select a database that was created for a different mode. Doing so will switch the server mode of the current report server instance. For more information about modes, see **Planning a Deployment Mode**.

#### Credentials

Specifies the account by which the report server connects to the report server database. Valid values include the service account of the Report Server Web service, a SQL Server database login defined on the Database Engine instance you are using to host the report server, or a Windows account. If you are using a Windows account, you can specify a local account (*<computername*>\*<username*>) if the report server and the database are on the same computer, or a domain user account (*<domain*>\ *<username*>) if they are on different computers in the same domain.

The report server will create a database login and assign database permissions for the account you specify.

The report server does not create the account itself. The account you specify must already exist and it must be valid for your deployment configuration. Specifically, if the database is on a remote computer and you want to use a Windows account, you must specify an account that has log on permissions on that computer.

If the computer is in a different or non-trusted domain, consider using a SQL Server database login. For more information about choosing an account, see <u>Configuring a Report Server Database Connection</u>.

#### Summary

Verify the settings before Setup configures the connection.

#### **Progress and Finish**

Monitor the progress of each task.

### Tasks

How to: Create a Report Server Database (Reporting Services Configuration)

### Reference

Database (Reporting Services Configuration) Change Credentials Wizard (Reporting Services Configuration)

### Concepts

Reporting Services Configuration F1 Help Topics Configuring a Report Server Database Connection Configuring Reporting Services for Scale-Out Deployment

### **Other Resources**

Planning a Deployment Mode Report Server Database Administering a Report Server Database

# SQL Server Setup Help Change Credentials Wizard (Reporting Services Configuration)

# See Also

### - -

<u>Configuring Reporting Services > Reporting Services Configuration F1 Help Topics > Database</u> (<u>Reporting Services Configuration</u>) >

The Reporting Services Configuration tool provides the Change Credentials Wizard to guide you through the steps of reconfiguring the account that the report server uses to connect to the report server database. When you change credentials, the tool will update all permissions and database login information on the database server for the report server database that is actively used by the report server.

To start the wizard, click **Change Credentials** on the Database page in the Reporting Services Configuration tool. For instructions on how to start the Reporting Services Configuration tool, see <u>How to: Start Reporting Services</u> <u>Configuration</u>.

# • Options

#### **Database Server**

Specifies the name of the SQL Server Database Engine instance that runs the report server database.

To connect to the Database Engine instance, you must use credentials that have permission to log on to the server and update database information. The Reporting Services Configuration tool uses your current Windows credentials, but if you do not have a login or database permissions, you can specify a SQL Server database login.

You cannot specify different Windows credentials. If you want to connect as a different Windows user, login as that user and then start the Reporting Services Configuration tool.

#### Credentials

Specifies the account by which the report server connects to the report server database. Valid values include the service account of the Report Server Web service, a SQL Server database login defined on the Database Engine instance you are using to host the report server, or a Windows account. If you are using a Windows account, you can specify a local account (*<computername*>\*<username*>) if the report server and the database are on the same computer, or a domain user account (*<domain*>\ *<username*>) if they are on different computers in the same domain.

The report server will create a database login and assign database permissions for the account you specify.

The report server does not create the account itself. The account you specify must already exist and it must be valid for your deployment configuration. Specifically, if the database is on a remote computer and you want to use a Windows account, you must specify an account that has log on permissions on that computer.

If the computer is in a different or non-trusted domain, consider using a

SQL Server database login. For more information about choosing an account, see <u>Configuring a Report Server Database Connection</u>.

#### Summary

Verify the settings before the wizard runs.

### **Progress and Finish**

Monitor the progress of each task.

### Tasks

How to: Create a Report Server Database (Reporting Services Configuration)

### Reference

Database (Reporting Services Configuration) Change Database Wizard (Reporting Services Configuration)

### Concepts

Reporting Services Configuration F1 Help Topics Configuring a Report Server Database Connection Configuring Reporting Services for Scale-Out Deployment

### **Other Resources**

**Report Server Database Administering a Report Server Database** 

## SQL Server Setup Help Report Manager URL (Reporting Services Configuration)



### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

Use the Report Manager URL page to configure or modify the URL used to access Report Manager. By default, the Report Manager URL inherits the prefix, IP address, and port of the Report Server Web service URL. This is because Report Manager provides front-end access to the Web service that runs within the same Report Server service. If you are isolating the service applications and using Report Manager to access a Report Server Web service on a different computer, you must edit RSReportServer.config file to point Report Manager to a different instance. For more information about configuring a Report Manager.

If you are configuring the report server to run in SharePoint integrated mode, do not create a Report Manager URL. Report Manager is not supported on a report server that runs in SharePoint integrated mode. If a URL already exists for Report Manager, it will become unavailable after you configure the report server to run in SharePoint integrated mode.

To open this page, start the Reporting Services Configuration tool and click **Report Manager URL** in the navigation pane. For more information about how to start the tool, see <u>How to: Start Reporting Services Configuration</u>.

## <sup>I</sup>Note:

If Report Manager is not enabled, you cannot set options on this page. For more information about enabling Report Manager, see <u>How to: Turn</u> <u>Reporting Services Features On or Off</u>.

# • Options

#### **Virtual Directory**

Specifies the virtual directory name for Report Manager. You can only have one virtual directory name for each Report Manager instance on the same computer.

#### URLs

Displays the URL defined for the current Report Manager instance.

#### Advanced

Add an additional URL for the current Report Manager instance.

### Tasks

How to: Configure a URL (Reporting Services Configuration)

### Concepts

<u>URLs in Configuration Files (Reporting Services)</u> <u>Reporting Services Configuration F1 Help Topics</u> <u>Configuring Report Server URLs</u>

### **Other Resources**

Service Architecture (Reporting Services) RSReportServer Configuration File Planning a Deployment Mode Report Manager

# SQL Server Setup Help E-mail Settings (Reporting Services Configuration)



### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

Use this page to specify Simple Mail Transport Protocol (SMTP) settings that enable report server e-mail delivery from the report server. You can use the Report Server E-Mail delivery extension 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 <u>Configuring a Report Server for E-Mail Delivery</u> and **Configuration Files** (**Reporting Services**).

To open this page, start the Reporting Services Configuration tool and click **E-mail Settings** in the navigation pane. For more information, see <u>How to: Start</u> <u>Reporting Services Configuration</u>.

# • Options

#### **Sender Address**

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.

#### **Current SMTP 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 <u>Configuring a Report Server for E-Mail Delivery</u>.

#### **SMTP Server**

Specify the SMTP server or gateway to use. You can use a local server or an SMTP server on your network.

# $\Box$ See Also

### Concepts

Configuring a Report Server for E-Mail Delivery Reporting Services Configuration F1 Help Topics

# SQL Server Setup Help Execution Account (Reporting Services Configuration)



#### - ▼

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

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 account. 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 the same way 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.

# <sup>ℤ</sup>Note:

If you are using SQL Server 2008 Express 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. SQL Server Express does not support a data source connection to a remote server.

To open this page, start the Reporting Services Configuration tool and select **Execution Account** in the navigation pane. For more information, see <u>How to:</u> <u>Start Reporting Services Configuration</u>.

# <sup>□</sup> Options

### Specify an execution account

Select to specify an account.

#### Account

Enter a Windows domain user account. Use this format: *<domain>\<user account>*.

#### Password

Type the password.

#### **Confirm password**

Retype the password.

### Concepts

Reporting Services Configuration F1 Help Topics Storing Encrypted Report Server Data Configuring the Unattended Execution Account

### **Other Resources**

**RSReportServer Configuration File**
# SQL Server Setup Help Encryption Keys (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

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 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 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 service. Only the Report Server 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 service, or if you migrate the report server to a new computer, the private key of the Report Server 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 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 re-create the key.

# <sup>I</sup> 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.

To open this page, start the Reporting Services Configuration tool and select the link in the navigation pane. For more information, see <u>How to: Start Reporting</u> <u>Services Configuration</u>.

# • 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 the report servers that use the report server database. For more information about initializing report servers, see <u>Initializing a Report Server</u>.

#### Change

Recreates the symmetric key and re-encrypts all encrypted values in the report server database. Be sure to stop the Report Server 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 **Scale-out Deployment** page. Stop the 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.

# □ See Also

## Concepts

Reporting Services Configuration F1 Help Topics Backing Up and Restoring Encryption Keys Deleting and Re-creating Encryption Keys Initializing a Report Server Storing Encrypted Report Server Data

#### **Other Resources**

**Troubleshooting Initialization and Encryption Key Errors** 

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Backup Encryption Key (Reporting Services Configuration)



#### - ▼

<u>Configuring Reporting Services > Reporting Services Configuration F1 Help Topics > Encryption Keys</u> (<u>Reporting Services Configuration</u>) >

Reporting Services uses an encryption key to secure sensitive data that is stored in the report server database. Having a backup of this key is essential for ensuring continued access to encrypted connection strings and credentials.

You must have a backup copy of this key if you move the report server database to another computer or if you change the user name or password of the Report Server service account. Both operations require that you restore the key from a backup copy that you previously created.

To open the Backup Encryption Key dialog box, click **Encryption Keys** in the navigation pane of the Reporting Services Configuration tool, and then click **Backup**. This dialog box also appears when you update the service account using the Service Account page in the Reporting Services Configuration tool.

# • Options

#### **File Location**

Specify a file name and location for Reporting Services to the symmetric key. The symmetric key is never stored in plain text. You must type a password to protect the file.

#### Password

Type a password that protects the file against unauthorized access. Only users who know the password will be able to restore the key that is locked inside the file. Reporting Services enforces a strong password policy. The password must be at least 8 characters and include a combination of uppercase and lowercase alphanumeric characters and at least one symbol character.

#### **Confirm Password**

Re-type the password you entered.

# □ See Also

#### Tasks

How to: Start Reporting Services Configuration

#### Reference

Encryption Keys (Reporting Services Configuration)

## Concepts

Reporting Services Configuration F1 Help Topics Backing Up and Restoring Encryption Keys Deleting and Re-creating Encryption Keys Initializing a Report Server Storing Encrypted Report Server Data

#### **Other Resources**

**Troubleshooting Initialization and Encryption Key Errors** 

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Restore Encryption Key (Reporting Services Configuration)



#### - -

<u>Configuring Reporting Services > Reporting Services Configuration F1 Help Topics > Encryption Keys</u> (Reporting Services Configuration) >

Reporting Services uses an encryption key to secure sensitive data that is stored in the report server database. To ensure that you have continued access to encrypted data, it is important that you create a backup of the encryption key in case you need to restore it later due to changes in the service account or as part of a planned migration.

To restore the key, you must have previously saved a backup copy of the key to a password protected file. During key restoration, the report server will replace an existing key with the key that is found in the password protected file. The key that is inside the file must be identical to the key that is used to encrypt and decrypt data.

To verify whether you restored a valid key, use Report Manager to view subscriptions or any report that has a data source that uses stored credentials. If you receive "The report server is unable to access encrypted data" error when attempting to open a subscription definition page, or if you are prompted to enter credentials when opening a report that previously used stored credentials for the report data source, you restored an invalid key.

If you a restore an invalid key that is different from the one used to encrypt the data, it will be impossible to decrypt data that is currently stored in the report server database. If you restore an invalid key, you should immediately restore a backup copy of the correct key if it is available. If you do not have a backup copy of the key that was used to encrypt data, you must delete all encrypted data. Click the **Delete** button on the Encryption Keys page to perform this step. After you delete the encrypted content, you must manually update all subscriptions and re-specify all stored credentials defined for reports and data-driven subscriptions on the report server.

To open the Restore Encryption Key dialog box, click **Encryption Keys** in the navigation pane of the Reporting Services Configuration tool, and then click

**Restore**. This dialog box also appears when you update the service account using the Service Account page in the Reporting Services Configuration tool.

# • Options

#### **File Location**

Select the password-protected file that contains a copy of the symmetric key. The default file extension is .snk.

#### Password

Enter the password that unlocks the file. Only users who know the password can restore the key. Reporting Services enforces a strong password policy. The password must contain at least 8 characters and include a combination of uppercase and lowercase alphanumeric characters and at least one symbol character.

# □ See Also

#### Tasks

How to: Start Reporting Services Configuration

#### Reference

Encryption Keys (Reporting Services Configuration)

## Concepts

Reporting Services Configuration F1 Help Topics Backing Up and Restoring Encryption Keys Deleting and Re-creating Encryption Keys Initializing a Report Server Storing Encrypted Report Server Data

#### **Other Resources**

**Troubleshooting Initialization and Encryption Key Errors** 

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Scale-out Deployment (Reporting Services Configuration)



#### - -

<u>Configuring SQL Server 2008</u> > <u>Configuring Reporting Services</u> > <u>Reporting Services Configuration F1</u> <u>Help Topics</u> >

Use this page to view the initialization status for a scale-out deployment or to join a report server to a scale-out deployment.

An *initialized report server* describes a server that can encrypt and decrypt sensitive data that is stored in a report server database (stored credentials and connection strings are examples of encrypted data that is stored in the database). Report server initialization is a requirement for report server operations.

A *scale-out deployment* refers to two or more report server instances that share a single report server database. This configuration is used in the following scenarios:

- As a prerequisite for load balancing multiple report servers in a server cluster. Before you can load balance multiple report servers, you must first configure them to share the same report server database.
- To segment report server applications on different computers, by using one server for interactive report processing and a second server for scheduled report processing. In this scenario, each server instance processes different types of requests for the same report server content stored in the shared report server database.

To configure a scale-out deployment, start with two or more report server instances that are all connected to the same report server database. After all of the instances are installed, you connect to the first report server and then use the Scale-out Deployment page to join each additional instance. Only a report server that is already initialized to use a database can initialize additional nodes.

To open this page, start the Reporting Services Configuration tool and select **Scale-out Deployment** in the navigation pane. For more information, see <u>How</u> to: <u>Start Reporting Services Configuration</u>.

# • Options

#### **SQL Server Name**

Specify the name of the SQL Server Database Engine instance that hosts the report server database.

#### **Database** Name

Specifies the name of the database to which the report server instance is currently connected.

#### Server Mode

Displays the mode of server and database. The server mode is either Native or SharePoint integrated. Scale-out deployments are supported for both modes.

