Contents

SQL Server Installation

Planning a SQL Server Installation

- What's New in SQL Server Installation
- Hardware and Software Requirements for Installing SQL Server 2014
- Security Considerations for a SQL Server Installation
- Network Protocols and Network Libraries
- Work with Multiple Versions and Instances of SQL Server
- Local Language Versions in SQL Server
- File Locations for Default and Named Instances of SQL Server

Install SQL Server 2014

- About the SQL Server Database Engine
- Install Distributed Replay
- Distributed Replay Requirements
- Install Distributed Replay (Setup)
- Install Distributed Replay from the Command Prompt
- Install Distributed Replay Using a Configuration File
- Complete the Post-Installation Steps
- Modify the Controller and Client Services Accounts
- Repair a Distributed Replay Installation
- Uninstall an Existing Instance of Distributed Replay
- Install SQL Server Replication
- Install SQL Server Management Tools
- Install SQL Server PowerShell
- Install SQL Server 2014 on Server Core
- Install SQL Server with SMB Fileshare as a Storage Option
- Considerations for Installing SQL Server Using SysPrep
- Validate a SQL Server Installation
- Check Parameters for the System Configuration Checker
- Configure SQL Server on a Server Core Installation

- Configure the Windows Firewall to Allow SQL Server Access
- Configure a Multi-Homed Computer for SQL Server Access
- Installation How-to Topics
- Install SQL Server 2014 from the Installation Wizard (Setup)
- Install SQL Server 2014 from the Command Prompt
- Install SQL Server 2014 Using a Configuration File
- Install SQL Server 2014 Using SysPrep
- Add Features to an Instance of SQL Server 2014 (Setup)
- Drop a SQL Server 2014 Installation
- Rename a Computer that Hosts a Stand-Alone Instance of SQL Server
- Install SQL Server 2014 Servicing Updates
- View and Read SQL Server Setup Log Files

Install SQL Server 2014 Business Intelligence Features

- Install Analysis Services in Multidimensional and Data Mining Mode
- Install Analysis Services in Tabular Mode
- Install Data Quality Services
- Run DQSInstaller.exe to Complete Data Quality Server Installation
- Grant DQS Roles to Users
- Access Data for the DQS Operations
- Export and Import DQS Knowledge Bases Using DQSInstaller.exe
- Upgrade SQLCLR Assemblies After .NET Framework Update
- Upgrade DQS Databases Schema After Installing SQL Server Update
- Install Integration Services
- Interoperability and Coexistence
- Install Master Data Services
- Web Application Requirements (Master Data Services)
- Database Requirements (Master Data Services)
- Create a Master Data Services Database
- Create a Master Data Manager Web Application (Master Data Services)
- Secure a Master Data Manager Web Application
- Associate a Master Data Services Database and Web Application
- Enable Data Quality Services Integration with Master Data Services

- Multi-Lingual and Global Deployments (Master Data Services)
- Install Reporting Services Native Mode Report Server
- Perform a Files-Only Installation of Reporting Services
- Install Reporting Services and Internet Information Services Side-by-Side (SSRS Native Mode)
- Host a Report Server Database in a SQL Server Failover Cluster
- Reporting Services Configuration Manager (Native Mode)
  - Configure the Report Server Service Account (SSRS Configuration Manager)
  - Configure a Service Account (SSRS Configuration Manager)
  - Configure Report Server URLs (SSRS Configuration Manager)
  - About URL Reservations and Registration (SSRS Configuration Manager)
  - Configure a URL (SSRS Configuration Manager)
  - URL Reservation Syntax (SSRS Configuration Manager)
  - URLs in Configuration Files (SSRS Configuration Manager)
  - URL Reservations for Multi-Instance Report Server Deployments (SSRS Configuration Manager)
  - Create a Report Server Database (SSRS Configuration Manager)
  - Create a Native Mode Report Server Database (SSRS Configuration Manager)
  - Configure a Report Server Database Connection
  - Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager)
  - Configure the Unattended Execution Account (SSRS Configuration Manager)
  - Configure and Manage Encryption Keys (SSRS Configuration Manager)
  - Initialize a Report Server (SSRS Configuration Manager)
  - Store Encrypted Report Server Data (SSRS Configuration Manager)
  - Back Up and Restore Encryption Keys
  - Delete and Re-create Encryption Keys (SSRS Configuration Manager)
  - Add and Remove Encryption Keys for Scale-Out Deployment (SSRS Configuration Manager)
  - Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager)
  - Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
  - Connect to a Native Mode Report Server
  - Report Server Status (SSRS Native Mode)

Service Account (SSRS Native Mode)

Web Service URL (SSRS Native Mode)

Advanced Multiple Web Site Configuration (SSRS Native Mode)

Database (SSRS Native Mode)

Change Database Wizard (SSRS Native Mode)

Change Credentials Wizard (SSRS Native Mode)

Report Manager URL (SSRS Native Mode)

E-mail Settings - Configuration Manager (SSRS Native Mode)

Execution Account (SSRS Native Mode)

Encryption Keys (SSRS Native Mode)

Backup Encryption Key (SSRS Native Mode)

Restore Encryption Key (SSRS Native Mode)

Scale-out Deployment (Native Mode Report Server)

Install SQL Server BI Features with SharePoint (PowerPivot and Reporting Services)

Deployment Topologies for SQL Server BI Features in SharePoint

Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm

Checklists for Installing BI Features with SharePoint

Checklist: Reporting Services, Power View, and PowerPivot for SharePoint

Checklist: Scale-out by adding PowerPivot Servers to a SharePoint 2010 farm

Checklist: Install Reporting Services into an Existing SharePoint Farm

Checklist: Multi-Server Installation of PowerPivot for SharePoint 2010

CheckList: Use PowerShell to Verify PowerPivot for SharePoint

Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013)

Hardware and Software Requirements for Reporting Services in SharePoint Mode

Supported Combinations of SharePoint and Reporting Services Server and Add-in

Install Reporting Services for SharePoint 2013

Install Reporting Services for SharePoint 2010

Install or Uninstall the Reporting Services Add-in for SharePoint

Where to find the SSRS add-in for SharePoint

Add an Additional Report Server to a Farm (SSRS Scale-out)

Add an Additional Reporting Services Web Front-end to a Farm

Configure E-mail for a Reporting Services Service Application

Provision Subscriptions and Alerts for SSRS Service Applications

Claims to Windows Token Service (C2WTS) and SSRS

PowerPivot for SharePoint 2013 Installation

Hardware and Software Requirements for Analysis Services Server in SharePoint Mode (SQL Server 2014)

Example of a Minimum-Privilege Configuration for PowerPivot for SharePoint 2013

Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013)

Configure PowerPivot and Deploy Solutions (SharePoint 2013)

Migrate PowerPivot to SharePoint 2013

Upgrade Workbooks and Scheduled Data Refresh (SharePoint 2013)

PowerPivot for SharePoint 2010 Installation

Install PowerPivot for SharePoint 2010

Install the Analysis Services OLE DB Provider on SharePoint Servers

Install ADO.NET Data Services to support data feed exports of SharePoint lists

Install ADOMD.NET on Web Front-End Servers Running Central Administration

Install PowerPivot from the Command Prompt

Repair PowerPivot for SharePoint

Initial Configuration for a PowerPivot Single-Server Installation

Command Prompt Installation of Reporting Services SharePoint Mode and Native Mode

Verify or Troubleshoot an Installation

Verify a Reporting Services Installation

Troubleshoot a Reporting Services Installation

Verify a PowerPivot for SharePoint Installation

Troubleshoot a PowerPivot for SharePoint Installation

Upgrade to SQL Server 2014

Supported Version and Edition Upgrades

Are you upgrading from SQL Server 2005?

Use Upgrade Advisor to Prepare for Upgrades

Use the Distributed Replay Utility to Prepare for Upgrades

Upgrade Analysis Services

Upgrade Database Engine

Migrate Query Plans

- Upgrade Data Quality Services
- Upgrade Integration Services
- Upgrade Integration Services Packages
- Upgrade Integration Services Packages Using the SSIS Package Upgrade Wizard
- Migrate Scripts to VSTA
- Upgrade Lookup Transformations
- Upgrade Master Data Services
- Upgrade PowerPivot for SharePoint
- Upgrade Replicated Databases
- Upgrade and Migrate Reporting Services
- Migrate a Reporting Services Installation (Native Mode)
- Migrate a Reporting Services Installation (SharePoint Mode)
- Native to SharePoint Migration (SSRS)
- Upgrade a Report Server Database
- Upgrade Reports
- Backup and Restore Operations for Reporting Services
- Upgrade SQL Server Management Tools
- Upgrade How-to Topics
- Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)
- Upgrade to a Different Edition of SQL Server 2014 (Setup)

Uninstall SQL Server 2014

- Uninstall an Existing Instance of SQL Server (Setup)
- Remove Data Quality Server Objects
- Uninstall and Remove Master Data Services
- Uninstall PowerPivot for SharePoint
- Uninstall Reporting Services

Installation for SQL Server 2014

12/16/2019 • 2 minutes to read • Edit Online

Download SQL Server 2014 Express

Thank you to Scott Hanselman for collecting all of the installer package links in one place!

The SQL Server Installation Wizard provides a single feature tree to install all SQL Server components:

- Database Engine
- Analysis Services
- Reporting Services
- Integration Services
- Master Data Services
- Data Quality Services
- Management tools
- Connectivity components

You can install each component individually or select a combination of the components listed above. To make the best choice among the editions and components available in SQL Server, see Editions and Components of SQL Server 2014 and Features Supported by the Editions of SQL Server 2014. SQL Server 2014 is available in 32-bit and 64-bit editions.

Try it out:

- Have an Azure account? Then go **Here** to spin up a Virtual Machine with SQL Server 2014 Service Pack 1 (SP1) already installed. For more information on SQL Server 2014 (SP1), see SQL Server 2014 Service Pack 1 release information.

In This Section

Regardless of whether you use the SQL Server Installation Wizard or the command prompt to install SQL Server, Setup involves the following steps:

Planning a SQL Server Installation
Describes how to prepare your computer for SQL Server:

- Hardware and software requirements.
- System Configuration Checker requirements and blocking issues.
- Security considerations.

Install SQL Server 2014
Describes installation options for SQL Server.

Upgrade to SQL Server 2014
Describes options for upgrading to SQL Server.

Uninstall SQL Server 2014

Describes procedures to uninstall SQL Server and Master Data Services.

SQL Server Failover Cluster Installation
This section of the SQL Server Setup documentation explains how to install, and configure SQL Server failover cluster.

Install SQL Server 2014 BI Features
SQL Server features that are part of the Microsoft BI platform include Analysis Services, Integration Services, Reporting Services, and several client applications used for creating or working with analytical data. This section of the SQL Server Setup documentation explains how to install Analysis Services and Reporting Services.

Related Sections

Installation How-to Topics
Provides links to procedural topics for installing SQL Server 2014 from the installation wizard, from the command prompt, by using configuration files, and by using SysPrep.

Install SQL Server BI Features with SharePoint (PowerPivot and Reporting Services)
This section explains how to install SQL Server features in a SharePoint environment. It identifies which SQL Server features are available given a specific version and edition of SharePoint. It also includes installation procedures for PowerPivot for SharePoint and Reporting Services in SharePoint mode.

Installing SQL Server Samples and Sample Databases
Describes how to install and configure SQL Server samples and sample databases.

See Also

What's New in SQL Server Installation
Hardware and Software Requirements for Installing SQL Server 2014

Planning a SQL Server Installation

12/13/2018 • 2 minutes to read • Edit Online

To install SQL Server, follow these steps:

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

Regardless of the installation method, you are required to confirm acceptance of the software license terms as an individual or on behalf of an entity, unless your use of the software is governed by a separate agreement such as a Microsoft volume licensing agreement or a third-party agreement with an ISV or OEM.

The license terms are displayed for review and acceptance in the Setup user interface. Unattended installations (using the /Q or /QS parameters) must include the /IAcceptSQLServerLicenseTerms parameter. You can review the license terms separately at Microsoft Software License Terms.

> **NOTE**
> Depending on how you received the software (for example, through Microsoft volume licensing), your use of the software may be subject to additional terms and conditions.

In This Section

What's New in SQL Server Installation
This topic describes the details about the new or improved features of installation in this version of SQL Server 2014.

Hardware and Software Requirements for Installing SQL Server 2014
This topic lists the minimum hardware and software requirements to install and run an instance of SQL Server 2014.

Security Considerations for a SQL Server Installation
This topic describes some security best practices that you should consider before you install SQL Server and after you install SQL Server.

Configure Windows Service Accounts and Permissions
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 and after SQL Server installation.

Network Protocols and Network Libraries
This topic describes the default configuration of network protocols in this release of SQL Server, and the configuration options available.

Work with Multiple Versions and Instances of SQL Server
This topic describes the considerations for installing multiple versions and instances of SQL Server.

Local Language Versions in SQL Server
This topic describes about the localized versions of SQL Server 2014.

Related Sections

Install SQL Server 2014

This section provides an overview of different installation options we have for installing SQL Server 2014.

Install SQL Server 2014 BI Features

This section of the SQL Server Setup documentation explains how to install SQL Server features that are part of the Microsoft BI platform.

Upgrade to SQL Server 2014

The section provides an overview of upgrading instances of previous SQL Server versions to SQL Server 2014.

Uninstall SQL Server 2014

Refer this section to uninstall an existing instance of SQL Server 2014 completely, and prepare the system so that you can reinstall SQL Server.

SQL Server Failover Cluster Installation

This section of the SQL Server Setup documentation explains how to install, and configure SQL Server failover cluster.

See Also

Quick-Start Installation of SQL Server 2014
Install SQL Server 2014 from the Command Prompt
High Availability Solutions (SQL Server)
Before Installing Failover Clustering
Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)

What's New in SQL Server Installation

12/3/2018 • 2 minutes to read • Edit Online

Windows Vista is not a supported operating system for SQL Server 2014. Service Pack 1 remains as the minimum requirement for Windows 7 and Windows Server 2008 R2 operating systems. For more information on operating system requirements, see Hardware and Software Requirements for Installing SQL Server 2014.

Installation of SQL Server 2014 Express will prompt you to specify the directory to save the extracted package. If no location is entered, SQL Server 2014 will default to the computer's system drive. The extracted files will remain after SQL Server 2014 Express installation is complete.

In SQL Server 2014, SysPrep is supported for all installations of SQL Server. SysPrep now supports failover cluster installations. For more information, see Considerations for Installing SQL Server Using SysPrep and Install SQL Server 2014 Using SysPrep.

Upgrade from SQL Server 2005 is supported, but side-by-side is not supported. For more information about SQL Server 2005 support in SQL Server 2014, see Supported Version and Edition Upgrades.

Beginning in SQL Server 2014, the database engine in the Standard edition has a memory capacity of 128 GB. In SQL Server 2012, the database engine in the Standard edition had a memory capacity of 64 GB.

See Also

What's New in SQL Server 2014
Product Specifications for SQL Server 2014
Planning a SQL Server Installation
Hardware and Software Requirements for Installing SQL Server 2014

SQL Server 2014: Hardware and Software Requirements

3/16/2020 • 22 minutes to read • Edit Online

- Try SQL Server 2016 by installing the **Free Developer edition**!

Considerations

- We recommend running SQL Server 2014 on computers with the NTFS file format. FAT32 file system is supported but not recommended as it is less secure than the NTFS file system.
- You cannot install on read-only, mapped, or compressed drives.
- To allow the Visual Studio component to install correctly, SQL Server requires you to install an update. SQL Server Setup checks for the presence of this update then requires you to download and install that update before you can continue with the SQL Server installation. To avoid the interruption during SQL Server Setup, download and install the update **before running SQL Server Setup** as described below (or install all the updates for .NET 3.5 SP1 available on Windows Update):
  - For Windows Server 2008 SP2, get the required update here.
  - For any other supported operating system, this update is included.
- Launching SQL Server Setup through Terminal Services Client is not supported. Installation fails if you launch setup through Terminal Services Client.
- SQL Server Setup installs the following software components required by the product:
  - SQL Server Native Client
  - SQL Server Setup support files
- For minimum version requirements to install SQL Server on Windows Server 2012 or Windows 8, see Installing SQL Server on Windows Server 2012 or Windows 8 (https://support.microsoft.com/kb/2681562).

This topic contains the following sections:

- Hardware and Software Requirements
- Processor, Memory, and Operating System Requirements
- Cross-Language Support
- Extended System Support
- Hard Disk Space Requirements (32-Bit and 64 Bit)
- Storage Types for Data Files
- Installing SQL Server on a Domain Controller

Hardware and Software Requirements

The table in this section lists the minimum requirements for running SQL Server. There are also recommended

configuration options for optimal performance.

The following software requirements apply to all installations:

COMPONENT	REQUIREMENT
.NET Framework	.NET 3.5 SP1 is a requirement for SQL Server 2014 when you select Database Engine, Reporting Services, Master Data Services, Data Quality Services, Replication, or SQL Server Management Studio, and it is no longer installed by SQL Server Setup. -If you run Setup and you do not have .NET 3.5 SP1, SQL Server Setup requires you to download and install .NET 3.5 SP1 before you can continue with the SQL Server installation. (Install .NET 3.5 SP1 from Microsoft .NET Framework 3.5 Service Pack 1.) The error message includes a link to the download center, or you can download .NET 3.5 SP1 from Windows Update. To avoid interruption during SQL Server Setup, you can download and install .NET 3.5 SP1 before you run SQL Server Setup. -If you run Setup on a computer with Windows Server 2008 R2 SP1 or Windows 8, you must enable .NET Framework 3.5 SP1 before you install SQL Server 2014. -If there is no internet access, you must download and install .NET Framework 3.5 SP1 before you run Setup to install any of the above mentioned components. For more information about the recommendations and guidance on how to acquire and enable .NET Framework 3.5 on Windows 8 and Windows Server 2012, see Microsoft .NET Framework 3.5 Deployment Considerations (https://msdn.microsoft.com/library/windows/hardware/hh975396). .NET 4.0 is a requirement for SQL Server 2014. SQL Server installs .NET 4.0 during the feature installation step. -If you are installing the SQL Server Express editions, ensure that an Internet connection is available on the computer. SQL Server Setup downloads and installs the .NET Framework 4 because it is not included in the SQL Server Express media. -SQL Server Express does not install .NET 4.0 on the Server Core mode of Windows Server 2008 R2 SP1 or Windows Server 2012. You must install .NET 4.0 before you install SQL Server Express on a Server Core installation of Windows Server 2008 R2 SP1 or Windows Server 2012.
Windows PowerShell	SQL Server 2014 does not install or enable Windows PowerShell 2.0; however Windows PowerShell 2.0 is an installation prerequisite for Database Engine components and SQL Server Management Studio. If Setup reports that Windows PowerShell 2.0 is not present, you can install or enable it by following the instructions on the Windows Management Framework page.

COMPONENT	REQUIREMENT
Network Software	Supported operating systems for SQL Server 2014 have built-in network software. Named and default instances of a stand-alone installation support the following network protocols: Shared memory, Named Pipes, TCP/IP and VIA. Note: VIA protocol is not supported on failover clusters. Clients or applications running on the same node of the failover cluster as the SQL Server instance, can use Shared Memory protocol to connect to SQL Server using its local pipe address. However this type of connection is not cluster-aware and will fail after an instance failover. It is therefore not recommended and should only be used in very specific scenarios. The VIA protocol is deprecated. This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. For more information about Network Protocols and Network Libraries, see Network Protocols and Network Libraries.