#### Server

Shows the report server name. In most cases, this is the name of the computer on which the report server is installed.

#### Instance

Shows the report server instance name. Report server instances are based on SQL Server instances.

#### Status

Indicates whether the report server is initialized or waiting to join a scaleout deployment:

- For a standalone report server that is not part of a scale-out deployment, this page shows that the report server instance is initialized with respect to its dedicated report server database. Status is set to **Joined**.
- For a report server that is waiting to join a scale-out deployment, this page contains empty values for Server, Instance, and Status. A report

server is waiting to join a scale-out deployment if you selected an existing report server database that is already used by another report server instance. A message on this page instructs you to connect to a report server that is already joined to the farm. To complete this request, click **Connect**, select a report server that is already initialized to use the report server database, click **Scale-out Deployment**, select the report server instance that is **Waiting to Join**, and then click **Initialize**.

• 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 is reported identically for all nodes in the scale-out deployment.

For a report server that is already part of a scale-out deployment, you can use this page to add or remove nodes.

#### Initialize

Click **Initialize** to add a report server to the scale-out deployment. This step configures 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.

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 <u>Encryption Keys (Reporting Services Configuration)</u>.

# □ See Also

#### Tasks

<u>How to: Configure a Report Server Scale-Out Deployment (Reporting Services</u> <u>Configuration</u>)

## Concepts

Configuring Reporting Services for Scale-Out Deployment Initializing a Report Server Configuring and Managing Encryption Keys

Other Resources Planning a Deployment Mode rskeymgmt Utility

© 2008 Microsoft Corporation. All rights reserved.

# SQL Server Setup Help Configuring the Windows Firewall to Allow SQL Server Access

# 

#### - -

#### <u>Configuring SQL Server 2008</u> >

Firewall systems help prevent unauthorized access to computer resources. If a firewall is turned on but not correctly configured, attempts to connect to SQL Server might be blocked.

To access an instance of the SQL Server through a firewall, you must configure the firewall on the computer that is running SQL Server to allow access. The firewall is a component of Microsoft Windows. You can also install a firewall from another company. This topic discusses how to configure the Windows firewall, but the basic principles apply to other firewall programs.

## <sup>☑</sup> Note:

This topic provides an overview of firewall configuration and summarizes information of interest to a SQL Server administrator. For more information about the firewall and for authoritative firewall information, see the firewall documentation, such as <u>Windows Firewall with Advanced Security and</u> <u>IPsec</u> and <u>Windows Firewall with Advanced Security - Content Roadmap</u>.

Users familiar with the **Windows Firewall** item in Control Panel and with the Windows Firewall with Advanced Security Microsoft Management Console (MMC) snap-in and who know which firewall settings they want to configure can move directly to the topics in the following list:

- How to: Configure a Windows Firewall for Database Engine Access
- How to: Configure Windows Firewall for Analysis Services Access
- How to: Configure a Firewall for Report Server Access
- How to: Configure a Windows Firewall for Integration Services

# <sup>□</sup> In this Topic

This topic has the following sections:

**Basic Firewall Information** 

**Default Firewall Settings** 

Programs to Configure the Firewall

Ports Used by the Database Engine

Ports Used By Analysis Services

Ports Used By Reporting Services

Ports Used By Integration Services

Additional Ports and Services

Interaction with Other Firewall Rules

**Overview of Firewall Profiles** 

Additional Firewall Settings

Using the Windows Firewall with Advanced Security Snap-in

**Troubleshooting Firewall Settings** 

# <sup>Basic</sup> Firewall Information

Firewalls work by inspecting incoming packets, and comparing them against a set of rules. If the rules allow the packet, the firewall passes the packet to the TCP/IP protocol for additional processing. If the rules do not allow the packet, the firewall discards the packet and, if logging is enabled, creates an entry in the firewall logging file.

The list of allowed traffic is populated in one of the following ways:

- When the computer that has the firewall enabled initiates communication, the firewall creates an entry in the list so that the response is allowed. The incoming response is considered solicited traffic and you do not have to configure this.
- An administrator configures exceptions to the firewall. This allows either access to specified programs running on your computer, or access to specified connection ports on your computer. In this case, the computer accepts unsolicited incoming traffic when acting as a server, a listener, or a peer. This is the type of configuration that must be completed to connect to SQL Server.

Choosing a firewall strategy is more complex than just deciding if a given port should be open or closed. When designing a firewall strategy for your enterprise, make sure that you consider all the rules and configuration options available to you. This topic does not review all the possible firewall options. We recommend that you review the following documents:

<u>Getting Started with Windows Firewall with Advanced Security in Windows</u> <u>Vista and Windows Server 2008</u>

Windows Firewall with Advanced Security Design Guide

Introduction to Server and Domain

# <sup>□</sup> Default Firewall Settings

The first step in planning your firewall configuration is to determine the current status of the firewall for your operating system. If the operating system was upgraded from a previous version, the earlier firewall settings may have been preserved. Also, the firewall settings could have been changed by another administrator or by a Group Policy in your domain. However, the default settings are as follows:

- Windows Server 2008 The firewall is on and is blocking remote connections.
- Windows Server 2003 The firewall is off. Administrators should consider turning on the firewall.
- Windows Vista The firewall is on and is blocking remote connections.
- Windows XP, Service Pack 2 or later
  - The firewall is on and is blocking remote connections.
- Window XP, Service Pack 1 or earlier The firewall is off and should be turned on.

## <sup>I</sup>Note:

Turning on the firewall will affect other programs that access this computer, such as file and print sharing, and remote desktop connections. Administrators should consider all applications that are running on the computer before adjusting the firewall settings.

# <sup>B</sup> Programs to Configure the Firewall

There are three ways to configure the Windows Firewall settings.

#### • Windows Firewall item in Control Panel

The **Windows Firewall** item can be opened from Control Panel.

## <sup>Important:</sup>

Changes made in the **Windows Firewall** item in Control Panel only affect the current profile. Mobile devices, for example a laptop, should not use the **Windows Firewall** item in Control Panel as the profile might change when it is connected in a different configuration. Then the previouslyconfigured profile will not be in effect. For more information about profiles, see <u>Getting Started with Windows Firewall with Advanced</u> <u>Security in Windows Vista and Windows Server 2008</u>.

The **Windows Firewall** item in Control Panel allows you to configure basic options. These include the following:

- Turning the **Windows Firewall** item in Control Panel on or off
- Enabling and disabling rules
- Granting exceptions for ports and programs
- Setting some scope restrictions

The **Windows Firewall** item in Control Panel is most appropriate for users who are not experienced in firewall configuration, and who are configuring basic firewall options for computers that are not mobile. You can also open the **Windows Firewall** item in Control Panel from the **run** command by using the following procedure:

To open the Windows Firewall item

- 1. On the **Start** menu, click **Run**, and then enter firewall.cpl.
- 2. Click **OK**.

#### • Microsoft Management Console (MMC)

The Windows Firewall with Advanced Security MMC snap-in lets you configure more advanced firewall settings. This snap-in is only available for Microsoft Vista and Windows Server 2008; however, it presents most of the firewall options in an easy-to-use manner, and presents all firewall

profiles. For more information, see <u>Using the Windows Firewall with</u> <u>Advanced Security Snap-in</u> later in this topic.

• netsh

The **netsh.exe** tool can be used by an administrator to configure and monitor Windows-based computers at a command prompt. By using the **netsh** tool, you can direct the context commands you enter to the appropriate helper, and the helper then performs the command. A helper is a Dynamic Link Library (.dll) file that extends the functionality of the **netsh** tool by providing configuration, monitoring, and support for one or more services, utilities, or protocols. All operating systems that support SQL Server have a firewall helper. Microsoft Windows Vista and Windows Server 2008 also have an advanced firewall helper called **advfirewall**. How to use **netsh** is not discussed in this topic. However, many of the configuration options described can be configured by using **netsh**. For scripts to configure SQL Server using netsh, see <u>How to use a script to</u> programmatically open ports for SQL Server to use on systems that are running Windows XP Service Pack. For more information about **netsh**, see the following links:

- How to Use the Netsh.exe Tool and Command-Line Switches
- How to use the "netsh advfirewall firewall" context instead of the "netsh firewall" context to control Windows Firewall behavior in Windows Server 2008 and in Windows Vista
- <u>The "netsh firewall" command together with the "profile=all"</u> parameter does not configure the public profile on a Windows Vistabased computer
- <u>Troubleshooting Windows Firewall settings in Windows XP Service</u> <u>Pack 2 for advanced users</u>

# <sup>©</sup> Ports Used By SQL Server

The following tables can help you identify the ports being used by SQL Server.

## **Ports Used By the Database Engine**

The following table lists the ports that are frequently used by the Database Engine.

Scenario	Port	Comments
SQL Server default instance running over TCP	TCP port 1433	This is the most common port allowed through the firewall. It applies to routine connections to the default installation of the Database Engine, or a named instance that is the only instance running on the computer. (Named instances have special considerations. See <u>Dynamic Ports</u> later in this topic.)
SQL Server named instances in the default configuration	The TCP port is a dynamic port determined at the time the Database Engine starts.	See the discussion below in the section <u>Dynamic Ports</u> . UDP port 1434 might be required for the SQL Server Browser Service when you are using named instances.
SQL Server named instances when they are configured to use a fixed port	The port number configured by the administrator.	See the discussion below in the section Dynamic Ports.
Dedicated Admin	TCP port 1434	By default, remote connections to the Dedicated Administrator Connection (DAC) are not enabled. To enable remote DAC, use the Surface Area

Connection		Configuration facet. For more information, see <b>Understanding Surface Area Configuration</b> .
SQL Server Browser service	UDP port 1434	The SQL Server Browser service listens for incoming connections to a named instance and provides the client the TCP port number that corresponds to that named instance. Normally the SQL Server Browser service is started whenever named instances of the Database Engine are used. The SQL Server Browser service does not have to be started if the client is configured to connect to the specific port of the named instance.
SQL Server default instance running over an HTTP endpoint.	TCP port 80	Used for an HTTP connection through a URL.
SQL Server default instance running over an HTTPS endpoint.	TCP port 443	Used for an HTTPS connection through a URL. HTTPS is an HTTP connection that uses secure sockets layer (SSL).
Service Broker	TCP port 4022	There is no default port for SQL Server Service Broker, but this is the conventional configuration.
Database Mirroring	Administrator chosen port. To determine the port, execute the following query: SELECT name, protocol_desc, port, state_desc FROM	There is no default port for Database mirroring, but this is the conventional configuration. It is very important to avoid interrupting an in-use mirroring endpoint, especially in high-safety mode with automatic failover. Your firewall configuration must avoid

	sys.tcp_endpoints WHERE type_desc = 'DATABASE_MIRRORING'	breaking quorum. For more information, see <b>Specifying a Server</b> <b>Network Address (Database</b> <b>Mirroring)</b> .
Replication	Replication connections to SQL Server use the typical regular Database Engine ports (TCP port 1433, etc.) Web synchronization and FTP/UNC access for replication snapshot require additional ports to be opened on the firewall. To transfer initial data and schema from one location to another, replication can use FTP (TCP port 21), or sync over HTTP (TCP port 80) or File and Print Sharing (TCP port 137,138, or 139).	For sync over HTTP, replication uses the IIS endpoint (ports for which are configurable but is port 80 by default), but the IIS process connects to the backend SQL Server through the standard ports (1433 for the default instance. During Web synchronization using FTP, the FTP transfer is between IIS and the SQL Server publisher, not between subscriber and IIS. For more information, see Configuring Microsoft Internet Security and Acceleration Server for Microsoft SQL Server 2000 Replication over the Internet.
Transact- SQL debugger	TCP port 135 See <u>Special Considerations</u> for Port 135 The <u>IPsec</u> exception might also be required.	If using Visual Studio, on the Visual Studio host computer, you must also add <b>Devenv.exe</b> to the Exceptions list and open TCP port 135. If using Management Studio, on the Management Studio host computer, you must also add <b>ssms.exe</b> to the Exceptions list and open TCP port 135. For more information, see <b>Configuring and Starting the</b> <b>Transact-SQL Debugger</b> .

For step by step instructions to configure the Windows Firewall for the Database Engine, see **How to: Configure a Windows Firewall for Database Engine Access**.

## **Dynamic Ports**

Named instances (including SQL Server Express) use dynamic ports. That means that every time that the Database Engine starts, it identifies an available port and uses that port number. If the named instance is the only instance of the Database Engine installed, it will probably use TCP port 1433. If other instances of the Database Engine are installed, it will probably use a different TCP port. Because the port selected might change every time that the Database Engine is started, it is difficult to configure the firewall to enable access to the correct port number. Therefore, if a firewall is started, we recommend reconfiguring the Database Engine to use the same port number every time. This is called a fixed port or a static port. For more information, see **Configuring a Fixed Port**.

An alternative to configuring a named instance to listen on a fixed port is to create an exception in the firewall for a SQL Server program such as **sqlservr.exe** (for the Database Engine). This can be convenient, but the port number will not appear in the **Local Port** column of the **Inbound Rules** page when you are using the Windows Firewall with Advanced Security MMC snapin. This can make it more difficult to audit which ports are open. Another consideration is that a service pack or cumulative update can change the path to the SQL Server executable which will invalidate the firewall rule.

## <sup>I</sup>∕ Note:

The following procedure uses the **Windows Firewall** item in Control Panel. The Windows Firewall with Advanced Security MMC snap-in can configure a more complex rule. This includes configuring a service exception which can be useful for providing defense in depth. See <u>Using the Windows Firewall with</u> <u>Advanced Security Snap-in</u> below.

To add a program exception to the firewall using the Windows Firewall item in Control Panel.

- 1. On the **Exceptions** tab of the **Windows Firewall** item in Control Panel, click **Add a program**.
- Browse to the location of the instance of SQL Server that you want to allow through the firewall, for example C:\Program Files\Microsoft SQL Server\MSSQL10.<instance\_name>\MSSQL\Binn, select sqlservr.exe,

and then click **Open**.

3. Click **OK**.

For more information about endpoints, see **Network Protocols and TDS Endpoints** and **Endpoints Catalog Views (Transact-SQL)**.