COMPONENT	REQUIREMENT
Virtualization	SQL Server 2014 is supported in virtual machine environments running on the Hyper-V role in: - Windows Server 2008 SP2 Standard, Enterprise and Datacenter editions -Windows Server 2008 R2 SP1 Standard, Enterprise, and Datacenter editions. - Windows Server 2012 Datacenter and Standard editions. 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 2014 instance. Requirements are listed later in this topic.* Within the Hyper-V role on Windows Server 2008 SP2 or Windows Server 2008 R2 SP1, a maximum of 4 (four) virtual processors can be allocated to virtual machines running Windows Server 2008 SP2 32-bit/64-bit or Windows Server 2008 R2 SP1 64-bit or Windows Server 2012 64-bit editions. Within the Hyper-V role on Windows Server 2012: A maximum of 8 (eight) virtual processors can be allocated to virtual machines running Windows Server 2008 SP2 32-bit/64-bit. A maximum of 64 (sixty-four) virtual processors can be allocated to virtual machines running Windows Server 2008 R2 SP1 64-bit or Windows Server 2012 64-bit editions. For more information on compute capacity limits for different editions of SQL Server 2014 and how they differ in physical and virtualized environments with hyperthreaded processors, see Compute Capacity Limits by Edition of SQL Server. For more information about the Hyper-V role, see the Windows Server 2008 Web site. ** Important ** Guest failover clustering is supported in SQL Server 2014. For more information about the supported versions of SQL Server and operating systems for guest failover clustering, and the support for virtualization, see Support policy for Microsoft SQL Server products running in a hardware virtual environment.
Hard Disk	SQL Server 2014 requires a minimum of 6 GB of available hard-disk space. Disk space requirements will vary with the SQL Server 2014 components you install. For more information, see Hard Disk Space Requirements (32-Bit and 64 Bit) later in this topic. For information on supported storage types for data files, see Storage Types for Data Files.
Drive	A DVD drive, as appropriate, is required for installation from disc.
Monitor	SQL Server 2014 requires Super-VGA (800x600) or higher resolution monitor.

COMPONENT	REQUIREMENT
Internet	Internet functionality requires Internet access (fees may apply).

*Running SQL Server 2014 on a virtual machine will be slower than running natively because of the overhead of virtualization.

Processor, Memory, and Operating System Requirements

The following memory and processor requirements apply to all editions of SQL Server 2014:

COMPONENT	REQUIREMENT
Memory[1]	**Minimum:** Express Editions: 512 MB All other editions: 1 GB **Recommended:** Express Editions: 1 GB All other editions: At least 4 GB and should be increased as database size increases to ensure optimal performance.
Processor Speed	**Minimum:** x86 Processor: 1.0 GHz x64 Processor: 1.4 GHz **Recommended:** 2.0 GHz or faster
Processor Type	x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support x86 Processor: Pentium III-compatible processor or faster

[1]The minimum memory required for installing the Data Quality Server component in Data Quality Services (DQS) is 2 GB of RAM, which is different from the SQL Server 2014 minimum memory requirement. For information about installing DQS, see Install Data Quality Services.

WOW64 Support:

WOW64 (Windows 32-bit on Windows 64-bit) is 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 a 64-bit operating system.

- On a supported 64-bit operating system, SQL Server 32-bit edition can be installed to the WOW64 32-bit subsystem of a 64-bit server. WOW64 is supported only for stand-alone instances of SQL Server. WOW64 is not supported for SQL Server failover cluster installations.
- For SQL Server 64-bit edition installations on supported 64-bit operating systems, Management Tools are supported in WOW64. For more information about supported operating systems, select an edition of SQL Server 2014 from the sections below.

Server Core Support:

SQL Server 2014 is now supported on a Server Core installation of Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019.

Installing SQL Server 2014 on Server Core mode is supported by the following editions of Windows Server:

Windows Server 2019 Standard	Windows Server 2019 Datacenter
Windows Server 2016 Standard	Windows Server 2016 Datacenter
Windows Server 2012 R2 Standard	Windows Server 2012 R2 Datacenter
Windows Server 2012 Standard	Windows Server 2012 Datacenter
Windows Server 2008 R2 SP1 Standard	Windows Server 2008 R2 SP1 Datacenter
Windows Server 2008 R2 SP1 Enterprise	Windows Server 2008 R2 SP1 Web

For more information on installing SQL Server 2014 on Server Core, see Install SQL Server 2014 on Server Core.

> **NOTE**
> SQL Server editions that are supported on Windows Server 2008 R2 64-bit x64 Standard edition are also supported on Windows Small Business Server 64-bit x64.

Operating System Support:

The SQL Server 2014 editions are classified into the following:

- Principal Editions of SQL Server 2014
- Specialized Editions of SQL Server 2014
- Breadth Editions of SQL Server 2014

Principal Editions Operating Systems requirements

The following table shows the operating system requirements for the principal editions of SQL Server 2014:

SQL SERVER EDITION	32-BIT	64-BIT
SQL Server Enterprise	Windows Server 2019 Datacenter	Windows Server 2019 Datacenter
	Windows Server 2019 Standard	Windows Server 2019 Standard
	Windows Server 2016 Datacenter	Windows Server 2016 Datacenter
	Windows Server 2016 Standard	Windows Server 2016 Standard
	Windows Server 2012 R2 Datacenter 64-bit	Windows Server 2012 R2 Datacenter 64-bit
	Windows Server 2012 R2 Standard	Windows Server 2012 R2 Standard 64-bit

SQL SERVER EDITION	32-BIT	64-BIT
	64-bit Windows Server 2012 R2Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 32-bit Windows Server 2008 SP2 Standard 32-bit Windows Server 2008 SP2 Web 32-bit	Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Web 64-bit
SQL Server Business Intelligence	Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2012 R2 Datacenter 64-bit	Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2012 R2 Datacenter 64-bit

SQL SERVER EDITION	32-BIT	64-BIT
	Windows Server 2012 R2 Standard 64-bit Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 32-bit Windows Server 2008 SP2 Standard 32-bit Windows Server 2008 SP2 Web 32-bit	Windows Server 2012 R2 Standard 64-bit Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Web 64-bit
SQL Server Standard	Windows 10 Home 64-bit Windows 10 Professional 64-bit Windows 10 Enterprise 64-bit Windows 10 Home 32-bit	Windows 10 Home 64-bit Windows 10 Professional 64-bit Windows 10 Enterprise 64-bit Windows Server 2019 Datacenter Windows Server 2019 Standard

SQL SERVER EDITION	32-BIT	64-BIT
	Windows 10 Professional 32-bit	Windows Server 2016 Datacenter
	Windows 10 Enterprise 32-bit	Windows Server 2016 Standard
	Windows Server 2019 Datacenter	Windows Server 2012 R2 Datacenter 64-bit
	Windows Server 2019 Standard	Windows Server 2012 R2 Standard 64-bit
	Windows Server 2016 Datacenter	Windows Server 2012 R2 Essentials 64-bit
	Windows Server 2016 Standard	Windows Server 2012 R2 Foundation 64-bit
	Windows Server 2012 R2 Datacenter 64-bit	Windows Server 2012 Datacenter 64-bit
	Windows Server 2012 R2 Standard 64-bit	Windows Server 2012 Standard 64-bit
	Windows Server 2012 R2 Essentials 64-bit	Windows Server 2012 Essentials 64-bit
	Windows Server 2012 R2 Foundation 64-bit	Windows Server 2012 Foundation 64-bit
	Windows Server 2012 Datacenter 64-bit	Windows Server 2008 R2 SP1 Datacenter 64-bit
	Windows Server 2012 Standard 64-bit	Windows Server 2008 R2 SP1 Enterprise 64-bit
	Windows Server 2012 Essentials 64-bit	Windows Server 2008 R2 SP1 Standard 64-bit
	Windows Server 2012 Foundation 64-bit	Windows Server 2008 R2 SP1 Foundation 64-bit
	Windows Server 2008 R2 SP1 Datacenter 64-bit	Windows Server 2008 R2 SP1 Web 64-bit
	Windows Server 2008 R2 SP1 Enterprise 64-bit	Windows 8.1 64-bit
	Windows Server 2008 R2 SP1 Standard 64-bit	Windows 8.1 Pro 64-bit
	Windows Server 2008 R2 SP1 Foundation 64-bit	Windows 8.1 Enterprise 64-bit
	Windows Server 2008 R2 SP1 Web 64-bit	Windows 8 64-bit
	Windows 8.1 32-bit	Windows 8 Pro 64-bit
	Windows 8.1 Pro 32-bit	Windows 8 Enterprise 64-bit
	Windows 8.1 Enterprise 32-bit	Windows 7 SP1 Ultimate 64-bit
	Windows 8.1 64-bit	Windows 7 SP1 Enterprise 64-bit
	Windows 8.1 Pro 64-bit	Windows 7 SP1 Professional 64-bit
	Windows 8.1 Enterprise 64-bit	Windows Server 2008 SP2 Datacenter 64-bit
	Windows 8 32-bit	Windows Server 2008 SP2 Enterprise
	Windows 8 Pro 32-bit	

SQL SERVER EDITION	32-BIT	64-BIT
	Windows 8 Enterprise 32-bit Windows 8 64-bit Windows 8 Pro 64-bit Windows 8 Enterprise 64-bit Windows 7 SP1 Ultimate 64-bit Windows 7 SP1 Enterprise 64-bit Windows 7 SP1 Professional 64-bit Windows 7 SP1 Ultimate 32-bit Windows 7 SP1 Enterprise 32-bit Windows 7 SP1 Professional 32-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Foundation 64-bit Windows Server 2008 SP2 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 32-bit Windows Server 2008 SP2 Standard 32-bit Windows Server 2008 SP2 Web 32-bit	Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Foundation 64-bit Windows Server 2008 SP2 Web 64-bit

Specialized Editions Operating System requirements

The following table shows the operating system requirements for the specialized editions of SQL Server 2014:

SQL SERVER EDITION	32-BIT	64-BIT
SQL Server Web	Windows 10 Home 64-bit Windows 10 Professional 64-bit Windows 10 Enterprise 64-bit Windows 10 Home 32-bit Windows 10 Professional 32-bit Windows 10 Enterprise 32-bit	Windows 10 Home 64-bit Windows 10 Professional 64-bit Windows 10 Enterprise 64-bit Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2016 Datacenter

SQL SERVER EDITION	32-BIT	64-BIT
	Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2012 R2 Datacenter 64-bit Windows Server 2012 R2 Standard 64-bit Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 32-bit Windows Server 2008 SP2 Standard 32-bit Windows Server 2008 SP2 Web 32-	Windows Server 2016 Standard Windows Server 2012 R2 Datacenter 64-bit Windows Server 2012 R2 Standard 64-bit Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Web 64-bit

SQL SERVER EDITION	32-BIT	64-BIT
	bit	

Breadth Editions Operating System requirements

The following table shows the operating system requirements for the breadth editions of SQL Server 2014:

SQL SERVER EDITION	32-BIT	64-BIT
SQL Server Developer	Windows 10 Home 64-bit Windows 10 Professional 64-bit Windows 10 Enterprise 64-bit Windows 10 Home 32-bit Windows 10 Professional 32-bit Windows 10 Enterprise 32-bit Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2012 R2 Datacenter 64-bit Windows Server 2012 R2 Standard 64-bit Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows 8.1 32-bit	Windows 10 Home 64-bit Windows 10 Professional 64-bit Windows 10 Enterprise 64-bit Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2012 R2 Datacenter 64-bit Windows Server 2012 R2 Standard 64-bit Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows 8.1 64-bit Windows 8.1 Pro 64-bit Windows 8.1 Enterprise 64-bit Windows 8 64-bit

SQL SERVER EDITION	32-BIT	64-BIT
	Windows 8.1 Pro 32-bit Windows 8.1 Enterprise 32-bit Windows 8.1 64-bit Windows 8.1 Pro 64-bit Windows 8.1 Enterprise 64-bit Windows 8 32-bit Windows 8 Pro 32-bit Windows 8 Enterprise 32-bit Windows 8 64-bit Windows 8 Pro 64-bit Windows 8 Enterprise 64-bit Windows 7 SP1 Ultimate 64-bit Windows 7 SP1 Enterprise 64-bit Windows 7 SP1 Professional 64-bit Windows 7 SP1 Home Premium 64-bit Windows 7 SP1 Home Basic 64-bit Windows 7 SP1 Ultimate 32-bit Windows 7 SP1 Enterprise 32-bit Windows 7 SP1 Professional 32-bit Windows 7 SP1 Home Premium 32-bit Windows 7 SP1 Home Basic 32-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Web 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 32-bit Windows Server 2008 SP2 Standard 32-bit	Windows 8 Pro 64-bit Windows 8 Enterprise 64-bit Windows 7 SP1 Ultimate 64-bit Windows 7 SP1 Enterprise 64-bit Windows 7 SP1 Professional 64-bit Windows 7 SP1 Home Premium 64-bit Windows 7 SP1 Home Basic 64-bit Windows Server 2008 SP2 Datacenter 64-bit Windows Server 2008 SP2 Enterprise 64-bit Windows Server 2008 SP2 Standard 64-bit Windows Server 2008 SP2 Web 64-bit

SQL SERVER EDITION	32-BIT	64-BIT
SQL Server Express	Windows 10 Home 64-bit Windows 10 Professional 64-bit Windows 10 Enterprise 64-bit Windows 10 Home 32-bit Windows 10 Professional 32-bit Windows 10 Enterprise 32-bit Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2012 R2 Datacenter 64-bit Windows Server 2012 R2 Standard 64-bit Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Foundation 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows 8.1 32-bit Windows 8.1 Pro 32-bit	Windows 10 Home 64-bit Windows 10 Professional 64-bit Windows 10 Enterprise 64-bit Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2012 R2 Datacenter 64-bit Windows Server 2012 R2 Standard 64-bit Windows Server 2012 R2 Essentials 64-bit Windows Server 2012 R2 Foundation 64-bit Windows Server 2012 Datacenter 64-bit Windows Server 2012 Standard 64-bit Windows Server 2012 Essentials 64-bit Windows Server 2012 Foundation 64-bit Windows Server 2008 R2 SP1 Datacenter 64-bit Windows Server 2008 R2 SP1 Enterprise 64-bit Windows Server 2008 R2 SP1 Standard 64-bit Windows Server 2008 R2 SP1 Foundation 64-bit Windows Server 2008 R2 SP1 Web 64-bit Windows 8.1 64-bit Windows 8.1 Pro 64-bit Windows 8.1 Enterprise 64-bit Windows 8 64-bit Windows 8 Pro 64-bit

SQL SERVER EDITION	32-BIT	64-BIT
	Windows 8.1 Enterprise 32-bit	Windows 8 Enterprise 64-bit
	Windows 8.1 64-bit	Windows 7 SP1 Ultimate 64-bit
	Windows 8.1 Pro 64-bit	Windows 7 SP1 Enterprise 64-bit
	Windows 8.1 Enterprise 64-bit	Windows 7 SP1 Professional 64-bit
	Windows 8 32-bit	Windows 7 SP1 Home Premium 64-bit
	Windows 8 Pro 32-bit	Windows 7 SP1 Home Basic 64-bit
	Windows 8 Enterprise 32-bit	Windows Server 2008 SP2 Datacenter 64-bit
	Windows 8 64-bit	Windows Server 2008 SP2 Enterprise 64-bit
	Windows 8 Pro 64-bit	Windows Server 2008 SP2 Standard 64-bit
	Windows 8 Enterprise 64-bit	Windows Server 2008 SP2 Foundation 64-bit
	Windows 7 SP1 Ultimate 64-bit	Windows Server 2008 SP2 Web 64-bit
	Windows 7 SP1 Enterprise 64-bit	
	Windows 7 SP1 Professional 64-bit	
	Windows 7 SP1 Home Premium 64-bit	
	Windows 7 SP1 Home Basic 64-bit	
	Windows 7 SP1 Ultimate 32-bit	
	Windows 7 SP1 Enterprise 32-bit	
	Windows 7 SP1 Professional 32-bit	
	Windows 7 SP1 Home Premium 32-bit	
	Windows 7 SP1 Home Basic 32-bit	
	Windows Server 2008 SP2 Datacenter 64-bit	
	Windows Server 2008 SP2 Enterprise 64-bit	
	Windows Server 2008 SP2 Standard 64-bit	
	Windows Server 2008 SP2 Foundation 64-bit	
	Windows Server 2008 SP2 Web 64-bit	
	Windows Server 2008 SP2 Datacenter 64-bit	
	Windows Server 2008 SP2 Enterprise 32-bit	
	Windows Server 2008 SP2 Standard 32-bit	

SQL SERVER EDITION	32-BIT	64-BIT
	Windows Server 2008 SP2 Web 32-bit	

Minimum SQL Server version requirements for Windows 10

Before you install SQL Server on a computer that is running Windows 10, you must make sure that you fulfill the following minimum requirements:

- For SQL Server 2014, you must apply SQL Server 2014 Service Pack 1 or a later update. For more information, see How to obtain the latest service pack for SQL Server 2014.
- For SQL Server 2012, you must apply SQL Server 2012 Service Pack 2 or a later update. For more information, see How to obtain the latest service pack for SQL Server 2012.
- SQL Server 2008 R2
  and SQL Server 2008 is not supported on Windows 10.

Cross-Language Support

For more information about cross-language support and considerations for installing SQL Server in localized languages, see Local Language Versions in SQL Server.

Extended System Support

SQL Server 2014 64-bit versions include support for extended systems, also known as Windows 32-bit on Windows 64-bit (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 64-bit.

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

During installation 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 **6.0 GB** of available disk space on the system drive for these files. This requirement applies even if you install components to a non-default drive.

Actual hard disk space requirements depend on your system configuration and the features that you decide to install. For a list of features supported by the editions of SQL Server, see Features Supported by the Editions of SQL Server 2014. The following table provides disk space requirements for SQL Server 2014 components.

FEATURE	DISK SPACE REQUIREMENT
Database Engine and data files, Replication, Full-Text Search, and Data Quality Services	811 MB
Analysis Services and data files	345 MB
Reporting Services and Report Manager	304 MB
Integration Services	591 MB
Master Data Services	243 MB
Client Components (other than SQL Server Books Online components and Integration Services tools)	1823 MB

FEATURE	DISK SPACE REQUIREMENT
SQL Server Books Online Components to view and manage help content[1]	375 KB

[1]The disk space requirement for downloaded Books Online content is 200 MB.

Storage Types for Data Files

The supported storage types for data files are:

- Local Disk
- Shared Storage
- SMB File Share

> **NOTE:** SMB storage is not supported for Analysis Services data files for either standalone or clustered installations. Use direct attached storage or a storage area network instead.

> **IMPORTANT!!** SMB storage can be hosted by a Windows File Server or a third party SMB storage device. If Windows File Server is used, the Windows File Server version should be 2008 or later. For more information about installing SQL Server using SMB file share as a storage option, see Install SQL Server with SMB Fileshare as a Storage Option.

> **WARNING!!!!** SQL Server failover cluster installation supports Local Disk only for installing the tempdb files. Ensure that the path specified for the tempdb data and log files is valid on **all** of the cluster nodes. During failover, if the tempdb directories are not available on the failover target node, the SQL Server resource will fail to come online.

Installing SQL Server on a Domain Controller - Limitations

For security reasons, we recommend that you do not install SQL Server 2014 on a domain controller. SQL Server Setup will not block installation on a computer that is a domain controller, but the following limitations apply:

- You cannot run SQL Server services on a domain controller under a local 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.

See Also

Planning a SQL Server Installation
Security Considerations for a SQL Server Installation

Security Considerations for a SQL Server Installation

12/3/2018 • 5 minutes to read • Edit Online

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
- Use firewalls
- Isolate services
- Configure a secure file system
- Disable NetBIOS and server message block
- Installing SQL Server on a domain controller

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 Configure the Windows Firewall to Allow SQL Server Access.

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 Configure Windows Service Accounts and Permissions.

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.

  > **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.

Installing SQL Server on a domain controller

For security reasons, we recommend that you do not install SQL Server 2014 on a domain controller. SQL Server Setup will not block installation on a computer that is a domain controller, but the following limitations apply:

- You cannot run SQL Server services on a domain controller under a local 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.

During or After Installation of 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 Configure Windows Service Accounts and Permissions.

Authentication mode

- Require Windows Authentication for connections to SQL Server.
- Use Kerberos authentication. For more information, see Register a Service Principal Name for Kerberos Connections.

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.

> **IMPORTANT**
> During setup of SQL Server Express a login is added for the BUILTIN\Users group. This allows all authenticated users of the computer to access the instance of SQL Server Express as a member of the public role. The BUILTIN\Users login can be safely removed to restrict Database Engine access to computer users who have individual logins or are members of other Windows groups with logins.

See Also

Hardware and Software Requirements for Installing SQL Server 2014
Network Protocols and Network Libraries
Register a Service Principal Name for Kerberos Connections

Network Protocols and Network Libraries

12/13/2018 • 2 minutes to read • Edit Online

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). Select an unassigned port number. Port number assignments are managed by the Internet Assigned Numbers Authority and are listed at http://www.iana.org.

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.

Server Message Block Protocol

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.

> **WARNING**
> Disabling Server Message Block will block the SQL Server or Windows Cluster service from accessing the remote file share. Do not disable SMB if you do or plan to do one of the following:
>
> - Use Windows Cluster Node and File Share Majority Quorum mode
> - Specify an SMB file share as the data directory during SQL Server installation
> - Create a database file on an SMB file share

To disable SMB

1. On the **Start** menu, point to **Settings**, and then click **Network and Dial-up Connections**.

   Right-click the Internet-facing connection, and then click **Properties**.

2. Select the **Client for Microsoft Networks** check box, and then click **Uninstall**.

3. Follow the uninstall steps.

4. Select **File and Printer Sharing for Microsoft Networks**, and then click **Uninstall**.

5. Follow the uninstall steps.

To disable SMB on servers accessible from the Internet

- In the Local Area Connection properties, use the **Transmission Control Protocol/Internet Protocol (TCP/IP) properties** dialog box to remove **File and Printer Sharing for Microsoft Networks** and **Client**

for Microsoft Networks.

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

For more information about endpoints, see Configure the Database Engine to Listen on Multiple TCP Ports and Endpoints Catalog Views (Transact-SQL).

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

- Server Network Configuration

See Also

Surface Area Configuration
Security Considerations for a SQL Server Installation
Planning a SQL Server Installation

Work with Multiple Versions and Instances of SQL Server

12/3/2018 • 4 minutes to read • Edit Online

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 Supported Version and Edition Upgrades.

Version Components and Numbering

The following concepts are useful in understanding the behavior of SQL Server for side-by-side instances of SQL Server.

The standard product version format for SQL Server is MM.nn.bbbb.rr where each segment is defined as:

MM - Major version

nn - Minor version

bbbb - Build number

rr - Build revision number

In each major or minor release of SQL Server, there is an increment to the version number to differentiate it from earlier versions. This change to the version is used for many purposes. This includes displaying version information in the user interface, controlling how files are replaced during upgrade, applying service packs, and also as a mechanism for functional differentiation between the successive versions.

Components shared by all versions of SQL Server

Certain components are shared by all instances of all installed versions of SQL Server. When you install different versions of SQL Server side-by-side on the same machine, these components are automatically upgraded to the latest version. Such components are usually uninstalled automatically when the last instance of SQL Server is uninstalled.

Examples: SQL Server Browser and Microsoft SQL Server VSS Writer.

Components shared across all instances of the same major version of SQL Server

SQL Server versions that have the same major version share some components across all instances. If the shared components are selected during upgrade, the existing components are upgraded to the latest version.

Examples: Integration Services, Master Data Services, SQL Server Management Studio, SQL Server Data Tools (SSDT), and SQL Server Books Online.

Components shared across minor versions

SQL Server versions that have the same major.minor version shared components.

Example: Setup support files.

Components specific to an instance of SQL Server

Some SQL Server components or services are specific to an instance of SQL Server. These are also known as instance-aware. They share the same version as the instance that hosts them, and are used exclusively for that instance.

Examples: Database Engine, Analysis Services, and Reporting Services.

Components that are independent of the SQL Server versions

Certain components are installed during SQL Server setup, but are independent of the versions of SQL Server. They may be shared across major versions or by all SQL Server versions.

Examples: Microsoft Sync Framework, SQL Server Compact.

For more information about SQL Server Compact installation, see Install SQL Server 2014 from the Installation Wizard (Setup). For more information about how to uninstall SQL Server Compact, see Uninstall an Existing Instance of SQL Server (Setup).

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.

Caution

SQL Server SysPrep does not support side by side installation of prepared instances of SQL Server 2014 with earlier versions of SQL Server on the same computer. For example, you cannot prepare a SQL Server 2014 instance side by side with a prepared instance of SQL Server 2012. However, you can install multiple prepared instances of the same major version of SQL Server side by side on the same computer. For more information, see Considerations for Installing SQL Server Using SysPrep.

SQL Server 2014 cannot be installed side-by-side with earlier versions of SQL Server on a computer that is running Windows Server 2008 R2 Server Core SP1. For more information on Server Core installations, see Install SQL Server 2014 on Server Core.

The following table shows side-by-side support for SQL Server 2014:

EXISTING INSTANCE OF SQL SERVER 2014	SIDE-BY-SIDE SUPPORT
SQL Server 2014 (32-bit)	SQL Server 2005 (32-bit) SQL Server 2005 (64-bit) x64 SQL Server 2008 (32-bit) SQL Server 2008 (64-bit) x64 SQL Server 2008 R2 (32-bit) SQL Server 2008 R2 (64-bit) x64 SQL Server 2012 (32-bit) SQL Server 2012 (64-bit) x64 SQL Server 2014 (32-bit) SQL Server 2014 (64-bit) x64

EXISTING INSTANCE OF SQL SERVER 2014	SIDE-BY-SIDE SUPPORT
SQL Server 2014 (64-bit) x64	SQL Server 2005 (32-bit) SQL Server 2005 (64-bit) x64 SQL Server 2008 (32-bit) SQL Server 2008 (64-bit) x64 SQL Server 2008 R2 (32-bit) SQL Server 2008 R2 (64-bit) x64 SQL Server 2012 (32-bit) SQL Server 2012 (64-bit) x64 SQL Server 2014 (32-bit) SQL Server 2014 (64-bit) x64

Preventing IP Address Conflicts

When a SQL Server Failover Cluster Instance is installed side-by-side with a standalone instance of the SQL Server Database Engine, take care to avoid TCP port number conflicts on the IP addresses. Conflicts usually occur when two instances of the Database Engine are both configured to use the default TCP port (1433). To avoid conflicts, configure one instance to use a non-default fixed port. Configuring a fixed port is usually easiest on the standalone instance. Configuring the Database Engine to use different ports will prevent an unexpected IP Address/TCP port conflict that blocks an instance startup when a SQL Server Failover Cluster Instance fails to the standby node

See Also

Hardware and Software Requirements for Installing SQL Server 2014
Install SQL Server 2014 from the Installation Wizard (Setup)
Supported Version and Edition Upgrades
Upgrade to SQL Server 2014
Features Supported by the Editions of SQL Server 2014
Backward Compatibility
Use Upgrade Advisor to Prepare for Upgrades

Local Language Versions in SQL Server

12/3/2018 • 2 minutes to read • Edit Online

SQL Server supports all languages that are supported by Windows operating systems.

Cross-Language Support

- The English-language version of SQL Server is supported on all localized versions of operating systems.
- Localized versions of SQL Server are supported on localized operating systems with the corresponding language or on English-language versions of supported operating systems by using the Windows Multilingual User Interface Pack (MUI) settings. For more information, see Configure Operating System to Support Localized Versions.
- Localized versions of SQL Server can only be upgraded to localized versions of the same language, and cannot be upgraded to the English-language version.
- Localized versions of SQL Server can also be installed side by side with English-language instances of SQL Server.

Configure Operating System to Support Localized Versions

Localized versions of SQL Server are supported on English-language versions of supported operating systems through the use of Windows Multilingual User Interface Pack (MUI) settings.

However, you must verify certain operating system settings before installing a localized version of SQL Server 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.

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.

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.

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

Hardware and Software Requirements for Installing SQL Server 2014
Install SQL Server 2014

File Locations for Default and Named Instances of SQL Server

12/10/2018 • 7 minutes to read • Edit Online

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.

> **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.
>
> System databases (master, model, MSDB, and tempdb), and Database Engine user databases can be installed with Server Message Block (SMB) file server as a storage option. This applies to both SQL Server stand-alone and SQL Server failover cluster installations (FCI). For more information, see Install SQL Server with SMB Fileshare as a Storage Option.
>
> 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.

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\120\, where <*drive*> is the drive letter where components are installed. The default is usually drive C.

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, followed by an underscore and the minor version when applicable, and a period, followed by the instance name.
- MSAS for Analysis Services, followed by the major version number, followed by an underscore and the minor version when applicable, and a period, followed by the instance name.
- MSRS for Reporting Services, followed by the major version number, followed by an underscore and the minor version when applicable, and a period, followed by the instance name.

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

- MSSQL12.MSSQLSERVER for a default instance of SQL Server 2014.
- MSAS12.MSSQLSERVER for a default instance of SQL Server 2014 Analysis Services (SSAS).
- MSSQL12.MyInstance for a named instance of SQL Server 2014 named "MyInstance."

The directory structure for a SQL Server 2014 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\MSSQL12.MyInstance\
- C:\Program Files\Microsoft SQL Server\MSAS12.MyInstance\

You can specify any value for the instance ID, but avoid special characters and reserved keywords.

You can specify a non-default instance ID during SQL Server Setup. Instead of <Program Files>\MicrosoftSQL Server, a <custom path>\MicrosoftSQL 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.

> **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\120\. 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\MicrosoftSQL Server\<*Instance_ID*> for instance-aware components. For example,

- HKLM\Software\Microsoft\MicrosoftSQL Server\MSSQL12.MyInstance
- HKLM\Software\Microsoft\MicrosoftSQL Server\MSAS12.MyInstance
- HKLM\Software\Microsoft\MicrosoftSQL Server\MSRS12.MyInstance

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\MicrosoftSQL Server\Instance Names\SQL]
  "InstanceName"="MSSQL12"
- [HKEY_LOCAL_MACHINE\Software\Microsoft\MicrosoftSQL Server\Instance Names\OLAP]
  "InstanceName"="MSAS12"
- [HKEY_LOCAL_MACHINE\Software\Microsoft\MicrosoftSQL Server\Instance Names\RS]
  "InstanceName"="MSRS12"

Specifying File Paths

During Setup, you can change the installation path for the following features:

The installation path is displayed in Setup only for features with a user-configurable destination folder:

COMPONENT	DEFAULT PATH[1, 2]	CONFIGURABLE[3] OR FIXED PATH
Database Engine server components	\Program Files\MicrosoftSQL Server\MSSQL12.<InstanceID>|Configurable	
Database Engine data files	\Program Files\MicrosoftSQL Server\MSSQL12.<InstanceID>|Configurable	
Analysis Services server	\Program Files\MicrosoftSQL Server\MSAS12.<InstanceID>|Configurable	
Analysis Services data files	\Program Files\MicrosoftSQL Server\MSAS12.<InstanceID>|Configurable	
Reporting Services report server	\Program Files\MicrosoftSQL Server\MSRS12.<InstanceID>\Reporting Services\ReportServer\Bin|Configurable	
Reporting Services report manager	\Program Files\MicrosoftSQL Server\MSRS12.<InstanceID>\Reporting Services\ReportManager|Fixed path	
Integration Services	<Install Directory>\120\DTS|Configurable[4]	
Client Components (except bcp.exe and sqlcmd.exe)	<Install Directory>\120\Tools|Configurable[4]	
Client Components (bcp.exe and sqlcmd.exe)	<Install Directory>\Client SDK\ODBC\110\Tools\Binn	Fixed path
Replication and server-side COM objects	<*drive*>:\Program Files\Microsoft SQL Server\120\COM\[5]	Fixed path
Integration Services component DLLs for the Data Transformation Run-time engine, the Data Transformation Pipeline engine, and the `dtexec` command prompt utility	<*drive*>:\Program Files\Microsoft SQL Server\120\DTS\Binn	Fixed path
DLLs that provide managed connection support for Integration Services	<*drive*>:\Program Files\Microsoft SQL Server\120\DTS\Connections	Fixed path
DLLs for each type of enumerator that Integration Services supports	<*drive*>:\Program Files\Microsoft SQL Server\120\DTS\ForEachEnumerators	Fixed path
SQL Server Browser Service, WMI providers	<*drive*>:\Program Files\Microsoft SQL Server\120\Shared|Fixed path	
Components that are shared between all instances of SQL Server	<*drive*>:\Program Files\Microsoft SQL Server\120\Shared|Fixed path	

[1]Ensure that the \Program Files\MicrosoftSQL Server\ folder is protected with limited permissions.

[2]The default drive for these locations is *systemdrive*, normally drive C.

[3]Installation paths for child features are determined by the installation path of the parent feature.

[4]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.

[5]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.

> **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\MicrosoftSQL Server\MSSQL12.<InstanceName>\MSSQL\", instance-specific 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*>\MSSQL12\MSSQL\MSSQL10_50\MSSQL\Data\. Instead, to use the USESYSDB feature, set an installation path for the SQL Data Files feature instead of the Database Engine feature.

> **NOTE**
> Data files are always expected to be found in a child directory named Data. For example, specify C:\Program Files\MicrosoftSQL Server\MSSQL12.<InstanceName>\ to specify the root path to the data directory of the system databases during upgrade when data files are found under C:\Program Files\MicrosoftSQL Server\MSSQL12.<InstanceName>\MSSQL\Data.

See Also

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

Install SQL Server 2014

8/8/2019 • 2 minutes to read • Edit Online

Download SQL Server 2014 Express

Thank you to Scott Hanselman for collecting all of the installer package links in one place!

This topic provides an overview of different installation options we have for installing SQL Server 2014. For more information about the various SQL Server components that can be installed, and the installation process, see Installation for SQL Server 2014.

> **NOTE:** SQL Server is available in 32-bit and 64-bit editions. The 64-bit and 32-bit editions of SQL Server are installed either through the Installation Wizard, or at a command prompt. For more information about SQL Server components, see Editions and Components of SQL Server 2014 and Features Supported by the Editions of SQL Server 2014.

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, see the CodePlex Web site. To read about support for SQL Server sample databases and sample code for SQL Server Express, see Databases and Samples Overview.

Before you install SQL Server, review installation requirements, system configuration checks, and security considerations. For more information, see Planning a SQL Server Installation. Review the topics in the next section for information about various installation scenarios for SQL Server 2014.

Install SQL Server 2014 Components

TOPIC	DESCRIPTION
About the SQL Server Database Engine	Describes how to install and configure the SQL Server Database Engine.
Install SQL Server Replication	Describes how to install and configure SQL Server Replication.
Install Distributed Replay	Lists down the topics to install the Distributed Replay feature.
Install SQL Server Management Tools	Describes how to install and configure SQL Server management tools.
Install SQL Server PowerShell	Describes the considerations for installing SQL Server PowerShell components.

How to Install SQL Server 2014

TITLE	DESCRIPTION
Installation How-to Topics	Provides links to procedural topics for installing SQL Server 2014 from the installation wizard, from the command prompt, by using configuration files, and by using SysPrep.

TITLE	DESCRIPTION
Install SQL Server 2014 on Server Core	Review this topic to install SQL Server 2014 on Windows Server Core.
Validate a SQL Server Installation	Review the use of the SQL Discovery report to verify the version of SQL Server and the SQL Server features installed on the computer.
Check Parameters for the System Configuration Checker	Discusses the function of the System Configuration Checker (SCC).

Configuration

TOPIC	DESCRIPTION
Configure the Windows Firewall to Allow SQL Server Access	This topic provides an overview of firewall configuration and how to configure the Windows firewall.
Configure a Multi-Homed Computer for SQL Server Access	This topic describes how to configure SQL Server and Windows Firewall with Advanced Security to provide for network connections to an instance of SQL Server in a multi-homed environment.
Configure the Windows Firewall to Allow Analysis Services Access	You can follow the steps provided in this topic to configure both port and firewall settings to allow access to Analysis Services or PowerPivot for SharePoint.

Related Sections

Install SQL Server 2014 BI Features
SQL Server features that are part of the Microsoft BI platform include Analysis Services, Integration Services, Reporting Services, Data Quality Services, and several client applications used for creating or working with analytical data. This section of the SQL Server Setup documentation explains how to install Analysis Services, Integration Services, Master Data Services, Data Quality Services, and Reporting Services.

SQL Server Failover Cluster Installation
This section of the SQL Server Setup documentation explains how to install, and configure SQL Server failover cluster.

See Also

Planning a SQL Server Installation
Upgrade to SQL Server 2014
Uninstall SQL Server 2014
High Availability Solutions (SQL Server)

About the SQL Server Database Engine

12/13/2018 • 2 minutes to read • Edit Online

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 Install SQL Server 2014 from the Installation Wizard (Setup).

Important For local installations, you must run Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.

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
- Data Quality Services - is an optional component

> **NOTE**
> In this release, selecting the **Data Quality Services** check box in setup does not install the Data Quality Services (DQS) server. You will have to perform additional steps post installation to install DQS server. For more information, see Install Data Quality Services.

The following additional features are options for many typical user scenarios:

- Data Quality Client
- Integration Services
- Connectivity components
- Programming models
- Management tools
- Management Studio
- Documentation components

> **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 CodePlex Web site.

See Also

Features Supported by the Editions of SQL Server 2014

Editions and Components of SQL Server 2014

Planning a SQL Server Installation

High Availability Solutions (SQL Server)

Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)

Install Distributed Replay from the Command Prompt

12/3/2018 • 2 minutes to read • Edit Online

Installing a new instance of Distributed Replay at the command prompt enables you to specify the features to install and how they should be configured. The command prompt installation supports installing, repairing, upgrading, and uninstalling of the Distributed Replay components. When installing through the command prompt, SQL Server supports full quiet mode by using the /Q parameter.

> **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.

Installation Parameters

The list of top-level features include SQL Server, Analysis Services, Reporting Services, Integration Services, and Tools. The Tools feature will install SQL Server Management Tools, SQL Server Books Online, SQL Server Data Tools (SSDT), and other shared components. To install the Distributed Replay components, specify the following parameters:

COMPONENT	PARAMETER
Distributed Replay controller	**DREPLAY_CTLR**
Distributed Replay client	**DREPLAY_CLT**
Administration Tool	**Tools**

> **IMPORTANT**
> After you install Distributed Replay you must create firewall rules on the controller and client computers, and grant each client computer permissions on the target server. For more information, see Complete the Post-Installation Steps.

Use the parameters in the following table to develop command line scripts for installation.

PARAMETER	DESCRIPTION	SUPPORTED VALUES
/CTLRSVCACCOUNT **Optional**	Service account for the Distributed Replay controller service.	Checks account and password
/CTLRSVCPASSWORD **Optional**	Password for the Distributed Replay controller service account.	Checks account and password

PARAMETER	DESCRIPTION	SUPPORTED VALUES
/CTLRSTARTUPTYPE **Optional**	Startup type for the Distributed Replay controller service.	Automatic Disabled Manual
/CTLRUSERS **Optional**	Specify which users have permissions for the Distributed Replay controller service.	Set of user account strings using " " (space) for delimiter **Important**: When you configure the Distributed Replay controller service, you can specify one or more user accounts that will be used to run the Distributed Replay client services. The following is the list of supported accounts: Domain user account User created local user account Administrator Virtual account and MSA (Managed Service Account) Network Services, Local Services, and System Group accounts (local or domain) and other built-in accounts (like Everyone) are not accepted.
/CLTSVCACCOUNT **Optional**	Service account for the Distributed Replay client service.	Checks account and password
/CLTSVCPASSWORD **Optional**	Password for the Distributed Replay client service account.	Checks account and password
/CLTSTARTUPTYPE **Optional**	Startup type for the Distributed Replay client service.	Automatic Disabled Manual
/CLTCTLRNAME **Optional**	The computer name that the client communicates with for the Distributed Replay Controller service.	
/CLTWORKINGDIR **Optional**	The working directory for the Distributed Replay Client service.	Valid path

PARAMETER	DESCRIPTION	SUPPORTED VALUES
/CLTRESULTDIR Optional	The result directory for the Distributed Replay Client service.	Valid path

Sample Syntax:

To install the Distributed Replay controller component

```
setup /q /ACTION=Install /FEATURES=DREPLAY_CTLR /IAcceptSQLServerLicenseTerms /CTLRUSERS="domain\user1"
"domain\user2" /CTLRSVCACCOUNT="domain\svcuser" /CTLRSVCPASSWORD="password" /CTLRSTARTUPTYPE=Automatic
```

To install the Distributed Replay client component

```
setup /q /ACTION=Install /FEATURES=DREPLAY_CLT /IAcceptSQLServerLicenseTerms /CLTSVCACCOUNT="domain\svcuser"
/CLTSVCPASSWORD="password" /CLTSTARTUPTYPE=Automatic /CLTCTLRNAME=ControllerMachineName
/CLTWORKINGDIR="C:\WorkingDir" /CLTRESULTDIR="C:\ResultDir
```

Distributed Replay Requirements

12/3/2018 • 4 minutes to read • Edit Online

Before using the MicrosoftSQL Server Distributed Replay feature, consider the product requirements that are outlined in this topic.

Input Trace Requirements

To successfully replay trace data, it must meet the requirements for version and format, and contain the required events and columns.