## **Ports Used By Analysis Services**

Feature	Port	Comments
Analysis Services	TCP port 2383 for the default instance	The standard port for the default instance of Analysis Services.
SQL Server Browser service	TCP port 2382 only needed for an Analysis Services named instance	Client connection requests for a named instance of Analysis Services that do not specify a port number are directed to port 2382, the port on which SQL Server Browser listens. SQL Server Browser then redirects the request to the port that the named instance uses.
Analysis Services configured for use through IIS/HTTP (The PivotTable® Service uses HTTP or HTTPS)	TCP port 80	Used for an HTTP connection through a URL.
Analysis Services		

The following table lists the ports that are frequently used by Analysis Services.

configured for use through IIS/HTTPS	TCP port 443	Used for an HTTPS connection through a URL. HTTPS is an HTTP connection that uses secure sockets layer (SSL).
(The PivotTable® Service uses		
HTTP or HTTPS)		

If users access Analysis Services through IIS and the Internet, you must open the port on which IIS is listening and specify that port in the client connection string. In this case, no ports have to be open for direct access to Analysis Services. The default port 2389, and port 2382, should be restricted together with all other ports that are not required.

For step by step instructions to configure the Windows Firewall for Analysis Services, see **How to: Configure Windows Firewall for Analysis Services Access**.

## **Ports Used By Reporting Services**

The following table lists the ports that are frequently used by Reporting Services.

Feature	Port	Comments
Reporting Services Web Services	TCP port 80	Used for an HTTP connection to Reporting Services through a URL. We recommend that you do not use the preconfigured rule <b>World Wide Web Services (HTTP)</b> . For more information, see the <u>Interaction with Other Firewall Rules</u> section below.
Reporting Services configured for use through HTTPS	TCP port 443	Used for an HTTPS connection through a URL. HTTPS is an HTTP connection that uses secure sockets layer (SSL). We recommend that you do not use the preconfigured rule <b>Secure World Wide Web Services (HTTPS)</b> . For more information, see the <u>Interaction with Other Firewall Rules</u> section below.

When Reporting Services connects to an instance of the Database Engine or Analysis Services, you must also open the appropriate ports for those services. For step-by-step instructions to configure the Windows Firewall for Reporting Services, <u>How to: Configure a Firewall for Report Server Access</u>.

## **Ports Used By Integration Services**

The following table lists the ports that are used by the Integration Services service.

Feature	Port	Comments
Microsoft		The Integration Services service uses DCOM on port
remote		135. The Service Control Manager uses port 135 to
procedure		perform tasks such as starting and stopping the
calls (MS	TCP port 135	Integration Services service and transmitting control
RPC)		requests to the running service. The port number
	See <u>Special</u>	cannot be changed.
Used by	<b>Considerations</b>	
the	<u>for Port 135</u>	This port is only required to be open if you are
Integration		connecting to a remote instance of the Integration
Services		Services service from Management Studio or a
runtime.		custom application.

For step-by-step instructions to configure the Windows Firewall for Integration Services, see **Configuring a Windows Firewall for Integration Services Access** and **How to: Configure a Windows Firewall for Integration Services**.

## **Additional Ports and Services**

The following table lists ports and services that SQL Server might depend on.

Scenario	Port	Comments
Windows	WMI runs as	
Management	part of a	
	shared service	
	host with ports assigned	SQL Server Configuration Manager uses WMI
For more	ussigned	to list and manage services. We recommend that

information	through	you use the preconfigured rule group <b>Windows</b>
about WMI, see	DCOM. WMI	Management Instrumentation (WMI). For
WMI Provider	might be using	more information, see the <u>Interaction with Other</u>
for	TCP port 135.	Firewall Rules section below.
Configuration		
Management	See <u>Special</u>	
Concepts	<b>Considerations</b>	
	<u>for Port 135</u>	
Microsoft Distributed Transaction Coordinator (MS DTC)	TCP port 135 See <u>Special</u> <u>Considerations</u> for Port 135	If your application uses distributed transactions, you might 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. We recommend that you use the preconfigured <b>Distributed Transaction</b> <b>Coordinator</b> rule group.
The browse button in Management Studio uses UDP to connect to the SQL Server Browser Service. For more information, see <b>SQL</b> <b>Server</b> <b>Browser</b> <b>Service</b> .	UDP port 1434	<ul> <li>UDP is a connectionless protocol.</li> <li>The firewall has a setting, which is named UnicastResponsesToMulticastBroadcastDisabled Property of the INetFwProfile Interface which controls the behavior of the firewall with respect to unicast responses to a broadcast (or multicast) UDP request. It has two behaviors:</li> <li>If the setting is TRUE, no unicast responses to a broadcast are permitted at all. Enumerating services will fail.</li> <li>If the setting is FALSE (default), unicast responses are permitted for 3 seconds. The length of time is not configurable. in a congested or high-latency network, or for heavily loaded servers, tries to enumerate instances of SQL Server might return a partial list, which might mislead users.</li> </ul>
		If the domain policy requires network communications to be done through IPsec, you

IPsec traffic	UDP port 500 and UDP port 4500	must also add UDP port 4500 and UDP port 500 to the exception list. IPsec is an option using the <b>New Inbound Rule Wizard</b> in the Windows Firewall snap-in. For more information, see <u>Using the Windows Firewall with Advanced</u> <u>Security Snap-in</u> below.
Using Windows Authentication with Trusted Domains	Firewalls must be configured to allow authentication requests.	For more information, see <u>How to configure a</u> <u>firewall for domains and trusts</u> .
SQL Server and Windows Clustering	Clustering requires additional ports that are not directly related to SQL Server.	For more information, see <u>Enable a network for</u> <u>cluster use</u> .
URL namespaces reserved in the HTTP Server API (HTTP.SYS)	Probably TCP port 80, but can be configured to other ports. For general information, see <u>Configuring HTTP and</u> <u>HTTPS</u> .	For SQL Server specific information about reserving an HTTP.SYS endpoint using HttpCfg.exe, see <b>Reserving URL Namespaces</b> <b>by Using Http.sys</b> .

# Special Considerations for Port 135

When you use RPC with TCP/IP or with UDP/IP as the transport, inbound ports are frequently dynamically assigned to system services as required; TCP/IP and UDP/IP ports that are larger than port 1024 are used. These are frequently informally referred to as "random RPC ports." In these cases, RPC clients rely on the RPC endpoint mapper to tell them which dynamic ports were assigned to the server. For some RPC-based services, you can configure a specific port instead of letting RPC assign one dynamically. You can also restrict the range of ports that RPC dynamically assigns to a small range, regardless of the service. For more information about this topic, see the following references:

- <u>Service overview and network port requirements for the Windows Server</u> <u>system</u>
- <u>Troubleshooting RPC Endpoint Mapper errors using the Windows Server</u> 2003 Support Tools from the product CD
- Using Distributed COM with Firewalls
- <u>Remote</u>
- How to configure RPC dynamic port allocation to work with firewalls
- <u>Windows 2000 Startup and Logon Traffic Analysis</u>

# <sup>□</sup> Interaction with Other Firewall Rules

The Windows Firewall uses rules and rule groups to establish its configuration. Each rule or rule group is generally associated with a particular program or service, and that program or service might modify or delete that rule without your knowledge. For example, the rule groups **World Wide Web Services (HTTP)** and **World Wide Web Services (HTTPS)** are associated with IIS. Enabling those rules will open ports 80 and 443, and SQL Server features that depend on ports 80 and 443 will function if those rules are enabled. However, administrators configuring IIS might modify or disable those rules. Therefore, if you are using port 80 or port 443 for SQL Server, you should create your own rule or rule group that maintains your desired port configuration independently of the other IIS rules.

The Windows Firewall with Advanced Security MMC snap-in allows any traffic that matches any applicable allow rule. So if there are two rules that both apply to port 80 (with different parameters), traffic that matches either rule will be permitted. So if one rule allows traffic over port 80 from local subnet and one rule allows traffic from any address, the net effect is that all traffic to port 80 is permitted regardless of the source. To effectively manage access to SQL Server, administrators should periodically review all firewall rules enabled on the server.

# <sup>□</sup> Overview of Firewall Profiles

Firewall profiles are discussed in <u>Getting Started with Windows Firewall with</u> <u>Advanced Security in Windows Vista and Windows Server 2008</u> in the section **Network location-aware host firewall**. To summarize, Windows Vista and Windows Server 2008 identify and remember each of the networks to which they connect with regard to connectivity, connections, and category.

There are three network location types in Windows Firewall with Advanced Security:

- Domain. Windows can authenticate access to the domain controller for the domain to which the computer is joined.
- Public. Other than domain networks, all networks are initially categorized as public. Networks that represent direct connections to the Internet or are in public locations, such as airports and coffee shops should be left public.
- Private. A network identified by a user or application as private. Only trusted networks should be identified as private networks. Users will likely want to identify home or small business networks as private.

The administrator can create a profile for each network location type, with each profile containing different firewall policies. Only one profile is applied at any time. Profile order is applied as follows:

- 1. If all interfaces are authenticated to the domain controller for the domain of which the computer is a member, the domain profile is applied.
- 2. If all interfaces are either authenticated to the domain controller or are connected to networks that are classified as private network locations, the private profile is applied.
- 3. Otherwise, the public profile is applied.

Use the Windows Firewall with Advanced Security MMC snap-in to view and configure all firewall profiles. The **Windows Firewall** item in Control Panel only configures the current profile.

# <sup>□</sup> Additional Firewall Settings Using the Windows Firewall item in Control Panel

Exceptions that you add to the firewall can restrict the opening of the port to incoming connections from specific computers or the local subnet. This restriction of the scope of the port opening can reduce how much your computer is exposed to malicious users, and is recommended.

## <sup>I</sup>∕ Note:

Using the **Windows Firewall** item in Control Panel only configures the current firewall profile.

To change the scope of a firewall exception using the Windows Firewall item in Control Panel

- 1. In the **Windows Firewall** item in Control Panel, select a program or port on the **Exceptions** tab, and then click **Properties** or **Edit**.
- 2. In the Edit a Program or Edit a Port dialog box, click Change Scope.
- 3. Choose one of the following options:

#### • Any computer (including those on the Internet)

Not recommended. This will allow any computer that can address your computer to connect to the specified program or port. This setting might be necessary to allow information to be presented to anonymous users on the internet, but increases your exposure to malicious users. Your exposure can be further increased if you enable this setting and also allow Network Address Translation (NAT) traversal, such as the Allow edge traversal option.

#### • **My network (subnet) only** This is a more secure setting than **Any computer**. Only computers on the local subnet of your network can connect to the program or port.

#### • Custom list:

Only computers that have the IP addresses you list can connect. This can be a more secure setting than **My network (subnet) only**, however, client
computers using DHCP can occasionally change their IP address. Then the intended computer will not be able to connect. Another computer, which you had not intended to authorize, might accept the listed IP address and then be able to connect. The **Custom list** option might be appropriate for listing other servers which are configured to use a fixed IP address; however, IP addresses might be spoofed by an intruder. Restricting firewall rules are only as strong as your network infrastructure.

# <sup>D</sup> Using the Windows Firewall with Advanced Security Snap-in

On computers that are running Vista or Windows Server 2008, additional advanced firewall settings can be configured by using the Windows Firewall with Advanced Security MMC snap-in. The snap-in includes a rule wizard and exposes additional settings that are not available in the **Windows Firewall** item in Control Panel. These settings include the following:

- Encryption settings
- Services restrictions
- Restricting connections for computers by name
- Restricting connections to specific users or profiles
- Edge traversal allowing traffic to bypass Network Address Translation (NAT) routers
- Configuring outbound rules
- Configuring security rules
- Requiring IPsec for incoming connections

To create a new firewall rule using the New Rule wizard

- 1. On the Start menu, click **Run**, type **WF.msc**, and then click **OK**.
- 2. In the **Windows Firewall with Advanced Security**, in the left pane, rightclick **Inbound Rules**, and then click **New Rule**.
- 3. Complete the **New Inbound Rule Wizard** using the settings that you want.

# <sup>D</sup> Troubleshooting Firewall Settings

The following tools and techniques can be useful in troubleshooting firewall issues:

- The effective port status is the union of all rules related to the port. When trying to block access through a port, it can be helpful to review all the rules which cite the port number. To do this, use the Windows Firewall with Advanced Security MMC snap-in and sort the inbound and outbound rules by port number.
- Review the ports that are active on the computer on which SQL Server is running. This review process includes verifying which TCP/IP ports are listening and also verifying the status of the ports. To verify which ports are listening, use the **netstat** command-line utility. In addition to displaying active TCP connections, the **netstat** utility also displays a variety of IP statistics and information. To list which TCP/IP ports are listening
  - 1. Open the Command Prompt window.
  - 2. At the command prompt, type **netstat -n -a**.

The **-n** switch instructs **netstat** to numerically display the address and port number of active TCP connections. The **-a** switch instructs **netstat** to display the TCP and UPD ports on which the computer is listening.

• The **PortQry** utility can be used to report the status of TCP/IP ports as listening, not listening, or filtered. (With a filtered status, the port might or might not be listening; this status indicates that the utility did not receive a response from the port.) The **PortQry** utility is available for download from the <u>Microsoft Download Center</u>.

For additional troubleshooting topics, see:

- <u>Troubleshooting Windows Firewall with Advanced Security in Windows</u> <u>Vista and Windows Server 2008 - Diagnostics and Tools</u>
- Troubleshooting (Database Engine)
- Troubleshooting (Analysis Services Data Mining)

- Troubleshooting (Reporting Services)Troubleshooting (Integration Services)

## □ See Also

#### Concepts

Before Installing Failover Clustering

#### **Other Resources**

Service overview and network port requirements for the Windows Server system

SQL Server Setup Help System and Sample Databases

-

<u>Configuring SQL Server 2008</u> >

When SQL Server 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
mssqlsystemresource	Mssqlsystemresource.mdf	Mssqlsystemresource.ldf

The system databases are **master**, **model**, **msdb**, **mssqlsystemresource** and **tempdb**.

#### <sup>I</sup>Note:

The default location of the database and log files is Program Files\Microsoft SQL Server\MSQL10.<InstanceName>\MSSQL\. This location will change if the default directory was changed when SQL Server was installed.

# <sup>□</sup> AdventureWorks Sample Databases

By default, the sample databases are not installed in this SQL Server release. For more information about installing sample databases and sample code, see the <u>CodePlex Web site</u>.

Additional information about samples is available after SQL Server 2008 has been installed. From the **Start** menu, click **All Programs**, click **Microsoft SQL Server 2008**, click **Documentation and Tutorials**, and then click **Microsoft SQL Server Samples Overview**.

## □ See Also

#### Tasks

Considerations for Installing SQL Server Samples and Sample Databases

## Concepts

Installing SQL Server 2008 Upgrading to SQL Server 2008

#### **Other Resources** Getting Started with SQL Server Projects and Administration

SQL Server Setup Help SQL Server Setup User Interface Reference

- -

This section describes the installation options presented by the SQL Server Installation Wizard.

## $\Box$ See Also

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Accept License Terms

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Accept License Terms** page of the SQL Server Installation Wizard to accept the license terms for this release of SQL Server.

# **□** Options

You can print the license agreement or copy it to the clipboard. To continue, accept the license terms, and then click **Next**. To quit the installation, click **Cancel**.

## □ See Also

#### Tasks

How to: Install SQL Server 2008 (Setup)

#### Reference

About the SQL Server License Terms

SQL Server Setup Help Add SQL Server Failover Cluster Node



#### - -

<u>SQL Server Setup User Interface Reference</u> >

The following rules are used by SQL Server Setup to validate your computer configuration before the Setup operation finishes.

Check item	Description	User action
SSMS_IsInternetConnected	Verifies that the computer is connected to the Internet. When a .NET Framework application such as Microsoft Management Studio starts, there might be a slight delay while the .NET Framework security check validates a certificate.	Ensure that the computer is connected to the Internet.
AclPermissionsFacet	Checks if the SQL Server registry keys are consistent.	SQL Server registry keys must be consistent to support SQL Server installation or upgrade.
		You must install or

BlockMixedArchitectureUpgrade	Checks whether the CPU architecture of features you are trying to upgrade is different from the CPU architecture of this installation program.	upgrade by using SQL Server installation media that matches the architecture of the instance of SQL Server that you are modifying.
Cluster_IsOnlineIfClustered	Verifies that the computer is not clustered, or that the computer is clustered and the cluster service is online.	The computer must either not be clustered, or the cluster must be up and online.
FacetDomainControllerCheck	Checks whether the computer is a domain controller. Installing SQL Server 2008 on a domain controller is not recommended.	Ensure that this computer is not a domain controller. See <u>Security</u> <u>Considerations</u> <u>for a SQL</u> <u>Server</u> <u>Installation</u> .
FacetPowerShellCheck	Checks whether Windows PowerShell is installed. Windows PowerShell is a pre- requisite for some SQL Server 2008 editions.	For installations of SQL Server 2008 Express with Advanced Services, ensure that Windows PowerShell is

		installed.
FacetWOW64PlatformCheck	Determines whether SQL Server Setup is supported on this operating system platform.	This rule will block unsupported installations on the 64-bit platform. See <u>Hardware and</u> <u>Software</u> <u>Requirements</u> <u>for Installing</u> <u>SQL Server</u> <u>2008</u> .
FusionRebootCheck	Checks if a pending computer restart is required. A pending restart can cause Setup to fail.	A computer restart is required before installing SQL Server.
PerfMonCounterNotCorruptedCheck	Checks if the existing performance counter registry hive is consistent.	Ensure that the performance counter registry hive is consistent.
SqlUnsupportedProductBlocker	Checks whether SQL Server 7.0 or SQL Server 7.0 OLAP Services is installed. SQL Server 2008 is not supported side-by-side with SQL Server 7.0.	Remove SQL Server 7.0, or install SQL Server 2008 on a different computer.

## □ See Also

#### Tasks

How to: Install SQL Server 2008 (Setup)

#### Reference

Check Parameters for the System Configuration Checker

#### SQL Server Setup Help Analysis Services Configuration - Account Provisioning

# See Also

## SQL Server Setup User Interface Reference >

- -

Beginning in SQL Server 2005, significant changes were implemented to ensure that SQL Server was more secure than previous versions. Changes included a "secure by design, secure by default, and secure in deployment" strategy designed to help protect the server and its databases from security attacks.

SQL Server 2008 continues the security hardening process by introducing more changes to the server and database components. SQL Server 2008 uses features such as the User Account Control (UAC) found in Windows Vista and Windows Server 2008.

The changes introduced in SQL Server 2008 help decrease the surface and attack areas for the server and its databases by instituting a policy of "Least Privileged Authority." This means that internal accounts are protected and separated into operating system functions and SQL Server functions. These measures include the following:

- The ability to provision one or more Windows principals into the Sysadmin server role inside SQL Server. This option is available during SQL Server Setup.
- The Surface Area Configuration (SAC) tool has been removed. It was replaced by policy-based management and changes in the Configuration Manager tool.

These changes will affect your security planning for SQL Server, and help you create a more complete security profile for your system.

# <sup>□</sup> Considerations for Running SQL Server 2008 on Windows Vista

On Windows 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. By default, administrators perform most actions as a standard user on Windows Vista and Windows Server 2008 operating systems. As a result, you should create a SQL Server login 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.

The following considerations apply when you install and run SQL Server 2008 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

# <sup>©</sup> Options

**Specify SQL Server Administrators** - You must specify at least one system administrator for the instance of SQL Server. To add the account under which SQL Server Setup is running, click the **Current User** button. To add or remove accounts from the list of system administrators, click **Add** or **Remove** and then edit the list of users, groups, or computers that will have administrator privileges for the instance of SQL Server.

When you are finished editing the list, click **OK**, then verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

## $\Box$ See Also

### Reference

Setting Up Windows Service Accounts

## SQL Server Setup Help Analysis Services Configuration - Data Directories



- -

<u>SQL Server Setup User Interface Reference</u> >

The default directories in the following table are user-configurable during SQL Server Setup.

Description	Default directory	Recommendations
Data root directory	C:\Program Files\Microsoft SQL Server\MSAS10. <instanceid>\OLAP\Data\</instanceid>	Ensure that the \Program files\Microsoft SQL Server\ folder is protected with limited permissions. Analysis Services performance depends, in many configurations, on the performance of the storage on which the data directory is located. Place this directory on the highest performing storage that is attached to the system. For failover cluster installations, ensure that data directories are placed on the shared disk.
Log file directory	C:\Program Files\Microsoft SQL Server\MSAS10. <instanceid>\OLAP\Log\</instanceid>	This is the directory for Analysis Services log files, and it includes the FlightRecorder log. If you increase the flight recorder duration, ensure that the log directory has adequate space.
Temp directory	C:\Program Files\Microsoft SQL Server\MSAS10. <instanceid>\OLAP\Temp\</instanceid>	Place the Temp directory on the high performance storage subsystem.
Backup	C:\Program Files\Microsoft	This is the directory for Analysis Services default backup files. Ensure appropriate permissions are set to prevent data loss, and that the user

directory	SQL Server\MSAS10.	group for the Analysis Services
	<instanceid>\OLAP\Backup</instanceid>	service has adequate permissions to
	-	write to the backup directory. Using a
		mapped drive for backup directories
		is not supported.

# <sup>□</sup> Notes

- When you add 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.
- If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.
- Program files and data files cannot be installed in the following situations:
  - On a removable disk drive
  - On a file system that uses compression
  - To a directory where system files are located
  - On shared drives on a failover cluster instance

## □ See Also

## Concepts

File Locations for Default and Named Instances of SQL Server

SQL Server Setup Help

**Cluster Disk Selection** 

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Cluster Disk Selection** page of the SQL Server Installation Wizard to select the shared cluster disk resource for yourSQL Server failover cluster. The cluster disk is where the SQL Server data will be placed.

# • Options

#### **Shared Disks**

Select a single disk from the list. The cluster disk is where the SQL Server data will be placed.

Only one disk can be specified. If you select the group containing the cluster quorum resource, a warning will be displayed. It is recommended that you do not install to the cluster quorum resource.

#### **Available Shared Disks**

Displays a list of available disks, whether each is qualified as a shared disk, and a description of each disk resource.

SQL Server Setup Help

**Cluster Network Selection** 

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Cluster Network Selection** page to specify the network resources for your failover cluster instance.

# <sup>©</sup> Options

**SQL Server Failover Cluster Network Name** - This is the name used to identify your failover cluster instance on the network.

**Network Settings** - Specify the IP type and IP address for your failover cluster instance.

SQL Server Setup Help Cluster Node Configuration

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Cluster Node Configuration** page to add or remove nodes from a failover cluster instance.

# <sup>©</sup> Options

**SQL Server Instance Name** - Use the drop-down list to select the SQL Server failover cluster instance to modify.

**Name of this node** - This field will be populated with the name of the computer that is running SQL Server Setup. This is the failover cluster node that will be added to or removed from the SQL Server failover cluster instance.

#### SQL Server Setup Help Cluster Node Configuration (Complete)



-

<u>SQL Server Setup User Interface Reference</u> >

Use the Cluster Node Configuration (Complete) page to specify an existing instance of SQL Server that has been prepared for clustering.

# <sup>©</sup> Options

From the drop-down boxes:

- SQL Server instance name Select the instance name for the SQL Server failover cluster.
- Name of this node This field will be pre-populated with the computer name where the SQL Server Setup program is running.
- SQL Server Failover Cluster Network Name This field is not prepopulated. Specify the network name for the new SQL Server failover cluster instance. This is the name that identifies the failover cluster instance on the network.

## $\Box$ See Also

## Tasks

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

SQL Server Setup Help Cluster Resource Group



- -

<u>SQL Server Setup User Interface Reference</u> >

Use the Cluster Resource Group page to specify the disk where SQL Server virtual server resources will be located.

# <sup>©</sup> Options

Specify the SQL Server cluster resource group name. You have two options:

- Use the drop-down box to specify an existing group to use.
- Type the name of a new group to create. Note that the name "Available storage" is not a valid group name.
### Tasks

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

SQL Server Setup Help

**Cluster Security Policy** 

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the Cluster Security Policy page to configure the security policy for your failover cluster instance.

# <sup>©</sup> Options

Specify global or local domain groups for clustered services. All resource permissions are controlled by domain-level groups that include SQL Server service accounts as group members. For information about service security IDs (SID) functionality on Windows Server 2008, see <u>Setting Up Windows Service Accounts</u>.

SQL Server Setup Help Complete (Upgrade)

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the Complete (Upgrade) page to verify that the upgrade has completed.



# <sup>□</sup> Options

This dialog box includes a link to the Setup log file for this operation. To continue, click **Close**.

### Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Credentials for Remote Database Upgrade



### - -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Credentials for Remote Database Upgrade** page of the Microsoft SQL Server Installation Wizard to specify a database login to a remote report server database.

If you are upgrading a report server database on a remote SQL Server Database Engine instance, and you are running Setup under credentials that do not have database update permissions on a remote instance, you must enter credentials that Setup can use to connect to and modify the report server database.

# Options

### **User Name**

Specifies a SQL Server database login that is used one time during an upgrade operation to connect to the Database Engine instance that hosts the report server database. The account must have permission to update the report server database schema.

Required if the credentials you use to run Setup do not have sysadmin or database owner permissions on the report server database.

This value must be specified if you are upgrading a report server that uses a remote report server database to store application data, and the credentials you used to run Setup do not have sufficient permissions on the remote server.

#### Password

The password for the login or an empty string if the login does not have a password.

### Concepts

<u>Considerations for Upgrading Reporting Services</u> <u>Considerations for Installing Reporting Services</u>

### SQL Server Setup Help Database Engine Configuration - Account Provisioning

# 

### - -

#### <u>SQL Server Setup User Interface Reference</u> >

Beginning in SQL Server 2005, significant changes were implemented to help ensure that SQL Server was more secure than previous versions. Changes included a "secure by design, secure by default, and secure in deployment" strategy designed to protect the server instance and its databases from security attacks.

SQL Server 2008 continues the security hardening process by introducing more changes to the server and database components. SQL Server 2008 leverages the User Account Control (UAC) found in Windows Vista and Windows Server 2008.

The changes introduced in SQL Server 2008 further decrease the surface and attack areas for the server and its databases by instituting a policy of "Least Privileged Authority." This means that internal accounts are protected and separated into operating system functions and SQL Server functions. These measures include:

- The ability to provision one or more Windows principals into the Sysadmin server role inside SQL Server. This option is available during SQL Server Setup.
- The Surface Area Configuration (SAC) tool has been removed, and replaced by poliyc-based management and changes in the Configuration Manager tool.

These changes will affect your security planning for SQL Server, and help you create a more complete security profile for your system.

# <sup>□</sup> Considerations for Running SQL Server 2008 on Windows Vista

On Windows 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. By default, administrators perform most actions as a standard user on Windows Vista and Windows Server 2008 operating systems. As a result, you should create a SQL Server login 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.

The following considerations apply when you install and run SQL Server 2008 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 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

# <sup>©</sup> Options

**Security Mode** - Select Windows Authentication or Mixed Mode Authentication for your installation.

Windows Principal Provisioning - In previous versions of SQL Server, the Windows Builtin\Administrator local group was placed into the SQL Server sysadmin server role, effectively granting Windows administrators access to the instance of SQL Server.

### <sup>I</sup>Important:

If your organization's processes or code depend on Windows Builtin\Administrator local group access, you must explicitly grant the right to log in to SQL Server. Setup will not allow you to continue until you complete this step.

**Specify SQL Server Administrators** - You must specify at least one Windows principal for the instance of SQL Server. To add the account under which SQL Server Setup is running, click the **Current User** button. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for the instance of SQL Server.

When you are finished editing the list, click **OK**, then verify the list of administrators in the configuration dialog. When the list is complete, click **Next**.

If you select Mixed Mode Authentication, you must provide logon credentials for the builtin SQL Server system administrator (SA) account.

### <sup>a</sup>Security Note:

Do not use a blank password. Use a strong password.

### Windows Authentication Mode

When a user connects through a Windows user account, SQL Server validates the account name and password using the Windows principal

token in the 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.