Input Trace Versions

Distributed Replay supports input trace data that is collected on the following versions of SQL Server:

- SQL Server 2014
- SQL Server 2012
- SQL Server 2008 R2
- SQL Server 2008
- SQL Server 2005

Input Trace Formats

The input trace data can be in any of the following formats:

- A single trace file that has the `.trc` extension.
- A set of rollover trace files that follow the file rollover naming convention, for example: `<TraceFile>.trc`, `<TraceFile>_1.trc`, `<TraceFile>_2.trc`, `<TraceFile>_3.trc`, ... `<TraceFile>_n.trc`.

Input Trace Events and Columns

The input trace data must contain specific events and columns to be replayed by Distributed Replay. The **TSQL_Replay** template in SQL Server Profiler contains all of the required events and columns, in addition to extra information. For more information about that template, see Replay Requirements.

> **WARNING**
> If you do not use the **TSQL_Replay** template to capture the input trace data, or if the input trace requirements are not satisfied, you may receive unexpected replay results.

You can also create a custom trace template and use it to replay events with Distributed Replay, as long as it contains the following events:

- Audit Login
- Audit Logout
- ExistingConnection
- RPC Output Parameter
- RPC:Completed

- RPC:Starting
- SQL:BatchCompleted
- SQL:BatchStarting

If you are replaying server-side cursors, the following events are also required:

- CursorClose
- CursorExecute
- CursorOpen
- CursorPrepare
- CursorUnprepare

If you are replaying server-side prepared SQL statements, the following events are also required:

- Exec Prepared SQL
- Prepare SQL

All input trace data must contain the following columns:

- Event Class
- EventSequence
- TextData
- Application Name
- LoginName
- DatabaseName
- Database ID
- HostName
- Binary Data
- SPID
- Start Time
- EndTime
- IsSystem

Supported Input Trace and Target Server Combinations

The following table lists the supported versions of trace data, and for each, the supported versions of SQL Server that data can be replayed against.

VERSION OF INPUT TRACE DATA	SUPPORTED VERSIONS OF SQL SERVER FOR THE TARGET SERVER INSTANCE
SQL Server 2005	SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, SQL Server 2014

VERSION OF INPUT TRACE DATA	SUPPORTED VERSIONS OF SQL SERVER FOR THE TARGET SERVER INSTANCE
SQL Server 2008	SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, SQL Server 2014
SQL Server 2008 R2	SQL Server 2008 R2, SQL Server 2012, SQL Server 2014
SQL Server 2012	SQL Server 2012, SQL Server 2014
SQL Server 2014	SQL Server 2014

Operating System Requirements

Supported operating systems for running the administration tool and the controller and client services is the same as your SQL Server instance. For more information about which operating systems are supported for your SQL Server instance, see Hardware and Software Requirements for Installing SQL Server 2014.

Distributed Replay features are supported on both x86-based and x64-based operating systems. For x64-based operating systems, only Windows on Windows (WOW) mode is supported.

Installation Limitations

Any one computer can only have a single instance of each Distributed Replay feature installed. The following table lists how many installations of each feature are allowed in a single Distributed Replay environment.

DISTRIBUTED REPLAY FEATURE	MAXIMUM INSTALLATIONS PER REPLAY ENVIRONMENT
SQL Server Distributed Replay controller service	1
SQL Server Distributed Replay client service	16 (physical or virtual computers)
Administration tool	Unlimited

NOTE

Although only one instance of the administration tool can be installed on a single computer, you can start multiple instances of the administration tool. Commands issued from multiple administration tools are resolved in the order in which they are received.

Data Access Provider

Distributed Replay only supports the SQL Server Native Client ODBC data access provider.

Target Server Preparation Requirements

We recommend that the target server be located in a test environment. To replay trace data against a different instance of SQL Server than it was originally recorded, make sure that the following has been done to the target server:

- All logins and users that are contained in the trace data must be present in the same database on the target server.
- All logins and users on the target server must have the same permissions they had on the original server.

- The database IDs on the target ideally should be the same as those on the source. However, if they are not the same, matching can be performed based on **DatabaseName** if it is present in the trace.
- The default database for each login that is contained in the trace data must be set (on the target server) to the respective target database of the login. For example, the trace data to be replayed contains activity for the login, **Fred**, in the database **Fred_Db** on the original instance of SQL Server. Therefore, on the target server, the default database for the login, **Fred**, must be set to the database that matches **Fred_Db** (even if the database name is different). To set the default database of the login, use the `sp_defaultdb` system stored procedure.

Replaying events associated with missing or incorrect logins results in replay errors, but the replay operation continues.

See Also

SQL Server Distributed Replay
Distributed Replay Security
Install Distributed Replay

Install Distributed Replay (Setup)

5/22/2019 • 4 minutes to read • Edit Online

Install the MicrosoftSQL Server Distributed Replay features with the SQL Server 2014 Installation Wizard. When planning where to install the features, consider the following:

- You can install the administration tool on the same computer as the Distributed Replay controller, or on different computers.
- There can only be one controller in each Distributed Replay environment.
- You can install the client service on up to 16 (physical or virtual) computers.
- Only one instance of the client service can be installed on the Distributed Replay controller computer. If your Distributed Replay environment will have more than one client, we do not recommend installing the client service on the same computer as the controller. Doing so may decrease the overall speed of the distributed replay.
- For performance testing scenarios, we do not recommend installing the administration tool, Distributed Replay controller service, or client service on the target instance of SQL Server. Installing all of these features on the target server should be limited to functional testing for application compatibility.
- After installation, the controller service, SQL Server Distributed Replay controller, must be running before you start the Distributed Replay client service on the clients.

> **NOTE**
> To remove or change the Distributed Replay features, use the Windows **Programs and Features** window in **Control Panel**. Select SQL Server 2014 in the **Uninstall or change a program** window, and then click **Remove** to open the SQL Server 2014 Installation Wizard. On the **Select Features** page, check the Distributed Replay features you want to remove.

Prerequisites:

- Make sure that the computers that you want to use meet the requirements that are described in the topic Distributed Replay Requirements.
- Before you begin this procedure, you create the domain user accounts that the controller and client services will run under. We recommend that these accounts are not members of the Windows Administrators group. For more information, see the User and Service Accounts section in the Distributed Replay Security topic.

  > **NOTE**
  > You can use local user accounts if you are running the administration tool, controller service, and client service on the same computer.

Installation Locations:

Assuming you use the default file locations and a standard installation, the base directory is found at C:\Program Files\Microsoft SQL Server. Within that, following are where the binaries and assemblies are installed to:

- On a 32-bit system:

  C:\Program Files\Microsoft SQL Server\120\Tools

- OR -

<Share Feature Directory>\Tools\(user-supplied alternative shared feature directory)

- On a 64-bit system:

  C:\Program Files\MicrosoftSQL Server (x86) \120\Tools

  - OR -

  <Share Feature Directory (x86)>\Tools\(user-supplied alternative shared feature (x86) directory)

To install Distributed Replay features

1. To start the installation of any of the Distributed Replay features, start the SQL Server 2014 Installation Wizard.
2. The **Setup Support Rules** page identifies issues that might occur when installing the SQL Server Setup support files. You must correct any Setup support failures before continuing with Setup.
3. On the **Product Key** page, select an option 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 Editions and Components of SQL Server 2014.
4. On the **License Terms** page, read the license agreement, and then select the check box to accept the license terms and conditions. To help improve SQL Server, you can also enable the feature usage option and send reports to Microsoft.
5. On the **Setup Support Files** page, click **Install** to install or update the Setup Support files for SQL Server 2014.
6. On the **Setup Role** page, select **SQL Server Feature Installation**, and then click **Next** to continue to the **Feature Selection** page.
7. On the **Feature Selection** page, configure which features you want to install.
   - To install the administration tool, select **Management Tools - Basic**.
   - To install the controller service, select **Distributed Replay Controller**.
   - To install the client service, select **Distributed Replay Client**.

   Important: When you configure Distributed Replay controller, you can specify one or more user accounts that will be used to run the Distributed Replay client services. The following is the list of supported accounts:

   - Domain user account
   - User created local user account
   - Administrator
   - Virtual account and MSA (Managed Service Account)
   - Network Services, Local Services, and System

   Group accounts (local or domain) and other built-in accounts (like Everyone) are not accepted.
8. Optionally, click the ellipsis (...) button to change the shared feature directory path.
   a. On 32-bit computers, the default installation path is **C:\Program Files\MicrosoftSQL Server**
   b. On 64-bit computers, the default installation path is **C:\Program Files (x86)\MicrosoftSQL Server**

9. When you are finished, click **Next**.
10. On the **Installation Rules** page, SQL Server Setup validates your computer configuration. Once the validation process is completed, click **Next**.
11. The **Disk Space Requirements** page calculates the required disk space for the features that you specify. Then it compares the required space to the available disk space.
12. On the **Error Reporting** page, specify the information that you want to send to Microsoft to help improve SQL Server. By default, option for error reporting is enabled.
13. On the **Installation Configuration Rules** page, the System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features that you have specified.
14. On the **Ready to Install the Program** page, click **Install**.

> **IMPORTANT**
> After you install Distributed Replay you must create firewall rules on the controller and client computers, and grant each client computer permissions on the target server. For more information, see Complete the Post-Installation Steps.

These additional topics document other ways to install Distributed Replay:

- Install Distributed Replay from the Command Prompt
- Install Distributed Replay Using a Configuration File

.NET Framework Security

You must have administrative permissions in order to install any of the Distributed Replay features. Only a SQL Server login having sysadmin permissions can add the client service accounts to the sysadmin server role of the test server. For more information about Distributed Replay security considerations, see Distributed Replay Security.

See Also

Features Supported by the Editions of SQL Server 2014
SQL Server Distributed Replay
Distributed Replay Requirements
Administration Tool Command-line Options (Distributed Replay Utility)
Configure Distributed Replay

Install Distributed Replay

12/3/2018 • 2 minutes to read • Edit Online

Use the following topics to install the Distributed Replay feature.

In This Section

TOPIC	DESCRIPTION
Install Distributed Replay (Setup)	Procedural topic for a typical Distributed Replay installation by using the Setup Wizard.
Install Distributed Replay from the Command Prompt	Procedural topic that provides sample syntax and installation parameters for running unattended Setup.
Install Distributed Replay Using a Configuration File	Procedural topic that provides sample syntax and installation parameters for running Distributed Reply through a configuration file.
Complete the Post-Installation Steps	Procedural topic for completing a Distributed Replay installation.
Modify the Controller and Client Services Accounts	Procedural topic for how to start and stop the Distributed Replay controller and client services, and modify the service accounts.
Repair a Distributed Replay Installation	Procedural topic for repairing a failed Distributed Replay installation.
Uninstall an Existing Instance of Distributed Replay	Procedural topic for uninstalling an existing instance of Distributed Replay.

Install Distributed Replay Using a Configuration File

5/22/2019 • 2 minutes to read • Edit Online

SQL Server Setup provides the ability to generate a configuration file based on user input and system defaults. If you specify that you want the Management tools installed, you can use the configuration file to deploy the three Distributed Replay components (administration tool, Distributed Replay controller, and the Distributed Replay client). It supports Installing, repairing, and uninstalling of the Distributed Replay components.

Setup supports the use of the configuration file only through the command-line. The processing order of the parameters while using the configuration file is outlined below:

- The configuration file overwrites the defaults in a package
- Command-line values overwrite the values in the configuration file

For more information about how to use a configuration file, see Install SQL Server 2014 Using a Configuration File.

> **IMPORTANT**
> After you install Distributed Replay you must create firewall rules on the controller and client computers, and grant each client computer permissions on the target server. For more information, see Complete the Post-Installation Steps.

To generate a configuration file

1. Follow the Setup wizard through to the **Ready to Install** page. The path to the configuration file is specified in the **Ready to Install** page in the configuration file path section.
2. Cancel the setup without actually completing the installation, to generate the INI file.

To Install Distributed Replay Using the Configuration File

- Run the installation through the command prompt and supply the ConfigurationFile.ini using the ConfigurationFile parameter.

Sample Syntax

Following is an example on how to specify the configuration file at the command prompt:

```
Setup.exe /CTLRSVCPASSWORD="ctlrsvcpswd" /CLTSVCPASSWORD="cltsvcpswd" /
ConfigurationFile=ConfigurationFile.INI\
```

> **NOTE**
> You must specify both passwords in the command line because you cannot configure them in the configuration file.

Complete the Post-Installation Steps

12/3/2018 • 2 minutes to read • Edit Online

After you install Distributed Replay you must modify the Distributed Replay controller and client services accounts.

To complete the post-installation steps

1. **Create firewall rules**: On the controller and client computers, you must allow inbound traffic through the firewall for the corresponding service. Specify the firewall rules for the service executables, located in the installation folders.

   a. For the controller service, create a rule for **DReplayController.exe**, located in the installation folder. For example, the following command enables this rule, where `%InstallPath%` is the installation folder of the service:

   ```
   netsh advfirewall firewall add rule name="allow dreplay controller" dir=in
   program="%InstallPath%\DReplayController\DReplayController.exe" action=allow
   ```

   b. For the client service, on each client computer, create a rule for **DReplayClient.exe**, located in the installation folder. For example, the following command enables this rule, where `%InstallPath%` is the installation folder of the service:

   ```
   netsh advfirewall firewall add rule name="allow dreplay client" dir=in
   program="%InstallPath%\DReplayClient\DReplayClient.exe" action=allow
   ```

2. **Grant each client permissions on the target server**: After you have completed installing the client service on the client computers, you must manually add the client service accounts to the sysadmin role on the target instance of SQL Server.

.NET Framework Security

You must have administrative permissions in order to install any of the Distributed Replay features. Only a SQL Server login having sysadmin permissions can add the client service accounts to the sysadmin server role of the test server. For more information about Distributed Replay security considerations, see Distributed Replay Security.

Modify the Controller and Client Services Accounts

12/3/2018 • 2 minutes to read • Edit Online

In this topic, you will learn how to modify the Distributed Replay controller and client service accounts, and then reapply the access control lists (ACLs).

To start or stop the Distributed Replay services using Computer Management

1. On the computer on which the Distributed Replay services are installed, from the command prompt, type `dcomcnfg`.
2. Double-click **Services**, scroll down and right-click **SQL Server Distributed Replay <service name>**, and then click **Start** or **Stop**.

To modify the Distributed Replay controller service

1. On the controller computer, stop the SQL Server Distributed Replay controller service.
2. Under **Services**, right-click **SQL Server Distributed Replay Controller**, and then select **Properties**.
3. In the **SQL Server Distributed Replay Controller Properties** window, on the **Log On** tab, select **This account**, type or click **Browse** to enter the new logon account, and then click **OK**.

   Important: When you configure Distributed Replay Controller, you can specify one or more user accounts that will be used to run the Distributed Replay Client services. The following is the list of supported accounts:

   - Domain user account
   - User created local user account
   - Administrator
   - Virtual account and MSA (Managed Service Account)
   - Network Services, Local Services, and System

   Group accounts (local or domain) and other built-in accounts (like Everyone) are not accepted.
4. Start the Distributed Replay controller service.

To modify the Distributed Replay client service

1. Before you modify the Distributed Replay client service, make sure the client service account you are changing was specified during setup (in the CTLRUSERS parameter on the controller computer). If the client service account you are changing was not specified during setup, you must perform the following steps first:

   a. Stop the SQL Server Distributed Replay controller service.

   b. On the controller computer on which the controller service is installed, from the command prompt, type `dcomcnfg`.

   c. In the **Component Services** window, navigate to **Console Root -> Component Services -> Computers -> My Computer -> DCOM Config ->DReplayController**.

   d. Right-click **DReplayController**, and then click **Properties**.

   e. In the **DReplayController Properties** window, on the **Security** tab, click **Edit** in the **Launch and Activation Permissions** section.

   f. Grant the new client service logon account **Local and Remote activation** permissions, and then

click **OK**.

g. Click **Edit** in the **Access Permissions** section, and grant the new client service logon account **Local and Remote access** permissions, and then click **OK**.

h. Click **OK** to close the **DReplayController Properties** window.

i. On the controller computer, add the changed client service logon account to the **Distributed COM Users** group.

j. Start the SQL Server Distributed Replay controller service.

2. Stop the SQL Server Distributed Replay client service.

3. In the **SQL Server Distributed Replay Client Properties** window, on the **Log On** tab, select **This account**, type or click **Browse** to enter the new logon account, and then click **OK**.

4. Start the Distributed Replay client service.

Repair a Distributed Replay Installation

5/22/2019 • 2 minutes to read • Edit Online

Repairing a Distributed Replay component (controller or client) will do the following:

1. Install all associated files on disk again, and replace any existing files.
2. Recreate the Windows service account if the corresponding service account was removed.

You cannot use the Repair operation to add or remove components. To add or remove components, check or uncheck the corresponding component in the Feature tree on the **Feature Selection** page in SQL Server Setup.

To repair a failed installation of Distributed Replay

1. From the **Start** menu, click **Control Panel**, and then double-click **Add or Remove Programs**.
2. Select SQL Server 2014 in the **Uninstall or change a program** window, and then in the SQL Server 2014 dialog box, click **Repair**.
3. Follow the steps in the SQL Server 2014 wizard, and on the **Select Features** page, select the Distributed Replay components you want to repair, and then click **Next.**.
4. Complete the SQL Server 2014 Installation Wizard to repair the selected Distributed Replay features.

Uninstall an Existing Instance of Distributed Replay

5/22/2019 • 2 minutes to read • Edit Online

Uninstalling a Distributed Replay component (controller or client) will remove the following:

- All associated files
- Registry keys
- Corresponding services

To uninstall an instance of Distributed Replay

1. From the **Start** menu, click **Control Panel**, and then double-click **Add or Remove Programs**.

   Select SQL Server 2014 in the **Uninstall or change a program** window, and then in the SQL Server 2014 dialog box, click **Remove**.

2. Follow the steps in the SQL Server 2014 wizard, and on the **Select Features** page, select the Distributed Replay components you want to remove, and then click **Next.**.

3. Complete the SQL Server 2014 Installation Wizard to uninstall the selected Distributed Replay features.

Install SQL Server Replication

11/5/2018 • 2 minutes to read • Edit Online

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 Configure Distribution in SQL Server Books Online.

> **IMPORTANT**
> 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 Install SQL Server 2014 from the Command Prompt.

See Also

Install SQL Server 2014
Install SQL Server 2014 from the Command Prompt
Features Supported by the Editions of SQL Server 2014

Install SQL Server Management Tools

5/22/2019 • 2 minutes to read • Edit Online

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.
- SQL Server Data Tools (SSDT) add-ins to Microsoft Visual Studio

Note that SQL Server Data Tools (SSDT) 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 2014 Management Tools will be installed.

SQL Server 2014 Management Tools can run side-by-side on the same computer with earlier versions of SQL Server Management Tools.

See Also

Features Supported by the Editions of SQL Server 2014
Editions and Components of SQL Server 2014
Install SQL Server 2014 from the Installation Wizard (Setup)
Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)

Install SQL Server PowerShell

12/13/2018 • 2 minutes to read • Edit Online

SQL Server Setup will stop if it detects that you have selected SQL Server features that include PowerShell components, but Windows PowerShell 2.0 is not installed. You must install PowerShell by using the Windows Management Framework, and then rerun Setup.

Installing SQL Server PowerShell Support

You install the software that provides SQL Server support for Windows PowerShell by using SQL Server Setup. When you select any SQL Server features that require PowerShell support, Setup checks that Windows PowerShell 2.0 is installed. If PowerShell 2.0 is present, Setup then installs the following SQL Server PowerShell components:

- The SQL Server PowerShell snap-ins. The snap-ins are dll files that implement two types of Windows PowerShell support for SQL Server:
  - A set of SQL Server cmdlets. Cmdlets are commands that implement a specific action. For example, **Invoke-Sqlcmd** runs a Transact-SQL or XQuery script that can also be run by using the **sqlcmd** utility, and **Invoke-PolicyEvaluation** reports whether SQL Server objects comply with policy-based management policies.
  - A SQL Server provider. The provider lets you navigate the hierarchy of SQL Server objects using a path similar to a file system path. Each object is associated with a class from the SQL Server Management object models. You can use the methods and properties of the class to perform work on the objects. For example, if you cd to a databases object in a path, you can use the methods and properties of the Microsoft.SqlServer.Managment.SMO.Database class to manage the database.
- The **sqlps** module that is imported into Windows PowerShell 2.0 sessions to load the SQL Server snap-ins.
- The deprecated **sqlps** utility that starts a Windows PowerShell 2.0 session and imports the **sqlps** module.
- SQL Server Management Studio supports starting Windows PowerShell sessions from the Object Explorer tree. SQL Server Agent supports Windows PowerShell job steps.

If Windows PowerShell 2.0 has not been installed, or has been uninstalled, you must install it by following the instructions on the Windows Management Framework page.

If Windows PowerShell is uninstalled after Setup finishes, the SQL Server features for Windows PowerShell will not function. Windows PowerShell can be uninstalled by Windows users, and uninstalling Windows PowerShell might be required by some Windows operating system upgrades. To use the SQL Server PowerShell features, you must reinstall PowerShell 2.0 by using the Windows Management Framework.

See Also

SQL Server PowerShell

Install SQL Server 2014 on Server Core

11/5/2019 • 12 minutes to read • Edit Online

You can install SQL Server on a Server Core installation of Windows Server 2008 R2 SP1 or Windows Server 2012. This topic provides setup-specific details for installing SQL Server 2014 on Server Core.

The Server Core installation option for the Windows Server 2008 R2 or Windows Server 2012 operating system provides a minimal environment for running specific server roles. This helps to reduce maintenance and management requirements and the attack surface for those server roles. For more information on Server Core as implemented on Windows Server 2008 R2, see Server Core for Windows Server 2008 R2 (https://go.microsoft.com/fwlink/?LinkId=202439). For more information on Server Core as implemented on Windows Server 2012, see Server Core for Windows Server 2012 (https://msdn.microsoft.com/library/hh846323(VS.85).aspx).

Prerequisites

REQUIREMENT	HOW TO INSTALL
.NET Framework 2.0 SP2	Included in Server Core installation of Windows Server 2008 R2 SP1 and Windows Server 2012. If it is not enabled, Setup enables it by default. It is not possible to run versions 2.0, 3.0, and 3.5 side by side on a computer. When you install the .NET Framework 3.5 SP1, you get the 2.0 and 3.0 layers automatically.
.NET Framework 3.5 SP1 Full Profile	Included in Server Core installation of Windows Server 2008 R2 SP1. If it is not enabled, Setup enables it by default. On a computer with Windows server operating system you must download and install .NET Framework 3.5 SP1 before you run Setup, to install components dependent on .NET 3.5 SP1. For more information about the recommendations and guidance on how to acquire and enable .NET Framework 3.5 in Windows Server 2012, see Microsoft .NET Framework 3.5 Deployment Considerations (https://msdn.microsoft.com/library/windows/hardware/hh975396).
.NET Framework 4 Server Core Profile	For all editions of SQL Server 2014 except SQL Server Express, Setup installs the .NET Framework 4 Server Core Profile as a prerequisite. For SQL Server 2012 Express, download the .NET Framework 4 Server Core Profile from Microsoft .NET Framework 4 (Standalone Installer) for Server Core (https://www.microsoft.com/download/details.aspx?id=17718), and install it before you proceed with the setup.
Windows Installer 4.5	Shipped with Server Core installation of Windows Server 2008 R2 SP1 and Windows Server 2012.

REQUIREMENT	HOW TO INSTALL
Windows PowerShell 2.0	Shipped with Server Core installation of Windows Server 2008 R2 SP1 and Windows Server 2012.

Supported Features

Use the following table to find which features are supported in SQL Server 2014 on a Server Core installation of Windows Server 2008 R2 SP1 and Windows Server 2012.

FEATURE	SUPPORTED
Database Engine Services	Yes
SQL Server Replication	Yes
Full Text Search	Yes
Analysis Services	Yes
Reporting Services	No
SQL Server Data Tools (SSDT)	No
Client Tools Connectivity	Yes
Integration Services Server[1]	Yes
Client Tools Backward Compatibility	No
Client Tools SDK	No
SQL Server Books Online	No
Management Tools - Basic	Remote Only[2]
Management Tools - Complete	Remote Only[2]
Distributed Replay Controller	No
Distributed Replay Client	Remote Only[2]
SQL Client Connectivity SDK	Yes
Microsoft Sync Framework	Yes[3]
Master Data Services	No
Data Quality Services	No

[1]For more information about the new Integration Services Server and its features in SQL Server 2014, see Integration Services (SSIS) Server.

[2]Installation of these features on Server Core is not supported. These components can be installed on a different server that is not Windows Server 2008 R2 Server Core SP1 or Windows Server 2012 Server Core, and connected to the Database Engine services installed on Server Core.

[3]Microsoft Sync Framework is not included in the SQL Server 2014 installation package. You can download the appropriate version of Sync Framework from this Microsoft Download Center (https://go.microsoft.com/fwlink/?LinkId=221788) page and install it on a computer that is running Server Core installation of Windows Server 2008 R2 SP1 or Windows Server 2012.

Supported Scenario Matrix

The following table shows the supported scenario matrix for installing SQL Server 2014 on a Server Core installation of Windows Server 2008 R2 SP1 and Windows Server 2012.

SQL Server editions	All SQL Server 2014 64-bit editions[1]
SQL Server language	All languages
SQL Server language on OS language/locale (combination)	ENG SQL Server on JPN (Japanese) Windows ENG SQL Server on GER (German) Windows ENG SQL Server on CHS (Chinese-China) Windows ENG SQL Server on ARA (Arabic (SA)) Windows ENG SQL Server on THA (Thai) Windows ENG SQL Server on TRK (Turkish) Windows ENG SQL Server on pt-PT (Portuguese Portugal) Windows ENG SQL Server on ENG (English) Windows
Windows edition	Windows Server 2012 64-bit x64 Datacenter Windows Server 2012 64-bit x64 Standard Windows Server 2008 R2 SP1 64-bit x64 Data Center Server Core Windows Server 2008 R2 SP1 64-bit x64 Enterprise Server Core Windows Server 2008 R2 SP1 64-bit x64 Standard Server Core Windows Server 2008 R2 SP1 64-bit x64 Web Server Core

[1]Installing the 32-bit version of SQL Server 2014 editions is not supported on Server Core.

Upgrading

On Server Core installations, upgrading from SQL Server 2012 to SQL Server 2014 is supported.

Installation

SQL Server 2014 does not support setup by using the installation wizard on the Server Core operating system. When installing on Server Core, SQL Server Setup supports full quiet mode by using the /Q parameter, or Quiet Simple mode by using the /QS parameter. For more information, see Install SQL Server 2014 from the Command Prompt.

> **IMPORTANT**
> SQL Server 2014 cannot be installed side-by-side with earlier versions of SQL Server on a computer that is running Windows Server 2008 R2 Server Core SP1 or Windows Server 2012 Server Core.

Regardless of the installation method, you are required to confirm acceptance of the software license terms as an individual or on behalf of an entity, unless your use of the software is governed by a separate agreement such as a Microsoft volume licensing agreement or a third-party agreement with an ISV or OEM.

The license terms are displayed for review and acceptance in the Setup user interface. Unattended installations (using the /Q or /QS parameters) must include the /IACCEPTSQLSERVERLICENSETERMS parameter. You can review the license terms separately at Microsoft Software License Terms.

> **NOTE**
> Depending on how you received the software (for example, through Microsoft volume licensing), your use of the software may be subject to additional terms and conditions.

To install specific features, use the /FEATURES parameter and specify the parent feature or feature values. For more information about feature parameters and their use, see the following sections.

Feature Parameters

FEATURE PARAMETER	DESCRIPTION
SQLENGINE	Installs only the Database Engine.
REPLICATION	Installs the Replication component along with Database Engine.
FULLTEXT	Installs the FullText component along with Database Engine.
AS	Installs all Analysis Services components.
IS	Installs all Integration Services components.
CONN	Installs the connectivity components.

See the following examples of the usage of feature parameters:

PARAMETER AND VALUES	DESCRIPTION
/FEATURES=SQLEngine	Installs only the Database Engine.
/FEATURES=SQLEngine,FullText	Installs the Database Engine and full-text.
/FEATURES=SQLEngine,Conn	Installs the Database Engine and the connectivity components.

PARAMETER AND VALUES	DESCRIPTION
/FEATURES=SQLEngine,AS,IS,Conn	Installs the Database Engine, Analysis Services, Integration Services, and the connectivity components.

Installation Options

The Setup supports the following installation options while installing SQL Server 2014 on a Server Core operating system:

1. Installation from Command Line

   To install specific features using the command prompt installation option, use the /FEATURES parameter and specify the parent feature or feature values. The following is an example of using the parameters from the command line:

   ```
   setup.exe /qs /ACTION=Install /FEATURES=SQLEngine,Replication /INSTANCENAME=MSSQLSERVER
   /SQLSVCACCOUNT="<DomainName\UserName>" /SQLSVCPASSWORD="<StrongPassword>" /SQLSYSADMINACCOUNTS="
   <DomainName\UserName>" /AGTSVCACCOUNT="NT AUTHORITY\Network Service" /TCPENABLED=1
   /IACCEPTSQLSERVERLICENSETERMS
   ```

2. Installation using Configuration File

   Setup supports the use of the configuration file only through the command prompt. The configuration file is a text file with the basic structure of a parameter (name/value pair) and a descriptive comment. The configuration file specified at the command prompt should have an .INI file name extension. See the following examples of ConfigurationFile.INI:

   - Installing Database Engine

     The following example shows how to install a new stand-alone instance that includes the SQL ServerDatabase Engine:

```
; ssNoVersion Configuration File
[OPTIONS]

; Specifies a Setup work flow, like INSTALL, UNINSTALL, or UPGRADE. This is a required
parameter.

ACTION="Install"

; Specifies features to install, uninstall, or upgrade. The lists of features include SQLEngine,
FullText, Replication, AS, IS, and Conn.

FEATURES=SQLENGINE

; Specify a default or named instance. MSSQLSERVER is the default instance for non-Express
editions and SQLExpress for Express editions. This parameter is required when installing the
ssNoVersion Database Engine, and Analysis Services (AS).

INSTANCENAME="MSSQLSERVER"

; Specify the Instance ID for the ssNoVersion features you have specified. ssNoVersion directory
structure, registry structure, and service names will incorporate the instance ID of the
ssNoVersion instance.

INSTANCEID="MSSQLSERVER"

; Account for ssNoVersion service: Domain\User or system account.

SQLSVCACCOUNT="NT Service\MSSQLSERVER"

; Windows account(s) to provision as ssNoVersion system administrators.

SQLSYSADMINACCOUNTS="<DomainName\UserName>"

; Accept the License agreement to continue with Installation

IAcceptSQLServerLicenseTerms="True"
```

- Installing connectivity components

  The following example shows how to install the connectivity components:

```
; ssNoVersion Configuration File
[OPTIONS]

; Specifies a Setup work flow, like INSTALL, UNINSTALL, or UPGRADE. This is a required
parameter.

ACTION="Install"

; Specifies features to install, uninstall, or upgrade. The lists of features include SQLEngine,
FullText, Replication, AS, IS, and Conn.

FEATURES=Conn

; Specifies acceptance of License Terms

IAcceptSQLServerLicenseTerms="True
```

- Installing all supported features

  The following example shows how to install all supported features of SQL Server 2014 on Server Core:

```
; ssNoVersion Configuration File
[OPTIONS]
; Specifies a Setup work flow, like INSTALL, UNINSTALL, or UPGRADE. This is a required
parameter.

ACTION="Install"

; Specifies features to install, uninstall, or upgrade. The lists of features include SQLEngine,
FullText, Replication, AS, IS, and Conn.

FEATURES=SQLENGINE,FullText,Replication,AS,IS,Conn

; Specify a default or named instance. MSSQLSERVER is the default instance for non-Express
editions and SQLExpress for Express editions. This parameter is required when installing the
ssNoVersion Database Engine (SQL), or Analysis Services (AS).

INSTANCENAME="MSSQLSERVER"

; Specify the Instance ID for the ssNoVersion features you have specified. ssNoVersion directory
structure, registry structure, and service names will incorporate the instance ID of the
ssNoVersion instance.

INSTANCEID="MSSQLSERVER"

; Account for ssNoVersion service: Domain\User or system account.

SQLSVCACCOUNT="NT Service\MSSQLSERVER"

; Windows account(s) to provision as ssNoVersion system administrators.

SQLSYSADMINACCOUNTS="<DomainName\UserName>"

; The name of the account that the Analysis Services service runs under.

ASSVCACCOUNT= "NT Service\MSSQLServerOLAPService"

; Specifies the list of administrator accounts that need to be provisioned.

ASSYSADMINACCOUNTS="<DomainName\UserName>"

; Specifies the server mode of the Analysis Services instance. Valid values are
MULTIDIMENSIONAL, POWERPIVOT or TABULAR. ASSERVERMODE is case-sensitive. All values must be
expressed in upper case.

ASSERVERMODE="MULTIDIMENSIONAL"

; Optional value, which specifies the state of the TCP protocol for the ssNoVersion service.
Supported values are: 0 to disable the TCP protocol, and 1 to enable the TCP protocol.

TCPENABLED=1

;Specifies acceptance of License Terms

IAcceptSQLServerLicenseTerms="True"
```

The following examples show how you can launch the Setup using a configuration file.

- Configuration file

  Following are some examples of how to use the configuration file:

  - To specify the configuration file at the command prompt:

```
setup.exe /QS /ConfigurationFile=MyConfigurationFile.INI
```

- To specify passwords at the command prompt instead of in the configuration file:

```
setup.exe /QS /SQLSVCPASSWORD="************" /ASSVCPASSWORD="************"
/ConfigurationFile=MyConfigurationFile.INI
```

- DefaultSetup.ini

  If you have the DefaultSetup.ini file in the \x86 and \x64 folders at the root level of the SQL Server source media, open the DefaultSetup.ini file, and then add the *Features* parameter to the file.

  If the DefaultSetup.ini file does not exist, you can create it and copy it to the \x86 and \x64 folders at the root level of the SQL Server source media.

Configuring Remote Access of SQL Server Running on Server Core

Perform the actions described below to configure remote access of a SQL Server 2014 instance that is running on a Server Core installation of Windows Server 2008 R2 SP1 or Windows Server 2012.

Enable remote connections on the instance of SQL Server

To enable remote connections, use SQLCMD.exe locally and execute the following statements against the Server Core instance:

- `EXEC sys.sp_configure N'remote access', N'1'`

  `GO`

- `RECONFIGURE WITH OVERRIDE`

  `GO`

Enable and start the SQL Server Browser service

By default, the Browser service is disabled. If it is disabled on an instance of SQL Server running on Server Core, run the following command from the command prompt to enable it:

```
sc config SQLBROWSER start= auto
```

After it is enabled, run the following command from the command prompt to start the service:

```
net start SQLBROWSER
```

Create exceptions in Windows Firewall

To create exceptions for SQL Server access in Windows Firewall, follow the steps specified in Configure the Windows Firewall to Allow SQL Server Access.

Enable TCP/IP on the instance of SQL Server

The TCP/IP protocol can be enabled through Windows PowerShell for an instance of SQL Server on Server Core. Follow these steps:

1. On the computer that is running Windows Server 2008 R2 Server Core SP1 or Windows Server 2012 Server Core, launch Task Manager.
2. On the **Applications** tab, click **New Task**.
3. In the **Create New Task** dialog box, type **sqlps.exe** in the **Open** field and then click **OK**. This opens the **MicrosoftSQL Server Powershell** window.
4. In the **Microsoft SQL Server Powershell** window, run the following script to enable the TCP/IP protocol:

```
$smo = 'Microsoft.SqlServer.Management.Smo.'
$wmi = New-Object ($smo + 'Wmi.ManagedComputer')
# Enable the TCP protocol on the default instance.  If the instance is named, replace MSSQLSERVER with the
instance name in the following line.
$uri = "ManagedComputer[@Name='" + (get-item env:\computername).Value +
"']/ServerInstance[@Name='MSSQLSERVER']/ServerProtocol[@Name='Tcp']"
$Tcp = $wmi.GetSmoObject($uri)
$Tcp.IsEnabled = $true
$Tcp.Alter()
$Tcp
```

Uninstallation

After you log on to a computer that is running Windows Server 2008 R2 Server Core SP1 or Windows Server 2012 Server Core, you have a limited desktop environment with an Administrator command prompt. You can use this command prompt to initiate uninstallation of an instance of SQL Server 2014. To uninstall an instance of SQL Server 2014, launch the uninstallation from the command prompt in full quiet mode by using the /Q parameter, or quiet simple mode by using the /QS parameter. The /QS parameter shows progress through the UI, but does not accept any input. /Q runs in a quiet mode without any user interface.

To uninstall an existing instance of SQL Server:

```
setup.exe /Q /Action=Uninstall /FEATURES=SQLEngine,AS,IS /INSTANCENAME=MSSQLSERVER
```

To remove a named instance, specify the name of the instance instead of "MSSQLSERVER" in the preceding example.

WARNING

If you accidentally close the command prompt, you can start a new command prompt by following these steps:

1. Press Ctrl+Shift+Esc to display Task Manager.
2. On the **Applications** tab, click **New Task**.
3. In the **Create New Task** dialog box, type **cmd** in the **Open** field and then Click **OK**..

See Also

Install SQL Server 2014 Using a Configuration File
Install SQL Server 2014 from the Command Prompt
Features Supported by the Editions of SQL Server 2014
Server Core Installation Option Getting Started Guide
Configuring a Server Core installation: Overview
Failover Cluster Cmdlets in Windows PowerShell Listed by Task Focus
Mapping Cluster.exe Commands to Windows PowerShell Cmdlets for Failover Clusters

Install SQL Server with SMB Fileshare as a Storage Option

1/8/2019 • 6 minutes to read • Edit Online

Starting SQL Server 2012, system databases (Master, Model, MSDB, and TempDB), and Database Engine user databases can be installed with Server Message Block (SMB) file server as a storage option. This applies to both SQL Server stand-alone and SQL Server failover cluster installations (FCI).

> **NOTE**
> Filestream is currently not supported on an SMB file share.

Installation Considerations

SMB File share Formats:

While specifying the SMB file share, the following are supported Universal Naming Convention (UNC) Path formats for standalone and FCI databases:

- \\ServerName\ShareName\
- \\ServerName\ShareName

For more information about Universal Naming Convention, see UNC (https://go.microsoft.com/fwlink/?LinkId=245534).

The loopback UNC path (a UNC path whose server name is localhost, 127.0.0.1, or the local machine name) is not supported. As a special case, SQL Server using File Server Cluster which is hosted on the same node SQL Server is running is also not supported. To prevent this situation, it is recommended that SQL Server and File Server Cluster to be created on separated Windows Clusters.

The below UNC path formats are not supported:

- Loopback path, e.g., \\localhost\..\ or \\127.0.0.1\...\
- Administrative shares, e.g., \\servername\x$
- Other UNC path formats like \\?\x:\
- Mapped network drives.

Supported Data Definition Language (DDL) statements

The following Transact-SQL DDL statements and database engine stored procedures support SMB file shares:

1. CREATE DATABASE (SQL Server Transact-SQL)
2. ALTER DATABASE (Transact-SQL)
3. RESTORE (Transact-SQL)
4. BACKUP (Transact-SQL)
5. sp_attach_db (Transact-SQL)
6. sp_attach_single_file_db (Transact-SQL)

Installation options

- In setup UI "Database Engine Configuration" page, "Data Directories" tab, set the parameter "Data root directory as "\\fileserver1\share1".
- In command prompt installation, specify the "/INSTALLSQLDATADIR" as "\\fileserver1\share1".