### <sup>6</sup>Security Note:

When possible, use Windows Authentication.

### <sup>I</sup>Important:

Do not use a blank password. Use a strong password. Never set a blank or weak **sa** password.

### Mixed Mode (Windows Authentication or SQL Server Authentication)

Allows users to connect by using Windows Authentication or SQL Server Authentication. Users who connect through a Windows user account can use 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, you must set strong passwords for all SQL Server accounts.

### <sup>I</sup>∕ 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. 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.

### <sup>I</sup>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 computer 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 this page must meet strong password policy requirements. If you have any automation that uses SQL Server Authentication, ensure that the password meets strong password policy requirements.

### Reference

Setting Up Windows Service Accounts

SQL Server Setup Help Database Engine Configuration - Data Directories



### - -

<u>SQL Server Setup User Interface Reference</u> >

The following table lists the default directories that are user-configurable during SQL Server Setup:

Description	Default directory	Recommendations
Data root directory	C:\Program Files\Microsoft SQL Server\	SQL Server Setup will configure ACLs for SQL Server directories and break inheritance as part of configuration.
User database directory	C:\Program Files\Microsoft SQL Server\MSSQL10. <instanceid>\Data</instanceid>	Best practices for user data directories depend on workload and performance requirements. For failover cluster installations, ensure that data directories are placed on the shared disk.
User database log directory	C:\Program Files\Microsoft SQL Server\MSSQL10. <instanceid>\Data</instanceid>	Ensure that the log directory has adequate space.
Temp DB directory	C:\Program Files\Microsoft SQL Server\MSSQL10. <instanceid>\Data</instanceid>	Best practices for the Temp directory depend on workload and performance requirements.
Temp DB log directory	C:\Program Files\Microsoft SQL Server\MSSQL10. <instanceid>\Data</instanceid>	Ensure that the log directory has adequate space.
Backup directory	C:\Program Files\Microsoft SQL Server\MSSQL10. <instanceid>\Backup</instanceid>	Set appropriate permissions to prevent data loss, and ensure that the user group for the SQL Server service has adequate permissions to write to the backup directory. Using a mapped drive for backup directories is not supported.

# <sup>□</sup> Notes

- 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.
- If you specify non-default installation directories, ensure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server.
- Program files and data files cannot be installed in the following situations:
  - On a removable disk drive
  - On a file system that uses compression
  - To a directory where system files are located
  - On shared drives on a failover cluster instance

## Concepts

File Locations for Default and Named Instances of SQL Server

SQL Server Setup Help Database Engine Configuration - Filestream



### Ξ 🖌

<u>SQL Server Setup User Interface Reference</u> >

Use this page to enable FILESTREAM for this installation of SQL Server 2008.

### **Enable FILESTREAM for Transact-SQL access**

Select to enable FILESTREAM for Transact-SQL access. This control must be checked before the other control options will be available.

#### **Enable FILESTREAM for file I/O streaming access**

Select to enable Win32 streaming access for FILESTREAM.

#### Windows share name

Use this control to enter the name of the Windows share in which the FILESTREAM data will be stored.

#### Allow remote clients to have streaming access to FILESTREAM data

Select this control to allow remote clients to access this FILESTREAM data on this server.

Other Resources Getting Started with FILESTREAM Storage sp\_configure (Transact-SQL)

# SQL Server Setup Help Database Engine Configuration - User Instance

Ξ 🖌

<u>SQL Server Setup User Interface Reference</u> >

Use the **User Instance** 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, like 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 Are 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**.

SQL Server Setup Help Disk Cost Summary

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the Disk Cost Summary page to examine disk space requirements for the SQL Server 2008 features you selected for your installation.

# <sup>©</sup> Options

If required disk space is too high, consider the following options:

- Change installation directories to a drive with more disk space.
- Change the SQL Server features for your installation.
- Create more free space on the specified drive by moving other files or applications.

### Reference

Hardware and Software Requirements for Installing SQL Server 2008

SQL Server Setup Help Edition Upgrade Rules



### - -

<u>SQL Server Setup User Interface Reference</u> >

The following rules are used by SQL Server Setup to validate your computer configuration before the Setup operation completes.

Check item	Description	User action
Engine_OnlySupportedFeaturesUsedCheckSystem	Checks the system databases for features that are not supported in the SQL Server edition to which you are upgrading.	Turn off any features used in system databases tha are not available in the SQL Server edition to which you are upgrading
Engine_OnlySupportedFeaturesUsedCheckUser	Checks all user databases for features that are not supported by the SQL Server edition.	Turn off any features used in system databases tha are not available in the SQL Server edition to which you are upgrading
	Checks	Identify and correct the

Engine_SqlEngineRestartCheck	whether the SQL Server service can be restarted.	issue that is preventing th SQL Server service from coming onlin
Engine_SqlServerServiceDisabled_Id	Checks that SQL Server service is not set to Disabled.	Set the startu type for the SQL Server service to Manual or Automatic.
FacetDomainControllerCheck	Checks whether the computer is a domain controller. Installing SQL Server 2008 on a domain controller is not recommended.	Ensure that this computen is not a domain controller. Se <u>Security</u> <u>Consideration</u> for a SQL <u>Server</u> <u>Installation</u> .
FacetPowerShellCheck	Checks whether Windows PowerShell is installed. Windows PowerShell is a pre-requisite for some SQL Server 2008 editions.	For installations of SQL Server 2008 Express with Advanced Services, ensure that Windows PowerShell is installed.

PerfMonCounterNotCorruptedCheck	Checks if the existing performance counter registry hive is consistent.	Ensure that th performance counter registry hive consistent.
EditionUpgradeMatrixCheck	Checks whether the selected instance of SQL Server meets upgrade matrix requirements.	Some edition upgrade scenarios are not supported See <u>Version</u> and Edition Upgrades.
AS_ValidateEditionUpgrade	Checks whether Analysis Services is being upgraded to a valid edition.	Some edition upgrade scenarios are not supported See <u>Version</u> and Edition Upgrades.
Cluster_EditionDownGradeCheck	Checks whether the edition of the selected instance of SQL Server is supported in this scenario.	Some edition upgrade scenarios are not supported See <u>Version</u> and Edition Upgrades.
	Checks whether the	

BlockMixedArchitectureUpgrade	CPU architecture of features you are trying to upgrade is different than the CPU architecture of this installation program.	Some cross- platform upgrade scenarios are not supportec See <u>Version</u> and Edition Upgrades.
EditionDownGradeCheck	Checks whether the edition of the selected instance of SQL Server is supported in this scenario.	Some edition upgrade scenarios are not supportec See <u>Version</u> and Edition Upgrades.

### Reference

Check Parameters for the System Configuration Checker

SQL Server Setup Help Error and Usage Report Settings



- -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Error and Usage Report Settings** page of the SQL Server Installation Wizard to enable feature error and usage reporting functionality for SQL Server.

# <sup>©</sup> Options

Bu default, the Feature Usage data collection and Error Reporting features are enabled for SQL Server and its components in pre-release versions of SQL Server 2008. 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
- 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.

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 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 SQL Server privacy and data collection policy, see this <u>Microsoft SQL Server Privacy Statement</u>.

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 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 Services, Reporting Services, 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 software and services. Microsoft will use feature usage data to improve SQL Server. SQL Server components monitored by this feature include:

- The Database Engine
- Analysis Services
- Reporting 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

### Configuration Tools menu.

Information is collected once per day for all enabled SQL Server instances. By default, the time of collection is 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:

HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10\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.

### Other Resources Configuring a SQL Server Installation
SQL Server Setup Help Failover cluster report

#### - -

<u>SQL Server Setup User Interface Reference</u> >

Use the Failover Cluster Report page to view report results from the Setup system scan for this operation.

# <sup>©</sup> Options

To view a detailed report for the scan, click the report link.

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Feature Installation Rules



#### - -

<u>SQL Server Setup User Interface Reference</u> >

SQL Server 2008Setup validates your computer configuration before the Setup operation completes. The following table describes check parameters for the System Configuration Checker and resolution to blocking issues.

Check item	Description	Usei
RebootRequiredCheck	Checks if a pending computer restart is required. A pending restart can cause Setup to fail.	A con restart must 1 this co before install Serve
OsVersionCheck	Checks whether the computer meets minimum operating system version requirements.	The o syster meet t minin requir for thi produ <u>Hardv</u> <u>Softw</u> <u>Requi</u> <u>for Ins</u> <u>SQL 5</u> <u>2008</u> .

ThreadHasAdminPrivilegeCheck	Checks whether the account that is running SQL Server Setup has administrative privileges on the computer.	Ensur accou runnir Serve has admin privile the co
WmiServiceStateCheck	Checks whether the WMI service is started and running on the computer.	The V servic be rur
SSMS_IsInternetConnected	Checks to see if the computer is connected to the Internet. SQL Server Management Tools based on .NET Framework technology must check the certificate revocation list on startup.	Ensur comp conne the In
	Checks whether the computer is a	Ensur this co

domain controller. Installing SQL Server 2008 on a domain controller is not recommended.	is not domai contrc <u>Secur</u> <u>Consi</u> for a <u>Serve</u> Install
Checks whether Windows PowerShell is installed. Windows PowerShell is a pre-requisite of SQL Server 2008 Express with Advanced Services.	For install SQL S 2008 ] with Advar Servic ensure Windo Power install
Determine if the setup program is supported on operating system platform.	This r block unsup install on the platfo Hardv Softw Requi for Ins SQL S 2008.
	<ul> <li>domain</li> <li>controller.</li> <li>Installing SQL</li> <li>Server 2008</li> <li>on a domain</li> <li>controller is</li> <li>not</li> <li>recommended.</li> </ul> Checks <ul> <li>whether</li> <li>Windows</li> <li>PowerShell is</li> <li>installed.</li> <li>Windows</li> <li>PowerShell is</li> <li>a pre-requisite</li> <li>of SQL Server</li> <li>2008 Express</li> <li>with</li> <li>Advanced</li> <li>Services.</li> </ul>

PerfMonCounterNotCorruptedCheck	Checks if the existing performance counter registry hive is consistent.	Ensur perfor counte registe consis
SqlUnsupportedProductBlocker	Checks whether SQL Server 7.0 or SQL Server 7.0 OLAP Services is installed. SQL Server 2008 is not supported side-by-side with SQL Server 7.0.	Remo Serve install Serve on a d comp
BlockCrossLanguageInstall	Checks whether the Setup language is the same as the language of existing SQL Server features.	You n upgra Serve using langu versio existii instan SQL S
	Checks whether the installing features are	You n install upgra using Serve install

BlockMixedArchitectureInstall	the same CPU architecture as the specified instance.	media match archite the ins SQL S that ye modif
RS_DoesCatalogExist	Checks whether the catalog database file exists.	Renar move, delete files.
RS_DoesCatalogTempDBExist	Checks whether the catalog temporary database file exists.	Renar move, delete files.
$StandaloneInstall_HasClusteredOrPreparedInstanceCheck$	Checks if the selected instance name is already used by an existing cluster- prepared or clustered instance on any cluster node.	You cluse th selecto instan becau alread Specif differo Serven instan

## Reference

Check Parameters for the System Configuration Checker

SQL Server Setup Help Feature Selection

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the check boxes on the **Feature Selection** page of the SQL Server Installation Wizard to select components for your SQL Server installation.

# <sup>©</sup> Options

A description for each component group appears in the **Description** pane when you select it. You can select any combination of check boxes. To continue, you must make a selection.

The Database Engine and Analysis Services components are the only SQL Server 2008 features that support failover clustering. You can install other components, like Reporting Services or Integration Services, on a failover cluster node, but they are not cluster-aware, and their services do not fail over if the failover cluster node goes offline. For more information, see <u>Getting Started</u> with SQL Server 2008 Failover Clustering.

Select this component group	To install these components and features
Database Engine Services	<ul> <li>The SQL Server Database Engine includes the following components:</li> <li>The Database Engine is the core service for storing, processing, and securing data.</li> <li>Optional: 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.</li> <li>Optional: Full-Text Search provides functionality to issue full-text queries against plain character-based data in SQL Server tables.</li> <li>Tools for managing relational and XML data.</li> </ul>
Analysis Services	Analysis Services includes the tools for creating and managing online analytical processing (OLAP) and data mining applications.
Reporting Services <sup>2,3</sup>	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

<ul> <li>tegration Services is a set of graphical tools and ogrammable objects for moving, copying, and transforming ta.</li> <li>ient Tools includes components for communication tween clients and servers, including network libraries for 3-Library, OLEDB for OLAP, ODBC, ADODB, and DOMD+, and the following SQL Server Management ols:</li> <li>SQL Server Management Studio (SSMS) is an integrated environment for accessing configuring</li> </ul>
<ul> <li>ient Tools includes components for communication tween clients and servers, including network libraries for 3-Library, OLEDB for OLAP, ODBC, ADODB, and DOMD+, and the following SQL Server Management ols:</li> <li>SQL Server Management Studio (SSMS) is an integrated environment for accessing configuring</li> </ul>
<ul> <li>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.</li> <li>SQL Server Configuration Manager provides basic configuration management for SQL Server services, server protocols, client protocols, and client aliases.</li> <li>SQL Server Profiler provides a graphical user interface for monitoring an instance of the Database Engine or an instance of Analysis Services.</li> <li>Database Engine Tuning Advisor helps create optimal sets of indexes, indexed views, and partitions.</li> <li>Replication Monitor allows you to track the status and performance of publications and subscriptions across a replication topology.</li> <li>SQL Server Books Online is the core documentation for SQL Server.</li> </ul>
rvices, and Integration Services solutions.

programmable objects for moving, copying, and transform	rming
data.	

<sup>1</sup>Setup does not configure load-balancing and single-URL addressing for multiple nodes in a Reporting Services 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 <u>Configuring Reporting Services for Scale-Out Deployment</u>.

<sup>2</sup> Microsoft Internet Explorer 6.0 SP 1 is required for the Report Designer component of Reporting Services.

<sup>3</sup> 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.

<sup>4</sup>Internet Explorer 6.0 SP1 is required for SQL Server Management Studio installations.