  Here is the sample syntax to install SQL Server on a Standalone server using SMB file share option:

```
Setup.exe /q /ACTION=Install /FEATURES=SQL /INSTANCENAME=MSSQLSERVER /SQLSVCACCOUNT="
<DomainName\UserName>" /SQLSVCPASSWORD="<StrongPassword>" /SQLSYSADMINACCOUNTS="<DomainName\UserName>"
/AGTSVCACCOUNT="<DomainName\UserName>" /AGTSVCPASSWORD="<StrongPassword>"
/INSTALLSQLDATADIR="\\FileServer\Share1\" /IACCEPTSQLSERVERLICENSETERMS
```

  To install a single-node SQL Server failover cluster instance with the Database Engine and Analysis Services, default instance:

```
setup.exe /q /ACTION=InstallFailoverCluster /InstanceName=MSSQLSERVER /INDICATEPROGRESS
/ASSYSADMINACCOUNTS="<DomainName\UserName>" /ASDATADIR=<Drive>:\OLAP\Data /ASLOGDIR=<Drive>:\OLAP\Log
/ASBACKUPDIR=<Drive>:\OLAP\Backup /ASCONFIGDIR=<Drive>:\OLAP\Config /ASTEMPDIR=<Drive>:\OLAP\Temp
/FAILOVERCLUSTERDISKS="<Cluster Disk Resource Name - for example, 'Disk S:'"
/FAILOVERCLUSTERNETWORKNAME="<Insert Network Name>"
/FAILOVERCLUSTERIPADDRESSES="IPv4;xx.xxx.xx.xx;Cluster Network;xxx.xxx.xxx.x"
/FAILOVERCLUSTERGROUP="MSSQLSERVER" /Features=AS,SQL /ASSVCACCOUNT="<DomainName\UserName>"
/ASSVCPASSWORD="xxxxxxxxxxx" /AGTSVCACCOUNT="<DomainName\UserName>" /AGTSVCPASSWORD="xxxxxxxxxxx"
/INSTALLSQLDATADIR="\\FileServer\Share1\" /SQLCOLLATION="SQL_Latin1_General_CP1_CS_AS" /SQLSVCACCOUNT="
<DomainName\UserName>" /SQLSVCPASSWORD="xxxxxxxxxxx" /SQLSYSADMINACCOUNTS="<DomainName\UserName>
/IACCEPTSQLSERVERLICENSETERMS
```

  For more information about the usage of various command line parameter options in SQL Server 2014, see Install SQL Server 2014 from the Command Prompt.

Operating System Considerations (SMB Protocol vs. SQL Server)

Different Windows operating systems has different SMB protocol versions, and the SMB protocol version is transparent to SQL Server. You can find the benefits of different SMB protocol versions with respect to SQL Server 2014.

OPERATING SYSTEM	SMB2 PROTOCOL VERSION	BENEFITS TO SQL SERVER
Windows Server 2008 SP 2	2.0	Improved performance over previous SMB versions. Durability, which helps recover from temporary network glitches.

OPERATING SYSTEM	SMB2 PROTOCOL VERSION	BENEFITS TO SQL SERVER
Windows Server 2008 R2 SP 1, including Server Core	2.1	Support for large MTU, which benefits large data transfers, such as SQL backup and restore. This capability must be enabled by user. For more details on how to enable this capability, see What's New in SMB (https://go.microsoft.com/fwlink/?LinkID=237319). Significant performance improvements, specifically for SQL OLTP style workloads. These performance improvements require applying a hotfix. For more information on the hotfix, see this (https://go.microsoft.com/fwlink/?LinkId=237320).
Windows Server 2012, including Server Core	3.0	Support for transparent failover of file shares providing zero downtime with no administrator intervention required for SQL DBA or file server administrator in file server cluster configurations. Support for IO using multiple network interfaces simultaneously, as well as tolerance to network interface failure. Support for network interfaces with RDMA capabilities. For more information on these features and Server Message Block, see Server Message Block overview (https://go.microsoft.com/fwlink/?LinkId=253174). Support for Scale Out File Server (SoFS) with continuous availability.

OPERATING SYSTEM	SMB2 PROTOCOL VERSION	BENEFITS TO SQL SERVER
Windows Server 2012 R2, including Server Core	3.2	Support for transparent failover of file shares providing zero downtime with no administrator intervention required for SQL DBA or file server administrator in file server cluster configurations. Support for IO using multiple network interfaces simultaneously, as well as tolerance to network interface failure, using SMB Multichannel. Support for network interfaces with RDMA capabilities using SMB Direct. For more information on these features and Server Message Block, see Server Message Block overview (https://go.microsoft.com/fwlink/?LinkId=253174). Support for Scale Out File Server (SoFS) with continuous availability. Optimized for small random read/write I/O common to SQL Server OLTP. Maximum Transmission Unit (MTU) is turned on by default, which significantly enhances performance in large sequential transfers like SQL Server data warehouse and database backup or restore.

Security Considerations

- The SQL Server service account and SQL Server agent service account should have FULL CONTROL share permissions and NTFS permissions on the SMB share folders. The SQL Server service account can be a domain account or a system account if an SMB file server is used. For more information about share and NTFS permissions, see Share and NTFS Permissions on a File Server (https://go.microsoft.com/fwlink/?LinkId=245535).

  > **NOTE**
  > The FULL CONTROL share permissions and NTFS permissions on the SMB share folders should be restricted to: SQL Server service account, SQL Server Agent service account and windows users with admin server roles.

  It is recommended to use domain account as a SQL Server service account. If system account is used as a service account, grant the permissions for the machine account in the format: *<domain_name>**<computer_name>*$.

> **NOTE**
>
> - During SQL Server setup, it is required to specify domain account as a service account if SMB file share is specified as a storage option. With SMB file share, System account can only be specified as a service account post SQL Server installation.
> - Virtual accounts cannot be authenticated to a remote location. All virtual accounts use the permission of machine account. Provision the machine account in the format *<domain_name>**<computer_name>*$.

- The account used to install SQL Server should have FULL CONTROL permissions on the SMB file share folder used as the data directory, or any other data folders (User database directory, user database log directory, TempDB directory, TempDB log directory, backup directory) during Cluster Setup.
- The account used to install SQL Server should be granted SeSecurityPrivilege privileges on the SMB file server. To grant this privilege, use the Local Security Policy console on the file server to add the SQL Server setup account to the Manage auditing and security log policy. This setting is available in the User Rights Assignments section under Local Policies in the Local Security Policy console.

Known Issues

- After you detach a SQL Server 2014 database that resides on network-attached storage, you might run into database permission issue while trying to reattach the SQL Server database. The issue is defined in this KB article (https://go.microsoft.com/fwlink/?LinkId=237321). To work around this issue, see the **More Information** section in the KB article.
- Some third parties, like NetApp device do not support all SQL Server API calls. With these you might get:
  2015-06-04 13:14:19.97 spid9s Error: 17053, Severity: 16, State: 1.
  2015-06-04 13:14:19.97 spid9s DoDevIoCtlOut() GetOverlappedResult() : Operating system error 1(Incorrect function.) encountered.

  For NTFS, the error is harmless. but for ReFS, it can lead to significant performance degradation.
- If SMB file share is used as a storage option for a clustered instance of SQL Server, by default the SQL Server Failover Cluster Diagnostics Log cannot be written to the file share because SQL Server Resource DLL lacks the read/write permission on the file share. To resolve this issue, try one of the following methods:

  1. Grant read/write permissions on the file share to all computer objects in the cluster.
  2. Set the location of the diagnostic logs to a local file path. See the following example:

     ```
     ALTER SERVER CONFIGURATION
     SET DIAGNOSTICS LOG PATH = 'C:\logs';
     ```

See Also

Planning a SQL Server Installation
Installation How-to Topics
Configure Windows Service Accounts and Permissions

Considerations for Installing SQL Server Using SysPrep

12/13/2018 • 3 minutes to read • Edit Online

SQL Server SysPrep allows you to prepare a stand-alone instance of SQL Server on a computer and to complete the configuration at a later time. SQL Server SysPrep involves a two-step process to get to a configured stand-alone instance of SQL Server. The steps include the following:

- Prepare Image

  This step stops the installation process after the product binaries are installed, without configuring the computer, network, or account-specific information for the instance of SQL Server that is being prepared.

- Complete Image

  This step enables you to complete the configuration of a prepared instance of SQL Server. During this step, you can provide the computer, network, and account-specific information.

For more information about how to install SQL Server using SysPrep, see Install SQL Server 2014 Using SysPrep.

Common Uses for SQL Server SysPrep

You can use the SQL Server SysPrep capability in any of the following ways:

- By using the Prepare Image step, you can prepare one or more unconfigured instances of SQL Server on the same computer. You can configure these prepared instances by using the Complete Image step on the same computer.
- You can capture the SQL Server Setup configuration file of the prepared instance and use it to prepare additional unconfigured SQL Server instances on multiple computers for later configuration.
- In combination with the Windows System Preparation tool (also known as Windows SysPrep); you can create an image of the operating system including the unconfigured prepared instances of SQL Server on the source computer. You can then deploy the operating system image to multiple computers. After you complete the configuration of the operating system, you can configure the prepared instances by using the Complete Image step of SQL Server Setup.

  The Windows SysPrep tool is used to prepare Windows operating system images. It is used to capture a customized image of the operating system for deployment throughout an organization. For more information about SysPrep and its uses, see What is SysPrep.

Installation Media Considerations

If you are using a full version of SQL Server, consider the following:

- Non-Express editions of SQL Server:
  - The Prepare Image step uses SQL Server Evaluation edition to install the product binaries. When the instance is completed, the edition of SQL Server depends on the product ID provided during the complete image step.
  - If you provide an Evaluation edition product ID, the evaluation period is set to expire 180 days after the prepared instance is completed.

- Express editions of SQL Server:
  - To prepare an instance of SQL Server Express editions, use the Express installation media.
  - You cannot specify Product IDs for a prepared instance of SQL Server Express editions.

Supported SQL Server Installations

SysPrep in SQL Server 2014 supports all features, including tools, of SQL Server.

You can prepare multiple instances for side-by-side installations of SQL Server 2014 or earlier versions. The features of these instances must support SysPrep.

The SQL Server Native Client is automatically installed and completed at the end of the prepare image step.

SQL Server Browser and SQL Server Writer are automatically prepared when you prepare an instance of SQL Server. They are completed when you complete the SQL Server instance by using the Complete Image step.

For information about supported editions of SQL Server, see Features Supported by the Editions of SQL Server 2014.

You can perform an edition upgrade while configuring a prepared instance of SQL Server. This option is not supported for SQL Server Express editions.

Beginning in SQL Server 2014, SQL Server SysPrep supports SQL Server failover cluster installations from the command line.

SQL Server SysPrep Limitations

Repairing a prepared instance is not supported. If Setup fails during the Prepare Image or Complete Image step, you must run uninstall.

Prepare Image

The Prepare Image step installs the SQL Server product and features but does not configure the installation.

The SQL Server features to be installed and the installation location for SQL Server product installation files can be specified during this step. You can prepare an instance of SQL Server either through the **Image Preparation of a stand-alone instance for SysPrep deployment** on the **Advanced** page of the **Installation Center** or from the command prompt.

- You can prepare multiple instances of SQL Server on the same computer that can be completed later.
- You can add or remove features that are supported for SysPrep installations from the existing prepared instances of SQL Server.

After the instance is prepared, a shortcut on the **Start** menu becomes available to complete the configuration of the prepared instance of SQL Server.

Complete Image

You can complete the prepared instances of SQL Server by using either of the following methods:

- Use the shortcut on the Start menu.
- Access the **Image completion of a prepared stand-alone instance** step on the **Advanced** page of the **Installation Center**.

See Also

Planning a SQL Server Installation

Validate a SQL Server Installation

11/27/2018 • 2 minutes to read • Edit Online

The SQL Server discovery report can be used to verify the version of SQL Server and the SQL Server features installed on the computer. The **Installed SQL Server features discovery report** displays a report of all SQL Server 2000, SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, and SQL Server 2014 products and features that are installed on the local server. The SQL Server features discovery report is available on the **Tools** page on the SQL Server Installation center.

To run SQL Server features discovery report:

Launch the SQL Server Installation center, using the **Start** menu, point to **All Programs**, point to **MicrosoftSQL Server <Version Name>**, point to **Configuration Tools**, and click **SQL Server Installation Center**. To run the SQL Server features discovery report, click **Tools** in the left-hand navigation area of **SQL Server Installation Center**, and then click **Installed SQL Server features discovery report**.

The SQL Server discovery report is saved to %ProgramFiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<last Setup Session>.

You can also generate the discovery report through the command line. Run "Setup.exe /Action=RunDiscovery" from a command prompt If you add "/q" to the command line above no UI will be shown, but the report will still be created in %ProgramFiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<last Setup Session>.

See Also

View and Read SQL Server Setup Log Files

Check Parameters for the System Configuration Checker

11/5/2018 • 2 minutes to read • Edit Online

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.

The system configuration checker generates a report which contains a short description for each executed rule, and the execution status. The system configuration check report is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\.

See Also

Hardware and Software Requirements for Installing SQL Server 2014
Security Considerations for a SQL Server Installation
Supported Version and Edition Upgrades

Configure SQL Server on a Server Core Installation

10/23/2019 • 9 minutes to read • Edit Online

This topic covers details about configuring SQL Server on a Server Core installation of Windows Server 2008 R2 SP1.

Configure and Manage Server Core on Windows Server

The section provides references to the topics that help configure and manage a Server Core installation.

Not all features of SQL Server 2014 are supported in Server Core mode. Some of these features can be installed on a client computer or a different server that is not running Server Core, and connected to the Database Engine services installed on Server Core.

For more information about configuring and managing a Server Core installation remotely, see the following topics:

- Windows Server 2008 R2: Best Practices for Server Core Deployments (https://go.microsoft.com/fwlink/?LinkID=245957)
- Configuring a Server Core installation: Overview (https://go.microsoft.com/fwlink/?LinkId=245958)
- Configuring a Server Core installation of Windows Server 2008 R2 with Sconfig.cmd (https://go.microsoft.com/fwlink/?LinkId=245959)
- Installing a server role on a server running a Server Core installation of Windows Server 2008 R2: Overview (https://go.microsoft.com/fwlink/?LinkId=245960)
- Installing Windows Features on a server running a Server Core installation of Windows Server 2008 R2: Overview (https://go.microsoft.com/fwlink/?LinkId=245961)
- Managing a Server Core installation: Overview (https://go.microsoft.com/fwlink/?LinkId=245962)
- Administering a Server Core installation (https://go.microsoft.com/fwlink/?LinkId=245963)

Install updates

This section provides information about installing updates for SQL Server 2014 on a Windows Server Core machine. We recommend that customers evaluate and install latest SQL Server updates in a timely manner to make sure that systems are up-to-date with the most recent security updates. For more information about installing SQL Server 2014 on a Windows Server Core machine, see Install SQL Server 2014 on Server Core.

The following are the two scenarios for installing product updates:

- Installing Updates for SQL Server 2014 During a New Installation
- Installing Updates for SQL Server 2014 After It Has Been Installed

Installing updates during a new installation

SQL Server Setup supports only command prompt installations on Server Core operating system. For more information, see Install SQL Server 2014 from the Command Prompt.

SQL Server setup integrates the latest product updates with the main product installation so that the main product and its applicable updates are installed at the same time.

After Setup finds the latest versions of the applicable updates, it downloads and integrates them with the current SQL Server setup process. Product Update can pull in a cumulative update, service pack, or service pack plus cumulative update.

Specify the UpdateEnabled, and UpdateSource parameters to include the latest product updates with the main product installation. Refer the following example to enable product updates during the SQL Server Setup:

```
Setup.exe /qs /ACTION=Install /FEATURES=SQLEngine,Replication /INSTANCENAME=MSSQLSERVER /SQLSVCACCOUNT="
<DomainName\UserName>" /SQLSVCPASSWORD="<StrongPassword>" /SQLSYSADMINACCOUNTS="<DomainName\UserName>"
/AGTSVCACCOUNT="NT AUTHORITY\Network Service" /UpdateEnabled=True /UpdateSource="<SourcePath>"
/IACCEPTSQLSERVERLICENSETERMS
```

Installing updates after installation

On an installed instance of SQL Server 2014, we recommend that you apply the latest security updates and critical updates including General Distribution Releases (GDRs), and Service Packs (SPs). Individual Cumulative updates and security updates should be adopted on a case-by-case, "as-needed" basis. Evaluate the update; if it's needed, then apply it.

Apply an update at a command prompt, replacing <package_name> with the name of your update package:

- Update a single instance of SQL Server and all shared components. You can specify the instance either by using the InstanceName parameter or the InstanceID parameter.

  ```
  <package_name>.exe /qs /IAcceptSQLServerLicenseTerms /Action=Patch /InstanceName=MyInstance
  ```

- Update SQL Server shared components only:

  ```
  <package_name>.exe /qs /IAcceptSQLServerLicenseTerms /Action=Patch
  ```

- Update all instances of SQL Server on the computer and all shared components:

  ```
  <package_name>.exe /qs /IAcceptSQLServerLicenseTerms /Action=Patch /AllInstances
  ```

Start/Stop SQL Server service

The sqlservr Application application starts, stops, pauses, and continues an instance of SQL Server from a command prompt.

You can also use Net services to start and stop the SQL Server services.

Enable AlwaysOn Availability Groups

Being enabled for AlwaysOn Availability Groups is a prerequisite for a server instance to use availability groups as a high availability and disaster recovery solution. For more information about managing the AlwaysOn Availability Groups, see Enable and Disable AlwaysOn Availability Groups (SQL Server).

Using SQL Server Configuration Manager Remotely

These steps are meant to be performed on a PC running the client edition of Windows 7 or later, or another server that has the Server Graphical Shell installed (i.e. a full installation of Windows Server 2008 R2 or a Windows Server 8 installation with the Server Graphical Shell feature enabled).

1. Open Computer Management. To open Computer Management do one of the following:

   a. On Windows 7, Windows Server 2008, or Windows Server 2008 R2:

a. Click Start, click All Programs, click Administrative Tools, and then click Computer Management.

b. Click Start, click Run, type COMPMGMT.MSC, and then click OK.

b. On Windows 8 with Server Graphical Shell enabled:

a. Move your mouse to the bottom-left corner of the screen and right-click when you see the Start overlay.

b. Select Computer Management from the context menu.

2. In the console tree, right-click Computer Management, and then click Connect to another computer.
3. In the Select Computer dialog box, type the name of the Server Core machine that you want to manage, or click Browse to find it, and then click OK.
4. In the console tree, under Computer Management of the Server Core machine, click Services and Applications.
5. Double click SQL Server Configuration Manager.
6. In SQL Server Configuration Manager, click SQL Server Services, right-click SQL Server (<instance name>), where <instance name> is the name of a local server instance for which you want to enable AlwaysOn Availability Groups, and click Properties.
7. Select the AlwaysOn High Availability tab.
8. Verify that Windows failover cluster name field contains the name of the local failover cluster node. If this field is blank, this server instance currently does not support AlwaysOn Availability Groups. Either the local computer is not a cluster node, the WSFC cluster has been shut down, or this edition of SQL Server 2014 that does not support AlwaysOn Availability Groups.
9. Select the Enable AlwaysOn Availability Groups check box, and click OK.
10. SQL Server Configuration Manager saves your change. Then, you must manually restart the SQL Server service. This enables you to choose a restart time that is best for your business requirements. When the SQL Server service restarts, AlwaysOn will be enabled, and the IsHadrEnabled server property will be set to 1.

> **NOTE**
>
> - You must have the appropriate user rights or you must have been delegated the appropriate authority on the target computer to connect to that computer.
> - The name of the computer that you are managing appears in parentheses next to Computer Management in the console tree.

Using PowerShell Cmdlets to Enable AlwaysOn Availability Groups

The PowerShell Cmdlet, Enable-SqlAlwaysOn, is used to enable AlwaysOn Availability Group on an instance of SQL Server. If AlwaysOn Availability Groups is enable while the SQL Server service is running, the Database Engine service must be restarted for the change to complete. Unless you specify the -Force parameter, the cmdlet prompts you to ask whether you wish to restart the service; if cancelled, no operation occurs.

You must have Administrator permissions to execute this cmdlet.

You can use one of the following syntaxes to enable AlwaysOn Availability Groups for an instance of SQL Server:

```
Enable-SqlAlwaysOn [-Path <string>] [-Credential <PSCredential>] [-Force] [-NoServiceRestart] [-Confirm] [-
WhatIf] [<Commom Parameters>]
```

```
Enable-SqlAlwaysOn -InputObject <Server> [-Credential <PSCredential>] [-Force] [-NoServiceRestart] [-Confirm]
[-WhatIf] [<Commom Parameters>]
```

```
Enable-SqlAlwaysOn [-ServerInstance <string>] [-Credential <PSCredential>] [-Force] [-NoServiceRestart] [-
Confirm] [-WhatIf] [<Commom Parameters>]
```

The following PowerShell command enables AlwaysOn Availability Groups on an instance of SQL Server (Machine\Instance):

```
Enable-SqlAlwaysOn -Path SQLSERVER:\SQL\Machine\Instance
```

Configuring remote access of SQL Server running on Server Core

Perform the actions described below to configure remote access of a SQL Server 2014 instance that is running on Windows Server 2008 R2 Server Core SP1.

Enable remote connections on an instance of SQL Server

To enable remote connections, use SQLCMD.exe locally and execute the following statements against the Server Core instance:

- `EXEC sys.sp_configure N'remote access', N'1'`

  `GO`

- `RECONFIGURE WITH OVERRIDE`

  `GO`

Enable and start the SQL Server Browser service

By default, the Browser service is disabled. If it is disabled on an instance of SQL Server running on Server Core, run the following command from the command prompt to enable it:

```
sc config SQLBROWSER start= auto
```

After it is enabled, run the following command from the command prompt to start the service:

```
net start SQLBROWSER
```

Create exceptions in Windows Firewall

To create exceptions for SQL Server access in Windows Firewall, follow the steps specified in Configure the Windows Firewall to Allow SQL Server Access.

Enable TCP/IP on an instance of SQL Server

The TCP/IP protocol can be enabled through Windows PowerShell for an instance of SQL Server on Server Core. Follow these steps:

1. On the computer that is running Windows Server 2008 R2 Server Core SP1, launch Task Manager.
2. On the **Applications** tab, click **New Task**.
3. In the **Create New Task** dialog box, type **sqlps.exe** in the **Open** field and then click **OK**. This opens the **Microsoft SQL Server Powershell** window.

4. In the **Microsoft SQL Server Powershell** window, run the following script to enable the TCP/IP protocol:

```
$smo = 'Microsoft.SqlServer.Management.Smo.'
$wmi = new-object ($smo + 'Wmi.ManagedComputer')
# Enable the TCP protocol on the default instance.  If the instance is named, replace MSSQLSERVER with the
instance name in the following line.
$uri = "ManagedComputer[@Name='" + (get-item env:\computername).Value +
"']/ServerInstance[@Name='MSSQLSERVER']/ServerProtocol[@Name='Tcp']"
$Tcp = $wmi.GetSmoObject($uri)
$Tcp.IsEnabled = $true
$Tcp.Alter()
$Tcp
```

SQL Server Profiler

On a remote machine, start SQL Server Profiler and select New Trace from the File menu, the application displays a Connect to Server dialog box where you can specify the SQL Server instance, residing on the Server Core machine, to which you want to connect. For more information, see Start SQL Server Profiler.

For more information on the permissions required to run SQL Server Profiler, see Permissions Required to Run SQL Server Profiler.

For additional details about SQL Server Profiler, see SQL Server Profiler.

SQL Server Auditing

You can use SQL Server Management Studio or Transact-SQL remotely to define an audit. After the audit is created and enabled, the target will receive entries. For more information about creating and managing SQL Server audits, see SQL Server Audit (Database Engine).

SQL Servevr Command Prompt Utilities

You can use the following command prompt utilities that enable you to script SQL Server operations on a Server Core machine. The following table contains a list of command prompt utilities that ship with SQL Server for Server Core:

UTILITY	DESCRIPTION	INSTALLED IN
bcp Utility	Used to copy data between an instance of MicrosoftSQL Server and a data file in a user-specified format.	<*drive*>:\Program Files\Microsoft SQL Server\120\Tools\Binn
dtexec Utility	Used to configure and execute an Integration Services package.	<*drive*>:\Program Files\Microsoft SQL Server\120\DTS\Binn
dtutil Utility	Used to manage SSIS packages.	<*drive*>:\Program Files\Microsoft SQL Server\120\DTS\Binn
osql Utility	Allows you to enter Transact-SQL statements, system procedures, and script files at the command prompt.	<*drive*>:\Program Files\Microsoft SQL Server\120\Tools\Binn
sqlagent90 Application	Used to start SQL Server Agent from a command prompt.	<drive>:\Program Files\MicrosoftSQL Server\<*instance_name*>\MSSQL\Binn

UTILITY	DESCRIPTION	INSTALLED IN
sqlcmd Utility	Allows you to enter Transact-SQL statements, system procedures, and script files at the command prompt.	*<drive>*:\Program Files\Microsoft SQL Server\120\Tools\Binn
SQLdiag Utility	Used to collect diagnostic information for Microsoft Customer Service and Support.	*<drive>*:\Program Files\Microsoft SQL Server\120\Tools\Binn
sqlmaint Utility	Used to execute database maintenance plans created in previous versions of SQL Server.	<drive>:\Program Files\MicrosoftSQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn
sqlps Utility	Used to run PowerShell commands and scripts. Loads and registers the SQL Server PowerShell provider and cmdlets.	*<drive>*:\Program Files\Microsoft SQL Server\120\Tools\Binn
sqlservr Application	Used to start and stop an instance of Database Engine from the command prompt for troubleshooting.	<drive>:\Program Files\MicrosoftSQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn

Use troubleshooting tools

You can use SQLdiag Utility to collect logs and data files from SQL Server and other types of servers, and use it to monitor your servers over time or troubleshoot specific problems with your servers. SQLdiag is intended to expedite and simplify diagnostic information gathering for Microsoft Customer Support Services.

You can launch the utility on the administrator command prompt on the Server Core, using the syntax specified in the topic: SQLdiag Utility.

See Also

Install SQL Server 2014 on Server Core
Installation How-to Topics

Configure the Windows Firewall to Allow SQL Server Access

11/5/2019 • 23 minutes to read • Edit Online

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.

> **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 Windows Firewall with Advanced Security and IPsec.

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:

- Configure a Windows Firewall for Database Engine Access
- Configure the Windows Firewall to Allow Analysis Services Access
- Configure a Firewall for Report Server Access

Basic 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:

Windows Firewall with Advanced Security Getting Started Guide

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.

> **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.

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.

  > **IMPORTANT**
  > 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 previously-configured profile will not be in effect. For more information about profiles, see Windows Firewall with Advanced Security Getting Started Guide.

  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 presents most of the firewall options in an easy-to-use manner, and presents all

firewall profiles. For more information, see Using the Windows Firewall with Advanced Security Snap-in 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 or using a batch file**.** 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. Windows Server 2008 also has an advanced firewall helper called **advfirewall**. The details of using **netsh** are not discussed in this topic. However, many of the configuration options described can be configured by using **netsh**. For example, run the following script at a command prompt to open TCP port 1433:

  ```
  netsh firewall set portopening protocol = TCP port = 1433 name = SQLPort mode = ENABLE scope = SUBNET
  profile = CURRENT
  ```

  A similar example using the Windows Firewall for Advanced Security helper:

  ```
  netsh advfirewall firewall add rule name = SQLPort dir = in protocol = tcp action = allow localport =
  1433 remoteip = localsubnet profile = DOMAIN
  ```

  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
  - The "netsh firewall" command together with the "profile=all" parameter does not configure the public profile on a Windows Vista-based computer

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 Dynamic Ports later in this topic.)

SCENARIO	PORT	COMMENTS
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 Dynamic Ports. 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 Connection	TCP port 1434 for the default instance. Other ports are used for named instances. Check the error log for the port number.	By default, remote connections to the Dedicated Administrator Connection (DAC) are not enabled. To enable remote DAC, use the Surface Area Configuration facet. For more information, see Surface Area Configuration.
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 instance running over an HTTP endpoint.	Can be specified when an HTTP endpoint is created. The default is TCP port 80 for CLEAR_PORT traffic and 443 for SSL_PORT traffic.	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. To verify the port used, execute the following query:
	`SELECT name, protocol_desc, port, state_desc`
	`FROM sys.tcp_endpoints`
	`WHERE type_desc = 'SERVICE_BROKER'`
		There is no default port for SQL ServerService Broker, but this is the conventional configuration used in Books Online examples.

SCENARIO	PORT	COMMENTS

Database Mirroring	Administrator chosen port. To determine the port, execute the following query:
	
	`SELECT name, protocol_desc, port, state_desc FROM sys.tcp_endpoints`
	
	`WHERE type_desc = 'DATABASE_MIRRORING'`
		There is no default port for database mirroring however Books Online examples use TCP port 7022. 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 breaking quorum. For more information, see Specify a Server Network Address (Database Mirroring).

Replication	Replication connections to SQL Server use the typical regular Database Engine ports (TCP port 1433 for the default instance, 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 Sharing. File sharing uses UDP port 137 and 138, and TCP port 139 if it using NetBIOS. File Sharing uses TCP port 445.	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.
Transact-SQL debugger	TCP port 135 See Special Considerations for Port 135 The IPsec exception might also be required.	If using Visual Studio, on the Visual Studio host computer, you must also add **Devenv.exe** to the Exceptions list and open TCP port 135. If using Management Studio, on the Management Studio host computer, you must also add **ssms.exe** to the Exceptions list and open TCP port 135. For more information, see Configure the Transact-SQL Debugger.

For step by step instructions to configure the Windows Firewall for the Database Engine, see Configure a Windows Firewall for Database Engine Access.

Dynamic Ports

By default, 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 used, 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 Configure a Server to Listen on a Specific TCP Port (SQL Server Configuration Manager).

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 snap-in. 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.

> **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 Using the Windows Firewall with Advanced Security Snap-in 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**.
2. 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\MSSQL12.<instance_name>\MSSQL\Binn**, select **sqlservr.exe**, and then click **Open**.
3. Click **OK**.

For more information about endpoints, see Configure the Database Engine to Listen on Multiple TCP Ports and Endpoints Catalog Views (Transact-SQL).

Ports Used By Analysis Services

The following table lists the ports that are frequently 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 configured for use through IIS/HTTPS (The PivotTable?? Service uses HTTP or HTTPS)	TCP port 443	Used for an HTTPS connection through a URL. HTTPS is an HTTP connection that uses secure sockets layer (SSL).

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 Configure the Windows Firewall to Allow 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 **World Wide Web Services (HTTP)**. For more information, see the Interaction with Other Firewall Rules 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 **Secure World Wide Web Services (HTTPS)**. For more information, see the Interaction with Other Firewall Rules 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, Configure a Firewall for Report Server Access.

Ports Used By Integration Services

The following table lists the ports that are used by the Integration Services service.

FEATURE	PORT	COMMENTS
Microsoft remote procedure calls (MS RPC) Used by the Integration Services runtime.	TCP port 135 See Special Considerations for Port 135	The Integration Services service uses DCOM on port 135. The Service Control Manager uses port 135 to perform tasks such as starting and stopping the Integration Services service and transmitting control requests to the running service. The port number cannot be changed. This port is only required to be open if you are connecting to a remote instance of the Integration Services service from Management Studio or a custom application.

For step-by-step instructions to configure the Windows Firewall for Integration Services, see Configure a Windows Firewall for Access to the SSIS Service.

Additional Ports and Services

The following table lists ports and services that SQL Server might depend on.

SCENARIO	PORT	COMMENTS
Windows Management Instrumentation For more information about WMI, see WMI Provider for Configuration Management Concepts	WMI runs as part of a shared service host with ports assigned through DCOM. WMI might be using TCP port 135. See Special Considerations for Port 135	SQL Server Configuration Manager uses WMI to list and manage services. We recommend that you use the preconfigured rule group **Windows Management Instrumentation (WMI)**. For more information, see the Interaction with Other Firewall Rules section below.

SCENARIO	PORT	COMMENTS
Microsoft Distributed Transaction Coordinator (MS DTC)	TCP port 135 See Special Considerations 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 **Distributed Transaction Coordinator** rule group. When a single shared MS DTC is configured for the entire cluster in a separate resource group you should add sqlservr.exe as an exception to the firewall.
The browse button in Management Studio uses UDP to connect to the SQL Server Browser Service. For more information, see SQL Server Browser Service (Database Engine and SSAS).	UDP port 1434	UDP is a connectionless protocol. 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: If the setting is TRUE, no unicast responses to a broadcast are permitted at all. Enumerating services will fail. 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.
IPsec traffic	UDP port 500 and UDP port 4500	If the domain policy requires network communications to be done through IPsec, you must also add UDP port 4500 and UDP port 500 to the exception list. IPsec is an option using the **New Inbound Rule Wizard** in the Windows Firewall snap-in. For more information, see Using the Windows Firewall with Advanced Security Snap-in below.
Using Windows Authentication with Trusted Domains	Firewalls must be configured to allow authentication requests.	For more information, see How to configure a firewall for domains and trusts.

SCENARIO	PORT	COMMENTS
SQL Server and Windows Clustering	Clustering requires additional ports that are not directly related to SQL Server.	For more information, see Enable a network for cluster use.
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 Configuring HTTP and HTTPS.	For SQL Server specific information about reserving an HTTP.SYS endpoint using HttpCfg.exe, see About URL Reservations and Registration (SSRS Configuration Manager).

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. Because port 135 is used for many services it is frequently attacked by malicious users. When opening port 135, consider restricting the scope of the firewall rule.

For more information about port 135, see the following references:

- Service overview and network port requirements for the Windows Server system
- Troubleshooting RPC Endpoint Mapper errors using the Windows Server 2003 Support Tools from the product CD
- Remote procedure call (RPC)
- How to configure RPC dynamic port allocation to work with firewalls

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.

Overview of Firewall Profiles

Firewall profiles are discussed in Windows Firewall with Advanced Security Getting Started Guide in the section **Network location-aware host firewall**. To summarize, the operating systems 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.

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.

> **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.

Using the Windows Firewall with Advanced Security Snap-in

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, right-click **Inbound Rules**, and then click **New Rule**.
3. Complete the **New Inbound Rule Wizard** using the settings that you want.

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 UDP 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 Microsoft Download Center.

See Also

Service overview and network port requirements for the Windows Server system

Configure a Multi-Homed Computer for SQL Server Access

12/3/2018 • 7 minutes to read • Edit Online

When a server must provide a connection to two or more networks or network subnets, a typical scenario uses a multi-homed computer. Frequently this computer is located in a perimeter network (also known as DMZ, demilitarized zone, or screened subnet). This topic describes how to configure SQL Server and Windows Firewall with Advanced Security to provide for network connections to an instance of SQL Server in a multi-homed environment.

> **NOTE**
> A multi-homed computer has multiple network adapters or has been configured to use multiple IP addresses for a single network adapter. A dual-homed computer has two network adapters or has been configured to use two IP addresses for a single network adapter.

Before you continue in this topic, you should be familiar with the information provided in the topic Configure the Windows Firewall to Allow SQL Server Access. This topic contains basic information about how SQL Server components work with the firewall.

Assumptions for this example:

- There are two network adapters installed in the computer. One or more of the network adapters can be wireless. You can simulate having two network adapters by using the IP address of one network adapter, and using the loopback IP address (127.0.0.1) as the second network adapter.
- For simplicity, this example uses IPv4 addresses. The same procedures can be performed by using IPv6 addresses.

  > **NOTE**
  > IPv4 addresses are a series of four numbers known as octets. Each number is less than 255, separated by periods, such as 127.0.0.1. IPv6 addresses are a series of eight hexadecimal numbers separated by colons, such as fe80:4898:23:3:49a6:f5c1:2452:b994.

- Firewall rules could allow access through a specific port, such as port 1433. Or firewall rules could allow access to the SQL Server Database Engine program (sqlservr.exe). Neither method is better than the other. Because a server in a perimeter network is more vulnerable to attack than servers on an intranet, this topic assumes that you want to have more precise control, and individually select the ports that you open. For that reason, this topic assumes that you will configure SQL Server to listen on a fixed port. For more information about the ports that SQL Server uses, see Configure the Windows Firewall to Allow SQL Server Access.
- This example configures access to the Database Engine by using TCP port 1433. The other ports that are the different SQL Server components use can be configured by using the same general steps.

The general steps in this example are as follows:

- Determine the IP addresses on the computer.
- Configure SQL Server to listen on a specific TCP port.

- Configure Windows Firewall with Advanced Security.

Optional Procedures

If you already know the IP addresses available to your computer and that are used by SQL Server, you can skip these procedures.

To determine the IP addresses available on the computer

1. On the computer on which SQL Server is installed, click **Start**, click **Run**, type `cmd` and then Click **OK**..
2. In the Command Prompt window, type `ipconfig,` and then press ENTER to list the IP addresses available on this computer.

   > **NOTE**
   > The **ipconfig** command sometimes lists many possible connections, including connections that are disconnected. The **ipconfig** command can list both IPv4 and IPv6 addresses.

3. Note the IPv4 addresses and IPv6 addresses that are being used. The other information in the list, such as temporary addresses, subnet masks, and default gateways is important information for configuring a TCP/IP network. But this information is not used in this example.

To determine the IP addresses and ports used by SQL Server

1. Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2014**, point to **Configuration Tools**, and then click **SQL Server Configuration Manager**.
2. In **SQL Server Configuration Manager**, in the console pane, expand **SQL Server Network Configuration**, expand **Protocols for <instance name>**, and then double-click **TCP/IP**.
3. In the **TCP/IP Properties** dialog box, on the **IP Addresses** tab, several IP addresses appear in the format **IP1**, **IP2**, up to **IPAll**. One of these is for the IP address of the loopback adapter, 127.0.0.1. Additional IP addresses appear for each IP Address configured on the computer.
4. For any IP address if the **TCP Dynamic Ports** dialog box contains **0**, this indicates that the Database Engine is listening on dynamic ports. This example uses fixed ports instead of dynamic ports which could change upon restart. Therefore if the **TCP Dynamic Ports** dialog box contains **0**, delete the 0.
5. Note the TCP port that is listed for each IP address that you want to configure. For this example, assume that both IP addresses are listening on the default port, 1433.
6. If you do not want SQL Server to use some of the available ports, on the **Protocol** tab, change the **Listen All** value to **No**; and on the **IP Addresses** tab, change the **Active** value to **No** for the IP addresses that you do not want to use.

Configuring Windows Firewall with Advanced Security

After you know the IP addresses that the computer uses and the ports that SQL Server uses, you can create firewall rules, and then configure those rules for specific IP addresses.

To create a firewall rule

1. On the computer on which SQL Server is installed, log on as an administrator..
2. Click **Start**, click **Run**, type `wf.msc`, and click **OK**.
3. In the **User Account Control** dialog box, click **Continue** to use the Administrator credentials to open the Windows Firewall with Advanced Security snap-in.
4. On the **Overview** page, confirm that the Windows Firewall is enabled.

5. In the left pane, click **Inbound Rules**.
6. Right-click **Inbound Rules**, and then click **New Rule** to open the **New Inbound Rule Wizard**.
7. You could create a rule for the SQL Server program. However, because this example uses a fixed port, select **Port**, and then click **Next**.
8. On the **Protocols and Ports** page, select **TCP**.
9. Select **Specified local ports**. Type the port numbers separated by commas, and then click **Next**. In this example, you will configure the default port; therefore, enter `1433`.
10. On the **Action** page, review the options. In this example, you are not using the firewall to force secure connections. Therefore, click **Allow the connection**, and then click **Next**.

    > **NOTE**
    > Your environment might require secure connections. If you select one of the secure connections options, you might have to configure a certificate and the **Force Encryption** option. For more information about secure connections, see Enable Encrypted Connections to the Database Engine (SQL Server Configuration Manager) and Enable Encrypted Connections to the Database Engine (SQL Server Configuration Manager).

11. On the **Profile** page, select one or more profiles for the rule. If you are unfamiliar with firewall profiles, click the **Learn more about profiles** link in the firewall program.
    - If the computer is a server and is available only when it is connected to a domain, select **Domain**, and then click **Next**.
    - If the computer is a mobile computer (for example a laptop), it is likely to use multiple profiles when it connects to different networks. For a mobile computer, you can configure different access capabilities for different profiles. For example, you might allow access when the computer uses the Domain profile but not allow access when it uses the Public profile.
12. On the **Name** page, provide a name and description for the rule, and then click **Finish**.
13. Repeat this procedure to create another rule for each IP address that SQL Server will use.

After you have created one or more rules, perform the following steps to configure each IP address on the computer to use a rule.

To configure the firewall rule for a specific IP addresses

1. On the **Inbound Rules** page of the **Windows Firewall with Advanced Security**, right-click the rule that you just created, and then click **Properties**.
2. In the **Rule Properties** dialog box, select the **Scope** tab.
3. In the **Local IP address** area, select **These IP addresses**, and then click **Add**.
4. In the **IP Address** dialog box, select **This IP address or subnet**, and then type one of the IP addresses that you want to configure.
5. Click **OK**.
6. In the **Remote IP address** area, select **These IP addresses**, and then click **Add**.
7. Use the **IP Address** dialog box to configure connectivity for the selected IP address on the computer. You can enable connections from specified IP addresses, ranges of IP addresses, whole subnets, or from certain computers. To configure this option correctly, you must have a good understanding of the network. For information about the network, see the network administrator.