<sup>5</sup>Internet Explorer 6.0 SP1 is required for Business Intelligence Development Studio installations.

# <sup>B</sup> Shared Component Directory

Features shared by all instances of SQL Server on a single computer are installed to a single directory. They include the following:

- Client Tools
- SQL Server Books Online
- Business Intelligence Development Studio
- Integration Services

By default, shared components are installed to %Program Files%Microsoft SQL Server\. To change the installation path, click the **Browse** button. If you change the installation path for one shared component, you also change it for other shared components. Subsequent installations will install shared components to the same location as the original installation.

On x64-based operating systems, you can specify where 64-bit components will be installed, and where on the WOW64 subsystem 32-bit components will be installed.

# □ Installing AdventureWorks Sample Databases

By default, sample databases and sample code are not installed as part of SQL Server Setup. To install sample databases and sample code, see the <u>CodePlex</u> <u>Web site</u>.

Additional information about samples is available after SQL Server 2008 has been installed. From the **Start** menu, click **All Programs**, click **Microsoft SQL Server 2008**, click **Documentation and Tutorials**, and then click **Microsoft SQL Server Samples Overview**.

## Reference

Editions and Components of SQL Server 2008

SQL Server Setup Help Feature Selection (Uninstall)

#### - -

<u>SQL Server Setup User Interface Reference</u> >

Use the Feature Selection — Uninstall page to select the features to remove from the specified instance of SQL Server.

# <sup>©</sup> Options

You can select any combination of features.

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Features to Upgrade

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Features to Upgrade** page of the SQL Server 2008 Installation Wizard to specify the features to be upgraded to SQL Server 2008.

# <sup>©</sup> Options

• It is not possible to change the features to be upgraded in this release.

# □ See Also

#### Reference

Version and Edition Upgrades

#### Concepts

<u>Using Upgrade Advisor to Prepare for Upgrades</u> <u>Working with Multiple Versions and Instances of SQL Server</u>

SQL Server Setup Help Full-Text Search Upgrade Options



#### - -

<u>SQL Server Setup User Interface Reference</u> >

Use the Full-Text Search Upgrade Options page of the SQL Server Installation Wizard to select the full-text search upgrade option to use for the databases that you are upgrading at this time.

In SQL Server 2005 and earlier versions, each full-text index resides in a fulltext catalog that belongs to a filegroup, has a physical path, and is treated as a database file. In SQL Server 2008, a full-text catalog is a logical concept—a virtual object—that refers to a group of full-text indexes. Therefore, a new fulltext catalog is not treated as a database file with a physical path. However, during upgrade of any full-text catalog that contains data files, a new filegroup is created on same disk. This maintains the old disk I/O behavior after upgrade. Any full-text index from that catalog is placed in the new filegroup if the root path exists. If the old full-text catalog path is invalid, the upgrade keeps the fulltext index in the same filegroup as base table or, for a partitioned table, in the primary filegroup.

# • Options

When you upgrade to SQL Server 2008, choose one of the following full-text upgrade options.

#### Import

Full-text catalogs are imported. Typically, import is significantly faster than rebuild. For example, when using only one CPU, import runs about 10 times faster than rebuild. However, an imported full-text catalog does not use the new and enhanced word breakers introduced in SQL Server 2008, so you might want to rebuild your full-text catalogs eventually.

#### <sup>☑</sup> Note:

Rebuild can run in multi-threaded mode, and if more than 10 CPUs are available, rebuild might run faster than import if you allow rebuild to use all of the CPUs.

If a full-text catalog is not available, the associated full-text indexes are rebuilt. This option is available for only SQL Server 2005 databases.

For information about the impact of importing full-text index, see "Considerations for Choosing a Full-Text Upgrade Option," later in this topic.

#### Rebuild

Full-text catalogs are rebuilt using the new and enhanced word breakers. Rebuilding indexes can take awhile, and a significant amount of CPU and memory might be required after the upgrade.

#### Reset

Full-text catalogs are reset. SQL Server 2005 full-text catalog files are removed, but the metadata for full-text catalogs and full-text indexes is retained. After being upgraded, all full-text indexes are disabled for change tracking and crawls are not started automatically. The catalog will remain empty until you manually issue a full population, after the upgrade completes.

All of these upgrade options ensure that upgraded databases benefit fully from the full-text performance enhancements of SQL Server 2008.

# <sup>I</sup> Important:

For information about using new word breakers and upgrading noise-word files to stoplists after importing a SQL Server 2005 full-text index, see <u>Full-Text</u> <u>Search Upgrade</u>.

# <sup>□</sup> Considerations for Choosing a Full-Text Upgrade Option

When choosing the upgrade option for your upgrade, consider the following:

- How do you use word breakers? The SOL Server 2008 full-text search servic
  - The SQL Server 2008 full-text search service includes new word breakers and stemmers. These might change the results of full-text queries from previous releases for a specific text pattern or scenario. Therefore, how you use word breakers is important when choosing a suitable upgrade option:
    - If the word breakers of the full-text language you use did not change in SQL Server 2008, or if recall accuracy is not critical to you, importing is suitable. Later, if you experience any recall issues, you can upgrade to the new word breakers simply by rebuilding your full-text catalogs. For more information, see <u>Full-Text Search Upgrade</u>.
    - If you care about recall accuracy and you use one of the word breakers that have been improved in SQL Server 2008, rebuilding is suitable.
- Were any full-text indexes built on integer full-text key columns? Rebuilding performs internal optimizations that improve the query performance of the upgraded full-text index in some cases. Specifically, if you have full-text catalogs that contain full-text indexes for which the fulltext key column of the base table is an integer data type, rebuilding achieves ideal performance of full-text queries after upgrade. In this case, we highly recommend you to use the **Rebuild** option.

#### <sup>ℤ</sup> Note:

For full-text indexes in SQL Server 2008, we recommend that the column serving as the full-text key be an integer data type. For more information, see <u>Performance Tuning and Optimization (Full-Text Search)</u>.

• What is the priority for getting your server instance online? Importing or rebuilding during upgrade takes a lot of CPU resources, which delays getting the rest of the server instance upgraded and online. If getting the server instance online as soon as possible is important and if you are willing to run a manual population after the upgrade, **Reset** is suitable.

# <sup>□</sup> Additional Resources

Breaking Changes to Full-Text Search in SQL Server 2008

Behavior Changes to Full-Text Search in SQL Server 2008

Deprecated Database Engine Features in SQL Server 2008

Inquiring About the Full-Text Key Column

Word Breakers and Stemmers

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Installation is Complete

-

<u>SQL Server Setup User Interface Reference</u> >

Click **Close** to finish the installation.

## Reference

Editions and Components of SQL Server 2008

## Concepts

Validating a SQL Server Installation Configuring SQL Server 2008

SQL Server Setup Help Installation Prerequisites



#### - -

<u>SQL Server Setup User Interface Reference</u> >

For SQL Server to install successfully, the following software components are required on the installation computer. They will be installed by SQL Server Setup:

- .NET Framework version 3.5 for SQL Server Express and SQL Server Express Advanced
- .NET Framework version 2.0 SP1 for SQL Server 2008 on Windows Server 2003 (64-bit) IA64 SP1
- .NET Framework version 3.5 for all other editions of SQL Server 2008
- SQL Server Setup support files
- SQL Server Native Client

## <sup>✓</sup>Note:

These components are not removed if Setup fails or is canceled. To remove them, use **Add or Remove Programs** in Control Panel.

# <sup>□</sup> Options

To install the prerequisite files, click **Install**. To quit the installation, click **Cancel**.

## Reference

Hardware and Software Requirements for Installing SQL Server 2008

SQL Server Setup Help Installation Progress

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Setup Progress** page of the SQL Server Installation Wizard to monitor the status of SQL Server Setup.

# Concepts

How to: View SQL Server 2008 Setup Log Files

SQL Server Setup Help Installation Type

#### - -

<u>SQL Server Setup User Interface Reference</u> >

Use the Installation Type page of the SQL Server Installation Wizard to specify whether to install a new instance of SQL Server 2008, or add features to an existing instance.
# <sup>©</sup> Options

Select the radio button that specifies your choice:

- Perform a new installation of SQL Server 2008
- Add features to an existing instance of SQL Server 2008 If you select the option to add features to an existing instance, use the dropdown list to select the instance of SQL Server to update.

#### <sup>II</sup>Note:

You cannot add features to a failover cluster instance after it has been installed. To add SQL Server features to an existing failover cluster, you must perform a new installation to install a separate instance of SQL Server.

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Instance Configuration



#### - -

<u>SQL Server Setup User Interface Reference</u> >

Use the **Instance Configuration** page of the SQL Server Installation Wizard to specify whether to create a default instance or a named instance of SQL Server. If an instance of SQL Server is not already installed, a default instance will be created unless you specify a named instance.

Each instance of SQL Server consists of a distinct set of services that have specific settings for collations and other options The directory structure, registry structure, and service names all reflect the instance name and a specific instance ID created during SQL Server Setup.

An instance is either the default instance or a named instance. The default instance name is MSSQLSERVER; SQLExpress for SQL Server Express. 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. You can install SQL Server as a named instance without installing the default instance first. The default instance could be an installation of SQL Server 2000, SQL Server 2005, or SQL Server 2008. Only one installation of SQL Server, regardless of version, can be the default instance at one time.

# <sup>□</sup> Multiple Instances

SQL Server supports multiple instances of SQL Server 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.

For more information, see <u>Maximum Capacity Specifications for SQL Server</u> <u>2008</u>.

# • Options

Failover cluster instances only — Specify the SQL Server failover cluster network name. This name identifies the failover cluster instance on the network.

Default or Named instance — Consider the following information when you decide whether to install a default or named instance of SQL Server:

- If you plan to install a single instance of SQL Server on a database server, it should be a default instance.
- Use a named instance for situations where you plan to have multiple instances on the same computer. A server can host only one default instance.
- Any application that installs SQL Server Express should install it as a named instance. This will minimizes conflict when multiple applications are installed on the same computer.

#### **Default instance**

Select this option to install a default instance of SQL Server. 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.

#### Named instance

Select this option to create a new named instance. Be aware of the following when you name an instance of SQL Server:

- Instance names are not case sensitive.
- Instance names cannot contain the term "Default" or other reserved keywords. If a reserved keyword is used in an instance name, a Setup error will occur. For more information, see <u>Reserved Keywords</u> (Setup).
- If you specify MSSQLServer for the instance name, a default instance will be created. For SQL Server Express, if you specify SQLExpress for the instance name, a default instance will be created.
- Instance names are limited to 16 characters.

- The first character in the instance name must be a letter. Acceptable letters are those defined by the Unicode Standard 2.0. These include 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. The backslash (\), comma (,), colon (:), semi-colon (;), single quote ('), ampersand (&), and at sign (@) are also not allowed.
  Note:

Only characters that are valid in the current Windows code page can be used in SQL Server instance names. If an unsupported Unicode character is used, a Setup error will occur.

#### **Detected instances and features**

View a list of installed SQL Server instances and components on the computer where SQL Server Setup is running.

**Instance ID** - By default, the instance name is used as the Instance ID. This is used to identify installation directories and registry keys for your instance of SQL Server. This is the case for default instances and named instances. For a default instance, the instance name and instance ID would be MSSQLSERVER. To use a non-default instance ID, specify it in the **Instance ID** field.

### <sup>I</sup>Note:

Instance ID's that begin with an underscore (\_) or that contain the number sign (#) or the dollar sign (\$) are not supported.

For more information about directories, file locations, and instance ID naming, see <u>File Locations for Default and Named Instances of SQL Server</u>.

**Instance root directory** - By default, the instance root directory is C:\Program Files\Microsoft SQL Server\. To specify a non-default root directory, click the **Browse** button, or provide a path name.

All components of a given instance of SQL Server are managed as a unit. All SQL Server service packs and upgrades will apply to every component of an

instance of SQL Server.

All components of SQL Server that share the same instance name must meet the following criteria:

- Same version
- Same edition
- Same language settings
- Same clustered state
  <sup>I</sup> Note: Reporting Services is not cluster-aware.
- Same operating system

### **Other Resources** How to: Install SQL Server 2005

SQL Server Setup Help Instance Selection (Uninstall)

#### - -

<u>SQL Server Setup User Interface Reference</u> >

On the Instance Selection — Uninstall page, use the drop-down box to specify the instance of SQL Server to uninstall.

# **□** Options

You can specify an entire instance of SQL Server, or you can specify that only the shared features and Management Tools are uninstalled.

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Instance Selection (Upgrade)

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the Instance Selection (Upgrade) page to specify the instance of SQL Server to upgrade.

# <sup>©</sup> Options

All components of an instance must be upgraded at the same time. You cannot add or remove features during the upgrade.

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Minimize SQL Server 2008 Surface Area



#### - ▼

#### <u>SQL Server Setup User Interface Reference</u> >

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 SQL Server 2008, 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 Configuration Manager to enable, disable, start, or stop services and configure the remote connectivity of your Database Engine.
- Use Policy-Based Management to enable and disable optional features.

# <sup>D</sup> Using SQL Server Configuration Manager

The SQL Server Configuration Manager is installed with all editions of SQL Server.

To open the SQL Server Configuration Manager

 On the Start menu, point to All Programs, point to Microsoft SQL Server 2008, point to Configuration Tools, and then click SQL Server Configuration Manager.

To configure a service to start automatically

- 1. In SQL Server Configuration Manager, expand **SQL Server Services**.
- 2. In the details pane, right-click one of the SQL Server services, and then click **Properties**.
- 3. In the **Properties** dialog box, set **Start Mode** to **Automatic**.

To configure the Database Engine to accept remote connections

- 1. In SQL Server Configuration Manager, expand **SQL Server Network Configuration**, and then click Protocols for *<instancename>*.
- 2. In the details pane, right-click one of the available protocols, and then click **Properties**.

#### <sup>I</sup>Note:

The shared memory protocol cannot be enabled for remote connections.

3. To enable a protocol for remote connections, set the **Enabled** box to **Yes**.

For help with SQL Server Configuration Manager, view the SQL Server Configuration Manager help file, or see SQL Server Books Online. To configure SOAP and Service Broker endpoints, use **CREATE ENDPOINT** and **ALTER ENDPOINT**.

## <sup>D</sup> Using Policy-Based Management

Policy-Based Management is configured using SQL Server Management Studio. If Management Studio is not installed, run setup and install the client tools. Management Studio is not part of the installation of SQL Server Express. Download Management Studio Express from Microsoft.com. The features of Policy-Based Management are described in SQL Server Books Online.

To open the SQL Server Management Studio

1. On the **Start** menu, point to **All Programs**, point to **Microsoft SQL Server 2008**, and then click **SQL Server Management Studio**.

To configure Policy-Based Management

- 1. In Management Studio, connect to an instance of Database Engine, expand **Management**, and then expand **Policy Management**.
- 2. Configure Policy-Based Management by using the following three facets:
  - Surface Area Configuration
  - Surface Area Configuration for Analysis Services
  - Surface Area Configuration for Reporting Services

The surface area configuration features that are turned off by default should not be turned on unless they are required for a specific business need.

### **Recommended Settings:**

- Recommended settings for the Database Engine Set all properties of the Surface Area Configuration facet to false.
- Recommended settings for Analysis Services Set all properties of the Surface Area Configuration for Analysis Services facet to false.
- Recommended settings for Reporting Services Use the Surface Area Configuration for Reporting Services to disable any Reporting Services features that you do not need.

### **Other Resources Understanding Surface Area Configuration**

SQL Server Setup Help Post Uninstall

-

<u>SQL Server Setup User Interface Reference</u> >

Use the Post Uninstall page to verify that the operation completed successfully.



# <sup>©</sup> Options

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Product Key

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the Product Key page to validate your instance of SQL Server 2008, or to specify which free edition of SQL Server 2008 to install.

# <sup>©</sup> Options

To specify a free edition of SQL Server 2008, select the radio button for free installations, and then use the drop-down box to identify which edition to install.

To validate a paid edition of SQL Server 2008, select the radio button for the Product Key, and then provide your key in the space provided.

For failover cluster installations, use the same product key for each failover cluster node. If the product key is from a different SQL Server edition, an error will occur.

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Progress (Uninstall)

-

<u>SQL Server Setup User Interface Reference</u> >

Use the Progress (Uninstall) page to monitor the Setup progess.



### Tasks

How to: Install SQL Server 2008 (Setup)

### Concepts

How to: View SQL Server 2008 Setup Log Files

SQL Server Setup Help Ready to Install

- -

<u>SQL Server Setup User Interface Reference</u> >

The **Ready to Install** page provides a summary of 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**.

### Tasks

How to: Read a SQL Server 2008 Setup Log File

### Concepts

How to: View SQL Server 2008 Setup Log Files

## SQL Server Setup Help

**Recovery Method** 

- -

<u>SQL Server Setup User Interface Reference</u> >

Make a selection, and then click **Next** to continue.

### SQL Server Setup Help Remove SQL Server Failover Cluster Node



-

<u>SQL Server Setup User Interface Reference</u> >

Use the Remove Failover Cluster Node page to identify the computer to remove as a SQL Server 2008 failover cluster node.

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Report Server Installation Options



#### - -

<u>SQL Server Setup User Interface Reference</u> >

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.

In some cases, if a Secure Sockets Layer (SSL) certificate is installed on the computer and is bound to a strong wildcard, Setup will create the Reporting Services URLs using the HTTPS prefix. For more information about how certificates are mapped to Reporting Services URLs, see **Configuring a Report Server for Secure Sockets Layer (SSL) Connections** in SQL Server Books Online.

### <sup>I</sup>Note:

An existing report server database file can cause setup to fail if you choose one of the default installation options. When you choose a 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.

# <sup>©</sup> Options

#### Install the native mode default configuration

Installs a report server instance using the default values for the report server databases, service account, and URL reservations. 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**. For more information about the default configuration for a native mode report server, see <u>Default Configuration for a Native Mode Installation (Reporting Services)</u>.

#### Install the SharePoint integrated mode default configuration

Installs a report server instance using the default values for the report server databases, service account, and URL reservations. The report server database is created in a format that supports content storage and addressing from a SharePoint site.

For more information about this installation option, see <u>Default</u> <u>Configuration for SharePoint Integrated Mode (Reporting Services)</u>.

#### Install but do not configure the server

Installs the report server program files, creates the Report Server service account, and registers the report server Windows Management Instrumentation (WMI) provider. This installation option is referred to as a "files-only" installation. Select this option if you do not want to use the default configuration. If the default configuration cannot be installed, or if you are installing a SQL Server failover cluster that includes Reporting Services, this is the only option available. For more information about a files-only installation, see <u>Files-Only Installation (Reporting Services)</u>.

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 <u>Creating a Report Server Database</u> and <u>Configuring a Report Server Database Connection</u>.

# Installing the Reporting Services Add-in for SharePoint Technologies

To install the Reporting Services Add-in for SharePoint Technologies, click the rsSharePoint.msi file in the Setup folder on SQL Server 2008 installation media after SQL Server Setup is complete.
## □ See Also

#### Tasks

<u>How to: Start Reporting Services Configuration</u> <u>How to: Create a Report Server Database (Reporting Services Configuration)</u>

#### Concepts

<u>Considerations for Installing Reporting Services</u> <u>Default Configuration for a Native Mode Installation (Reporting Services)</u>

SQL Server Setup Help Reserved Keywords (Setup)



#### - -

<u>SQL Server Setup User Interface Reference</u> >

SQL Server uses reserved keywords to define, manipulate, and add databases. Reserved keywords are part of the Transact-SQL grammar that is used by SQL Server to parse and understand Transact-SQL statements and batches.

SQL Server instance names cannot match a reserved keyword. Use the following table to ensure that reserved keywords are not used in SQL Server 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
СНЕСК	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	ΙΝΤΟ	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	то
CURRENT_USER	LOAD	ТОР
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 ISO standard defines a list of reserved keywords. Avoid using ISO reserved keywords for object names and identifiers. The ODBC reserved keyword list, shown in the following table, is the same as the ISO reserved keyword list.

## <sup>ℤ</sup>Note:

The ISO reserved keywords list sometimes can be more restrictive than SQL Server and at other times less restrictive. For example, the ISO 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	HOUR	SCHEMA
CASCADED	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	ТО
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		

## Tasks

How to: Install SQL Server 2008 (Setup)

SQL Server Setup Help Select Instance

#### - -

<u>SQL Server Setup User Interface Reference</u> >

Use the Select Instance page of the SQL Server Installation Wizard to specify the instance of SQL Server for this operation.

# <sup>©</sup> Options

Use the radio buttons to specify whether to upgrade the instance of SQL Server and its shared components, or to upgrade only the shared components. Shared components in this SQL Server release are the following:

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

For information about supported upgrade scenarios and upgrade known issues, see <u>Version and Edition Upgrades</u>.

#### Concepts

<u>Using Upgrade Advisor to Prepare for Upgrades</u> <u>Working with Multiple Versions and Instances of SQL Server</u>

SQL Server Setup Help SQL Server Configuration - Collation

# 

#### Ξ 💌

<u>SQL Server Setup User Interface Reference</u> >

On the Server Configuration - Collation page of the SQL Server Installation Wizard, you can modify collation settings that the Database Engine and Analysis Services use for sorting purposes. Select the option to match collation settings of different installations of SQL Server, or of another computer.

# • Options

Customize for SQL Server and Analysis Services

SQL Server provides two groups of collations: Windows collations and SQL Server collations. You can specify separate collation settings for the Database Engine and Analysis Services, or you can specify the same collation for both.

By default, a SQL Server collation is selected for US-English system locales. The default collation for localized versions of SQL Server is determined by the Windows system locale setting for your computer. For more information, see <u>Using SQL Server Collations</u>.

The default settings should be changed 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** Analysis Services uses Windows collations only. If you plan to install Analysis Services, select a Windows collation during SQL Server Setup to ensure consistent results between the SQL Server Database Engine and Analysis Services.

For more information, see <u>Collation Settings in Setup</u>.

# <sup>□</sup> Best Practices

For a table of Windows System locales and the corresponding default collations used by SQL Server Setup, see <u>Collation Settings in Setup</u>.

If it is possible, use a single collation for your organization. This way, you do not have to explicitly specify the collation for every database, column, expression, or identifier. If you must work with multiple collations and code page settings, code your queries to consider the rules of collation precedence. For more information see the Books Online topic for <u>Collation Precedence (Transact-SQL</u>).

When you select a collation for SQL Server, consider the following recommendations:

- Select a binary collation if binary ordering is acceptable.
- If your applications use both **nchar/nvarchar** and **char/varchar** data types and can mix them in comparison, select a Windows collation for consistent comparison across data types.
- If your applications use only **char/varchar** data types and you use Analysis Services, select a Windows collation.
- If your applications use only **char/varchar** data types and all of the following are also true, select a SQL Server collation:
  - You are concerned about string comparisons/LIKE performance.
  - You are not concerned about the linguistic comparison semantics in Windows collations.
  - You want the dictionary sort order in SQL Server collations.
  - You are not concerned about sorting consistency with the operating system or with other applications.

### Reference

Collation and Unicode Support Collation Settings in Setup

SQL Server Setup Help SQL Server Configuration - Service Accounts



-

<u>SQL Server Setup User Interface Reference</u> >

Use the Server Configuration page of the SQL Server Installation Wizard to assign login accounts to SQL Server services. The actual services that are configured on this page depend on the features you have selected to install.

# <sup>©</sup> 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, are started manually, or are disabled. Microsoft recommends that you configure service accounts individually to ensure that SQL Server services are granted the minimum required permissions to complete their tasks. For more information, see <u>Setting Up Windows Service Accounts</u>.

#### Configure SQL Server service accounts individually (recommended)

Use the grid to provision each SQL Server service with a logon user name and password, and to set the startup type for the service. You can use builtin system accounts, a local account, local group, domain group, or domain user accounts for SQL Server services.

For failover cluster installations, account name and startup type infromation will be pre-populated on this page based on settings provided for the active node. You must provide passwords for each account.

Select any of the following services to customize its settings.

Select this service	To configure authentication settings for	
SQL	The service that runs jobs, monitors, SQL Server, and allows automation of administrative tasks.	
Server Agent	There is no default login account for this service.	
	The default startup type is Manual.	
SQL Server	The SQL Server Database Engine. There is no default login account for this service.	
	The default startup type is Automatic.	
	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 Sorvor	SQL Server and Integration Services.
Browser	The default login account is NT Authority\Local service and cannot be changed.
	The default startup type is Automatic.
	Analysis Services.
Analysis Services	There is no default login account for this service.
	The default startup type is Automatic.
	Reporting Services.
	Service accounts are used to configure a report server database connection. Choose the built-in network service if you want to use default authentication settings. If you specify a domain user account, be sure to register a service principle name (SPN) for it if you are using Windows Authentication on the report server. For more information, see <b>How to: Configure Windows</b> <b>Authentication in Reporting Services</b> .
Reporting Services	✓ Important: Microsoft recommends that you do not use the Network Service account for the SQL Server or the SQL Server Agent services if an account with lesser privileges is available, because Network Service is a shareable account. Network Service is appropriate for use as a SQL Server service account only if you can ensure that no other services that use the account are installed on the computer. Local User or Domain User accounts that are not a Windows administrator are more appropriate for SQL Server services.
	There is no default login account for this service.
	The default startup type is Automatic.
	Integration Services is a set of graphical tools and programmable objects for moving, copying, and transforming data.

Integration Services	The default login account for this service is NT Authority\Network Service.
	The default startup type is Automatic.
	The service that creates the fdhost.exe processes. This is required to host the word breakers and filters that process textual data for full-text indexing.
SQL Full-	Provide an account in which to run the FDHOST Launcher service. We highly recommend that you use a low privilege account. This account should be different from the account that you use for the SQL Server service. On Windows Vista and Windows Server 2008, the FDHOST Launcher service account defaults to LOCAL SERVICE.
Daemon Launcher	For security reasons, on Windows versions earlier than Windows Vista and Windows Server 2008, we recommend using a specially-created LOCAL USER as the FDHOST Launcher service account. The use of the LOCAL SYSTEM, LOCAL SERVICE, or NETWORK SERVICE might inadvertently provide increased privileges for the service and can reduce the security of your SQL Server installation.
	The FDHOST Launcher service is started automatically unless the service account is not valid or you do not specify a service account (on Windows Server 2003 or Windows XP).

### Reference

Setting Up Windows Service Accounts

### Concepts

Security Considerations for a SQL Server Installation

SQL Server Setup Help SQL Server Failover Cluster Wizard - Complete



#### - -

<u>SQL Server Setup User Interface Reference</u> >

The following rules are used by SQL Server Setup to validate your computer configuration before the Setup operation finishes.

Check item	Description	User action
SSMS_IsInternetConnected	Verifies that the computer is connected to the Internet. When a .NET Framework application such as Microsoft Management Studio starts, there might be a slight delay while the .NET Framework security check validates a certificate.	Ensure that the computer is connected to the Internet.
AclPermissionsFacet	Checks if the SQL Server registry keys are consistent.	SQL Server registry keys must be consistent to support SQL Server installation or upgrade.
		You must install or

BlockMixedArchitectureUpgrade	Checks whether the CPU architecture of features you are trying to upgrade is different from the CPU architecture of this installation program.	upgrade by using SQL Server installation media that matches the architecture of the instance of SQL Server that you are modifying.
Cluster_IsOnlineIfClustered	Verifies that the computer is not clustered, or that the computer is clustered and the cluster service is online.	The computer must either not be clustered, or the cluster must be up and online.
FacetDomainControllerCheck	Checks whether the computer is a domain controller. Installing SQL Server 2008 on a domain controller is not recommended.	Ensure that this computer is not a domain controller. See <u>Security</u> <u>Considerations</u> <u>for a SQL</u> <u>Server</u> <u>Installation</u> .
FacetPowerShellCheck	Checks whether Windows PowerShell is installed. Windows PowerShell is a pre- requisite for some SQL Server 2008 editions.	For installations of SQL Server 2008 Express with Advanced Services, ensure that Windows PowerShell is

		installed.
FacetWOW64PlatformCheck	Determines whether SQL Server Setup is supported on this operating system platform.	This rule will block unsupported installations on the 64-bit platform. See <u>Hardware and</u> <u>Software</u> <u>Requirements</u> <u>for Installing</u> <u>SQL Server</u> <u>2008</u> .
FusionRebootCheck	Checks if a pending computer restart is required. A pending restart can cause Setup to fail.	A computer restart is required before installing SQL Server.
PerfMonCounterNotCorruptedCheck	Checks if the existing performance counter registry hive is consistent.	Ensure that the performance counter registry hive is consistent.
SqlUnsupportedProductBlocker	Checks whether SQL Server 7.0 or SQL Server 7.0 OLAP Services is installed. SQL Server 2008 is not supported side-by-side with SQL Server 7.0.	Remove SQL Server 7.0,, or install SQL Server 2008 on a different computer.

### Tasks

How to: Install SQL Server 2008 (Setup)

### Reference

Check Parameters for the System Configuration Checker

SQL Server Setup Help SQL Server Failover Cluster Wizard - Install



#### - -

<u>SQL Server Setup User Interface Reference</u> >

The following rules are used by SQL Server Setup to validate your computer configuration before the Setup operation completes.

Check item	Description	User action
SSMS_IsInternetConnected	Verifies that the computer is connected to the Internet. When a .NET Framework application such as Microsoft Management Studio starts, there might be a slight delay while the .NET Framework security check validates a certificate.	Ensure that the computer is connected to the Internet.
AclPermissionsFacet	Checks if the SQL Server registry keys are consistent.	SQL Server registry keys must be consistent to support SQL Server installation or upgrade.
		You must install or

BlockMixedArchitectureUpgrade	Checks whether the CPU architecture of features you are trying to upgrade is different from the CPU architecture of this installation program.	upgrade by using SQL Server installation media that matches the architecture of the instance of SQL Server that you are modifying.
Cluster_IsOnlineIfClustered	Verifies that the computer is not clustered, or that the computer is clustered and the cluster service is online.	The computer must either not be clustered, or the cluster must be up and online.
FacetDomainControllerCheck	Checks whether the computer is a domain controller. Installing SQL Server 2008 on a domain controller is not recommended.	Ensure that this computer is not a domain controller. See <u>Security</u> <u>Considerations</u> <u>for a SQL</u> <u>Server</u> <u>Installation</u> .
FacetPowerShellCheck	Checks whether Windows PowerShell is installed. Windows PowerShell is a pre- requisite for some SQL Server 2008 editions.	For installations of SQL Server 2008 Express with Advanced Services, ensure that Windows PowerShell is

		installed.
FacetWOW64PlatformCheck	Determines whether SQL Server Setup is supported on this operating system platform.	This rule will block unsupported installations on the 64-bit platform. See <u>Hardware and</u> <u>Software</u> <u>Requirements</u> <u>for Installing</u> <u>SQL Server</u> <u>2008</u> .
FusionRebootCheck	Checks if a pending computer restart is required. A pending restart can cause Setup to fail.	A computer restart is required before installing SQL Server.
PerfMonCounterNotCorruptedCheck	Checks if the existing performance counter registry hive is consistent.	Ensure that the performance counter registry hive is consistent.
SqlUnsupportedProductBlocker	Checks whether SQL Server 7.0 or SQL Server 7.0 OLAP Services is installed. SQL Server 2008 is not supported side-by-side with SQL Server 7.0.	Remove SQL Server 7.0, or install SQL Server 2008 on a different computer.

### Tasks

How to: Install SQL Server 2008 (Setup)

### Reference

Check Parameters for the System Configuration Checker

SQL Server Setup Help SQL Server Failover Cluster Wizard - Prepare



#### - -

<u>SQL Server Setup User Interface Reference</u> >

The following rules are used by SQL Server Setup to validate your computer configuration before the Setup operation finishes.

Check item	Description	User action
SSMS_IsInternetConnected	Verifies that the computer is connected to the Internet. When a .NET Framework application such as Microsoft Management Studio starts, there might be a slight delay while the .NET Framework security check validates a certificate.	Ensure that the computer is connected to the Internet.
AclPermissionsFacet	Checks if the SQL Server registry keys are consistent.	SQL Server registry keys must be consistent to support SQL Server installation or upgrade.
		You must install or

BlockMixedArchitectureUpgrade	Checks whether the CPU architecture of features you are trying to upgrade is different from the CPU architecture of this installation program.	upgrade by using SQL Server installation media that matches the architecture of the instance of SQL Server that you are modifying.
Cluster_IsOnlineIfClustered	Verifies that the computer is not clustered, or that the computer is clustered and the cluster service is online.	The computer must either not be clustered, or the cluster must be up and online.
FacetDomainControllerCheck	Checks whether the computer is a domain controller. Installing SQL Server 2008 on a domain controller is not recommended.	Ensure that this computer is not a domain controller. See <u>Security</u> <u>Considerations</u> <u>for a SQL</u> <u>Server</u> <u>Installation</u> .
FacetPowerShellCheck	Checks whether Windows PowerShell is installed. Windows PowerShell is a pre- requisite for some SQL Server 2008 editions.	For installations of SQL Server 2008 Express with Advanced Services, ensure that Windows PowerShell is

		installed.
FacetWOW64PlatformCheck	Determines whether SQL Server Setup is supported on this operating system platform.	This rule will block unsupported installations on the 64-bit platform. See <u>Hardware and</u> <u>Software</u> <u>Requirements</u> <u>for Installing</u> <u>SQL Server</u> <u>2008</u> .
FusionRebootCheck	Checks if a pending computer restart is required. A pending restart can cause Setup to fail.	A computer restart is required before installing SQL Server.
PerfMonCounterNotCorruptedCheck	Checks if the existing performance counter registry hive is consistent.	Ensure that the performance counter registry hive is consistent.
SqlUnsupportedProductBlocker	Checks whether SQL Server 7.0 or SQL Server 7.0 OLAP Services is installed. SQL Server 2008 is not supported side-by-side with SQL Server 7.0.	Remove SQL Server 7.0, or install SQL Server 2008 on a different computer.

### Tasks

How to: Install SQL Server 2008 (Setup)

### Reference

Check Parameters for the System Configuration Checker

SQL Server Setup Help Summary (Upgrade)

- -

<u>SQL Server Setup User Interface Reference</u> >

Use the Summary page to view the specified options for this upgrade.



## Tasks

How to: Install SQL Server 2008 (Setup)
SQL Server Setup Help System Configuration Check (SCC)

# 

#### - -

<u>SQL Server Setup User Interface Reference</u> >

The **System Configuration Check** page of the SQL Server Installation Wizard shows progress of the system validation tool. System Configuration Check routines scan the computer where SQL Server will be installed for conditions that might block Setup.

## <sup>©</sup> Options

The **Next** button is available only if all check results are successful, or if failed checks do not block installation.

Use the **Show details** button to display a summary of rules and scan results. The **Status** column at the right-hand side of the scan results shows the status of each item. Click the status for an item to view details.

The following table describes check parameters for the SCC, requirements, and resolution to blocking issues.

Check item	Description	User action
RebootRequiredCheck	Checks if a pending computer restart is required. A pending restart can cause Setup to fail.	A computer restart is required. You must restart this computer before installing SQL Server.
OsVersionCheck	Checks whether the computer meets minimum operating system version requirements.	Ensure that the operating system version meets the minimum requirements for this product. See <u>Hardware and</u> <u>Software</u> <u>Requirements for</u> <u>Installing SQL</u> <u>Server 2008</u> .
ThreadHasAdminPrivilegeCheck	Checks whether the account running SQL Server Setup has administrator rights on the computer.	Ensure that the account that is running SQL Server Setup has administrative rights on the computer.

WmiServiceStateCheck	Checks whether the WMI service is started and running on the computer.	Ensure that the Windows Management Instrumentation (WMI) service is running.
SSMS_IsInternetConnected	Checks to see if the computer is connected to the Internet. SQL Server Management Tools that are based on .NET Framework technology must check the certificate revocation list on startup.	Ensure that the computer is connected to the Internet.
FacetDomainControllerCheck	Checks whether the computer is a domain controller. Installing SQL Server 2008 on a domain controller is not recommended.	Ensure that this computer is not a domain controller. See <u>Security</u> <u>Considerations for</u> <u>a SQL Server</u> <u>Installation</u> .
FacetPowerShellCheck	Checks whether Windows PowerShell is installed. Windows PowerShell is a pre- requisite of SQL Server 2008 Express with Advanced Services.	For installations of Microsoft SQL Server 2008 Express with Advanced Services, ensure that Windows PowerShell is installed.
FacetWOW64PlatformCheck	Determine if the Setup program is supported on the operating system platform.	This rule will block unsupported installations on the 64-bit platform. See <u>Hardware and</u> <u>Software</u>

		Requirements for Installing SOL
		<u>Server 2008</u> .
PerfMonCounterNotCorruptedCheck	Checks if existing performance counter registry hive is consistent.	Ensure that the performance counter registry hive is consistent.
SqlUnsupportedProductBlocker	Checks whether SQL Server 7.0 or SQL Server 7.0 OLAP Services is installed.	Remove SQL Server 7.0 or install SQL Server 2008 on a different computer.

# <sup>□</sup> Additional Rule topics

See the follwing topics for scenario-specific Setup rules:

- <u>Feature Installation Rules</u>
- Upgrade and Repair Rules Check
  Edition Upgrade Rules
  Uninstallation rules

### $\Box$ See Also

### Tasks

How to: Install SQL Server 2008 (Setup)

#### Reference

<u>Check Parameters for the System Configuration Checker</u> <u>Hardware and Software Requirements for Installing SQL Server 2008</u>

SQL Server Setup Help Uninstallation rules

- -

<u>SQL Server Setup User Interface Reference</u> >

The Uninstallation Rules page will run a set of rules to ensure that the Setup operation can complete sucessfully.

# <sup>©</sup> Options

The following table describes check parameters for the System Configuration Checker, requirements, and resolution to blocking issues:

Check item	Description	User action
AclPermissionsFacet	Checks if the SQL Server registry keys are consistent.	SQL Server registry keys must be consistent to support SQL Server installation or upgrade.
OsVersionCheck	Checks whether the computer meets minimum operating system version requirements.	Ensure that the operating system version meets the minimum requirements for this product.
RebootRequiredCheck	Checks if a pending computer restart is required. A pending restart can cause Setup to fail.	A computer restart is required. You must restart this computer before installing SQL Server.
ThreadHasAdminPrivilegeCheck	Checks whether the account that is running SQL Server Setup has administrative privileges on the computer.	Ensure that the account running SQL Server Setup has administrative privileges on the computer.
	Checks whether the	Ensure that the

WmiServiceStateCheck	WMI service is started and running on the computer.	WMI service is running.
BlockMismatchedArchitectureUninstall	Checks whether the CPU architecture of the features you are trying to remove is different from the CPU architecture of this installation program.	Setup architecture must be the same as that of the instance of SQL Server that you are uninstalling.

# <sup>□</sup> Additional Rule topics

See the follwing topics for scenario-specific Setup rules:

- <u>Feature Installation Rules</u>
- Upgrade and Repair Rules Check
  Edition Upgrade Rules
  Uninstallation rules

### $\Box$ See Also

### Tasks

How to: Install SQL Server 2008 (Setup)

#### Reference

<u>Check Parameters for the System Configuration Checker</u> <u>Hardware and Software Requirements for Installing SQL Server 2008</u>

SQL Server Setup Help Upgrade and Repair Rules Check



#### -

<u>SQL Server Setup User Interface Reference</u> >

The following rules are used by SQL Server Setup to validate your computer configuration before the Setup operation completes.

Check item	Description	User action
SSMS_IsInternetConnected	Verifies that the computer is connected to the Internet. When a .NET Framework application such as Microsoft Management Studio starts, there might be a slight delay while the .NET Framework security check validates a certificate.	Ensure that the computer is connected to the Internet.
AclPermissionsFacet	Checks if the SQL Server registry keys are consistent.	SQL Server registry keys must be consistent to support SQL Server installation or upgrade.
		You must install or

BlockMixedArchitectureUpgrade	Checks whether the CPU architecture of features you are trying to upgrade is different from the CPU architecture of this installation program.	upgrade by using SQL Server installation media that matches the architecture of the instance of SQL Server that you are modifying.
Cluster_IsOnlineIfClustered	Verifies that the computer is not clustered, or that the computer is clustered and the cluster service is online.	The computer must either not be clustered, or the cluster must be up and online.
FacetDomainControllerCheck	Checks whether the computer is a domain controller. Installing SQL Server 2008 on a domain controller is not recommended.	Ensure that this computer is not a domain controller. See <u>Security</u> <u>Considerations</u> <u>for a SQL</u> <u>Server</u> <u>Installation</u> .
FacetPowerShellCheck	Checks whether Windows PowerShell is installed. Windows PowerShell is a pre- requisite for some SQL Server 2008 editions.	For installations of SQL Server 2008 Express with Advanced Services, ensure that Windows PowerShell is

		installed.
FacetWOW64PlatformCheck	Determines whether SQL Server Setup is supported on this operating system platform.	This rule will block unsupported installations on the 64-bit platform. See <u>Hardware and</u> <u>Software</u> <u>Requirements</u> <u>for Installing</u> <u>SQL Server</u> <u>2008</u> .
FusionRebootCheck	Checks if a pending computer restart is required. A pending restart can cause Setup to fail.	A computer restart is required. You must restart this computer before installing SQL Server.
PerfMonCounterNotCorruptedCheck	Checks if the existing performance counter registry hive is consistent.	Ensure that the performance counter registry hive is consistent.
SqlUnsupportedProductBlocker	Checks whether SQL Server 7.0 or SQL Server 7.0 OLAP Services is installed. SQL Server 2008 is not supported side-by-side	Remove SQL Server 7.0 or install SQL Server 2008 on a different

	with SQL Server 7.0.	computer.
--	----------------------	-----------

## $\Box$ See Also

### Reference

Check Parameters for the System Configuration Checker

SQL Server Setup Help Verify Uninstall Options

Ξ 💙

<u>SQL Server Setup User Interface Reference</u> >

Verify your selection, and then click **Next** to continue.