8. To close the **IP Address** dialog box, click **OK**; and then click **OK** to close the **Rule Properties** dialog box.
9. To configure the other IP addresses on a multi-homed computer, repeat this procedure by using another IP address and another rule.

See Also

SQL Server Browser Service (Database Engine and SSAS)
Connect to SQL Server Through a Proxy Server (SQL Server Configuration Manager)

Installation How-to Topics

5/22/2019 • 2 minutes to read • Edit Online

Use the following topics to install SQL Server 2014.

In This Section

TOPIC	DESCRIPTION
Install SQL Server 2014 from the Installation Wizard (Setup)	Procedural topic for a typical SQL Server installation by using the Installation Wizard.
Install SQL Server 2014 from the Command Prompt	Procedural topic that provides sample syntax and installation parameters for running unattended Setup.
Install SQL Server 2014 Using a Configuration File	Procedural topic that provides sample syntax and installation parameters for running Setup through a configuration file.
Install PowerPivot for SharePoint 2010	Procedural topic that provides links to additional procedures for installing PowerPivot for SharePoint in different SharePoint server configurations.
Add Features to an Instance of SQL Server 2014 (Setup)	Procedural topic for updating components of an existing instance of SQL Server 2014.
Drop a SQL Server 2014 Installation	Procedural topic for repairing a corrupt SQL Server 2014 installation.
Rename a Computer that Hosts a Stand-Alone Instance of SQL Server	Procedural topic for updating system metadata that is stored in sys.servers.
View and Read SQL Server Setup Log Files	Procedural topic for checking errors in setup log files.
Uninstall an Existing Instance of SQL Server (Setup)	Procedural topic for manually uninstalling a stand-alone instance of SQL Server.
Upgrade How-to Topics	Procedural topics for upgrading a SQL Server stand-alone or cluster instance.

See Also

Planning a SQL Server Installation
Upgrade to SQL Server 2014
Installing SQL Server Native Client

Install SQL Server 2014 from the Installation Wizard (Setup)

12/13/2018 • 11 minutes to read • Edit Online

This topic provides a step-by-step procedure for installing a new instance of SQL Server 2014 using the SQL Server setup installation wizard. The SQL Server Installation Wizard provides a single feature tree for installation of all SQL Server components so that you do not have to install them individually. For more information on various components that can be installed, see Installation for SQL Server 2014. For more information about how to install the SQL Server components individually, see Install SQL Server 2014.

These additional topics document other ways to install SQL Server:

- Install SQL Server 2014 from the Command Prompt.
- Install SQL Server 2014 Using a Configuration File
- Install SQL Server 2014 Using SysPrep
- Create a New SQL Server Failover Cluster (Setup).
- Upgrade to SQL Server 2014 Using the Installation Wizard (Setup).

Prerequisites

Before you install SQL Server, review topics in Planning a SQL Server Installation.

> **NOTE**
> For local installations, you must run Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.

To install SQL Server 2014

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.
2. The Installation Wizard runs the SQL Server Installation Center. To create a new installation of SQL Server, click **Installation** in the left-hand navigation area, and then click **New SQL Server stand-alone installation or add features to an existing installation**.
3. On the Product Key page, select an option 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 Editions and Components of SQL Server 2014.

   To continue, click **Next**.
4. On the License Terms page, review the license agreement and, if you agree, select the **I accept the license terms** check box, and then click **Next**. To help improve SQL Server, you can also enable the feature usage option and send reports to Microsoft.
5. In the Global Rules window, the setup procedure will automatically advance to the Product Updates window if there are no rule errors.
6. The Microsoft Update page will appear next if the Microsoft Update check box in Control Panel\All Control

Panel Items\Windows Update\Change settings is not checked. Putting a check in the Microsoft Update page will change the computer settings to include the latest updates when you scan for Windows Update.

7. On the Product Updates page, the latest available SQL Server product updates are displayed. If no product updates are discovered, SQL Server Setup does not display this page and auto advances to the **Install Setup Files** page.

8. On the Install Setup files page, Setup provides the progress of downloading, extracting, and installing the Setup files. If an update for SQL Server Setup is found, and is specified to be included, that update will also be installed.

9. On the Setup Role page, select **SQL Server Feature Installation**, and then click **Next** to continue to the Feature Selection page.

10. On the Feature Selection page, select the components for your installation. A description for each component group appears in the **Feature description** 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 2014 and Features Supported by the Editions of SQL Server 2014.

    The prerequisites for the selected features are displayed in the **Prerequisites for selected features** pane. SQL Server Setup will install the prerequisite that are not already installed during the installation step described later in this procedure.

    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 in the field at the bottom of the dialog box, or click **Browse** to move to an installation directory. The default installation path is C:\Program Files\Microsoft SQL Server\120\.

    The path specified for the shared components must be an absolute path. The folder must not be compressed or encrypted. Mapped drives are not supported.

    If you are installing SQL Server on a 64-bit operating system, you will see the following options:

    a. Shared feature directory

    b. Shared feature directory (x86)

    The path specified for each of the above options must be different.

11. The Feature Rules window will automatically advance if all rules pass.

12. On the Instance Configuration page, specify whether to install a default instance or a named instance. For more information, see Instance Configuration.

    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 a different value for **Instance ID** text box.

    > **NOTE**
    > Typical stand-alone instances of SQL Server 2014, whether default or named instances, do not use a non-default value for the **Instance ID**.

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

    Installed instances - 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 2014.

The Work flow for the rest of the installation depends on the features that you have specified for your installation. You might not see all the pages, depending on your selections.

13. Use the Server Configuration - Service Accounts page to specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features that 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 have to have to complete their tasks. For more information, see Server Configuration - Service Accounts and Configure Windows Service Accounts and Permissions.

    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.

    > **IMPORTANT**
    > Do not use a blank password. Use a strong password.

    Use the Server Configuration - Collation page to specify non-default collations for the Database Engine and Analysis Services. For more information, see Server Configuration - Collation.

14. Use the Database Engine Configuration - Server Configuration 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.

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

    > **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. Further, installing SQL Server with the Data Directories on the root folder of a drive or a mount point will fail. For more information, please review SQL Server support for mounted volumes.

    For more information, see Database Engine Configuration - Data Directories.

    Use the Database Engine Configuration - FILESTREAM page to enable FILESTREAM for your instance of SQL Server. For more information, see Database Engine Configuration - Filestream.

15. Use the Analysis Services Configuration - Account Provisioning page to specify the server mode and the users or accounts that will have administrator permissions for Analysis Services. Server mode determines

which memory and storage subsystems are used on the server. Different solution types run in different server modes. If you plan to run multidimensional cube databases on the server, choose the default option, Multidimensional and Data Mining server mode. Regarding administrator permissions, 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 about server mode and administrator permissions, see Analysis Services 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**.

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

> **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. Further, installing SQL Server with the Data Directories on the root folder of a drive or a mount point will fail. For more information, please review SQL Server support for mounted volumes.

For more information, see Analysis Services Configuration - Data Directories.

16. Use the Reporting Services Configuration page to specify the kind of Reporting Services installation to create. For more information about Reporting Services configuration modes and the options we have, see Reporting Services Configuration Options (SSRS).

    After you choose an option, click **Next** to continue.

17. Use the Distributed Replay Controller Configuration page to specify the users you want to grant administrative permissions to for the Distributed Replay controller service. Users that have administrative permissions will have unlimited access to the Distributed Replay controller service.

    Click the **Add Current User** button to add the users to whom you want to grant access permissions for the Distributed Replay controller service. Click the **Add** button to add access permissions for the Distributed Replay controller service. Click the **Remove** button to remove access permissions from the Distributed Replay controller service.

    To continue, click **Next**.

18. Use the Distributed Replay Client Configuration page to specify the users you want to grant administrative permissions to for the Distributed Replay client service. Users that have administrative permissions will have unlimited access to the Distributed Replay client service.

    Controller Name is an optional parameter, and the default value is <*blank*>. Enter the name of the controller that the client computer will communicate with for the Distributed Replay client service. Note the following:

    - If you have already set up a controller, enter the name of the controller while configuring each client.
    - If you have not yet set up a controller, you can leave the controller name blank. However, you must manually enter the controller name in the **client configuration** file.

    Specify the **Working Directory** for the Distributed Replay client service. The default working directory is <*drive letter*>:\Program Files\MicrosoftSQL Server\DReplayClient\WorkingDir\.

Specify the **Result Directory** for the Distributed Replay client service. The default result directory is *<drive letter>*:\Program Files\MicrosoftSQL Server\DReplayClient\ResultDir\.

To continue, click **Next**.

19. The Ready to Install page shows a tree view of installation options that were specified during Setup. On this page, Setup indicates whether the Product Update feature is enabled or disabled and the final update version.

    To continue, click **Install**. SQL Server Setup will first install the required prerequisites for the selected features followed by the feature installation.

20. During installation, the Installation Progress page provides status so that you can monitor installation progress as Setup continues.

21. 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**.

22. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you have finished with Setup. For more information, see View and Read SQL Server Setup Log Files.

Next Steps

Configure your new SQL Server installation.

To reduce the attackable surface area of a system, SQL Server selectively installs and enables key services and features. For more information, see Surface Area Configuration.

See Also

Validate a SQL Server Installation
Drop a SQL Server 2014 Installation
View and Read SQL Server Setup Log Files
Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)
Install SQL Server 2014 from the Command Prompt

Install SQL Server 2014 from the Command Prompt

10/23/2019 • 60 minutes to read • Edit Online

Before you run SQL Server Setup, review Planning a SQL Server Installation.

Installing a new instance of SQL Server at 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.

> **NOTE**
> When installing through the command prompt, SQL Server supports full quiet mode by using the /Q parameter, or Quiet Simple mode by using the /QS parameter. The /QS switch only shows progress, does not accept any input, and displays no error messages if encountered. The /QS parameter is only supported when /Action=install is specified.

Regardless of the installation method, you are required to confirm acceptance of the software license terms as an individual or on behalf of an entity, unless your use of the software is governed by a separate agreement such as a Microsoft volume licensing agreement or a third-party agreement with an ISV or OEM.

The license terms are displayed for review and acceptance in the Setup user interface. Unattended installations (using the /Q or /QS parameters) must include the /IACCEPTSQLSERVERLICENSETERMS parameter. You can review the license terms separately at Microsoft Software License Terms.

> **NOTE**
> Depending on how you received the software (for example, through Microsoft volume licensing), your use of the software may be subject to additional terms and conditions.

Command prompt installation is supported in the following scenarios:

- Installing, upgrading, or removing an instance and shared components of SQL Server on a local computer by using syntax and parameters specified at the command prompt.
- Installing, upgrading, or removing a failover cluster instance.
- Upgrading from one SQL Server edition to another edition of SQL Server.
- Installing an instance of SQL Server on a local computer by using 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 at the command prompt, specify Setup parameters for your installation at the command prompt as part of your installation syntax.

> **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.

Proper Use of Setup Parameters

Use the following guidelines to develop installation commands that have correct syntax:

- /PARAMETER
- /PARAMETER=true/false
- /PARAMETER=1/0 for Boolean types
- /PARAMETER="value" for all single-value parameters. Using double quotation marks is recommended, but required if the value contains a space
- /PARAMETER="value1" "value2" "value3" for all multiple-value parameters. Using double quotation marks is recommended, but required if the value contains a space

Exceptions:

- /FEATURES, which is a multivalued parameter, but its format is /FEATURES=AS,RS,IS without a space, comma-delimited

Examples:

- /INSTANCEDIR=c:\Path is supported.
- /INSTANCEDIR="c:\Path" is supported

> **NOTE**
>
> - The relational server values support the additional terminating backslash formats (backslash or two backslash characters) for the path.
> - /PID, the value for this parameter should be enclosed in double quotation marks.

SQL Server Parameters

The following sections provide parameters to develop command line installation scripts for install, update, and repair scenarios.

Parameters that are listed for a SQL Server component are specific to that component. SQL Server Agent and SQL Server Browser parameters are applicable when you install the SQL Server Database Engine.

- Installation Parameters
- SysPrep Parameters
- Upgrade Parameters
- Repair Parameters
- Rebuild System Database Parameters

- Uninstall Parameters
- Failover Cluster Parameters
- Service Accounts Parameters
- Feature Parameters
- Role Parameters
- Controlling Failover Behavior using the /FAILOVERCLUSTERROLLOWNERSHIP Parameter
- Instance ID or InstanceID Configuration

Installation Parameters

Use the parameters in the following table to develop command line scripts for installation.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the installation workflow. Supported values: Install
SQL Server Setup Control	/IACCEPTSQLSERVERLICENSETERMS **Required only when the /Q or /QS parameter is specified for unattended installations.**	Required to acknowledge acceptance of the license terms.
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/UpdateEnabled **Optional**	Specify whether SQL Server setup should discover and include product updates. The valid values are True and False or 1 and 0. By default, SQL Server setup will include updates that are found.
SQL Server Setup Control	/UpdateSource **Optional**	Specify the location where SQL Server setup will obtain product updates. The valid values are "MU" to search Microsoft Update, a valid folder path, a relative path such as .\MyUpdates or a UNC share. By default, SQL Server setup will search Microsoft Update or a Windows Update Service through the Windows Server Update Services.
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ERRORREPORTING **Optional**	Specifies the error reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/FEATURES - Or - /ROLE **Required**	Specifies the components to install. Choose **/FEATURES** to specify individual SQL Server components to install. For more information, see Feature Parameters below. Choose Role Parameters to specify a setup role. Setup roles install SQL Server in a predetermined configuration.
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for installation parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file is piped to the console.
SQL Server Setup Control	/INSTALLSHAREDDIR **Optional**	Specifies a nondefault installation directory for 64-bit shared components. Default is %Program Files%\MicrosoftSQL Server Cannot be set to %Program Files(x86)%\MicrosoftSQL Server
SQL Server Setup Control	/INSTALLSHAREDWOWDIR **Optional**	Specifies a nondefault installation directory for 32-bit shared components. Supported only on a 64-bit system. Default is %Program Files(x86)%\MicrosoftSQL Server Cannot be set to %Program Files%\MicrosoftSQL Server
SQL Server Setup Control	/INSTANCEDIR **Optional**	Specifies a nondefault installation directory for instance-specific components.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/INSTANCEID **Optional**	Specifies a nondefault value for an InstanceID.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/PID **Optional**	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/QS **Optional**	Specifies that Setup runs and shows progress through the UI, but does not accept any input or show any error messages.
SQL Server Setup Control	/UIMODE **Optional**	Specifies whether to present only the minimum number of dialog boxes during setup. **/UIMode** can only be used with the **/ACTION=INSTALL** and **UPGRADE** parameters. Supported values: **/UIMODE=Normal** is the default for non-Express editions and presents all setup dialog boxes for the selected features. **/UIMODE=AutoAdvance** is the default for Express editions and skips nonessential dialog boxes. When combined with other parameters, **UIMODE** is overridden. For example, when **/UIMODE=AutoAdvance** and **/ADDCURRENTUSERASSQLADMIN=FALSE** are both provided, the provisioning dialog box is not auto populated with the current user. The **UIMode** setting cannot be used with the **/Q** or **/QS** parameters.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/SQMREPORTING **Optional**	Specifies feature usage reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.
SQL Server Agent	/AGTSVCACCOUNT **Required**	Specifies the account for the SQL Server Agent service.
SQL Server Agent	/AGTSVCPASSWORD Required	Specifies the password for SQL Server Agent service account.
SQL Server Agent	/AGTSVCSTARTUPTYPE **Optional**	Specifies the startup mode for the SQL Server Agent service. Supported values: Automatic Disabled Manual
Analysis Services	/ASBACKUPDIR **Optional**	Specifies the directory for Analysis Services backup files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Backup. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Backup.
Analysis Services	/ASCOLLATION **Optional**	Specifies the collation setting for Analysis Services. Default value: Latin1_General_CI_AS

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Analysis Services	/ASCONFIGDIR **Optional**	Specifies the directory for Analysis Services configuration files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Config. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Config.
Analysis Services	/ASDATADIR **Optional**	Specifies the directory for Analysis Services data files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Data. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Data.
Analysis Services	/ASLOGDIR **Optional**	Specifies the directory for Analysis Services log files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Log. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Log.
Analysis Services	/ASSERVERMODE **Optional**	Specifies the server mode of the Analysis Services instance. Valid values are MULTIDIMENSIONAL, POWERPIVOT or TABULAR. **ASSERVERMODE** is case-sensitive. All values must be expressed in upper case. For more information about valid values, see Install Analysis Services in Tabular Mode.
Analysis Services	/ASSVCACCOUNT **Required**	Specifies the account for the Analysis Services service.
Analysis Services	/ASSVCPASSWORD Required	Specifies the password for the Analysis Services service.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Analysis Services	/ASSVCSTARTUPTYPE **Optional**	Specifies the startup mode for the Analysis Services service. Supported values: Automatic Disabled Manual
Analysis Services	/ASSYSADMINACCOUNTS **Required**	Specifies the administrator credentials for Analysis Services.
Analysis Services	/ASTEMPDIR **Optional**	Specifies the directory for Analysis Services temporary files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server \ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Temp. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Temp.
Analysis Services	/ASPROVIDERMSOLAP **Optional**	Specifies whether the MSOLAP provider can run in-process. Default value: 1=enabled
Analysis Services	/FARMACCOUNT **Required for SPI_AS_NewFarm**	Specifies a domain user account for running SharePoint Central Administration services and other essential services in a farm. This parameter is used only for Analysis Services instances that are installed through Role Parameters= SPI_AS_NEWFARM.
Analysis Services	/FARMPASSWORD **Required for SPI_AS_NewFarm**	Specifies a password for the farm account.
Analysis Services	/PASSPHRASE **Required for SPI_AS_NewFarm**	Specifies a passphrase that is used to add additional application servers or Web front end servers to a SharePoint farm. This parameter is used only for Analysis Services instances that are installed through Role Parameters = SPI_AS_NEWFARM.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Analysis Services	/FARMADMINIPORT **Required for SPI_AS_NewFarm**	Specifies a port used to connect to the SharePoint Central Administration web application. This parameter is used only for Analysis Services instances that are installed through Role Parameters = SPI_AS_NEWFARM.
SQL Server Browser	/BROWSERSVCSTARTUPTYPE **Optional**	Specifies the startup mode for SQL Server Browser service. Supported values: Automatic Disabled Manual
SQL Server Database Engine	/ENABLERANU **Optional**	Enables run-as credentials for SQL Server Express installations.
SQL Server Database Engine	/INSTALLSQLDATADIR **Optional**	Specifies the data directory for SQL Server data files. Default values: For WOW mode on 64-bit:%Program Files(x86)%\MicrosoftSQL Server\ For all other installations:%Program Files%\MicrosoftSQL Server|
SQL Server Database Engine	/SAPWD **Required when /SECURITYMODE=SQL**	Specifies the password for the SQL Serversa account.
SQL Server Database Engine	/SECURITYMODE **Optional**	Specifies the security mode for SQL Server. If this parameter is not supplied, then Windows-only authentication mode is supported. Supported value: SQL
SQL Server Database Engine	/SQLBACKUPDIR **Optional**	Specifies the directory for backup files. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Backup
SQL Server Database Engine	/SQLCOLLATION **Optional**	Specifies the collation settings for SQL Server. The default value is based on the locale of your Windows operating system. For more information, see Collation Settings in Setup.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Database Engine	/ADDCURRENTUSERASSQLADMIN **Optional**	Adds the current user to the SQL Server `sysadmin` fixed server role. The /ADDCURRENTUSERASSQLADMIN parameter can be used when installing Express editions or when /Role=ALLFeatures_WithDefaults is used. For more information, see /ROLE below. Use of /ADDCURRENTUSERASSQLADMIN is optional, but either /ADDCURRENTUSERASSQLADMIN or /SQLSYSADMINACCOUNTS is required. Default values: True for editions of SQL Server Express False for all other editions
SQL Server Database Engine	/SQLSVCACCOUNT **Required**	Specifies the startup account for the SQL Server service.
SQL Server Database Engine	/SQLSVCPASSWORD Required	Specifies the password for SQLSVCACCOUNT.
SQL Server Database Engine	/SQLSVCSTARTUPTYPE **Optional**	Specifies the startup mode for the SQL Server service. Supported values: Automatic Disabled Manual
SQL Server Database Engine	/SQLSYSADMINACCOUNTS **Required**	Use this parameter to provision logins to be members of the sysadmin role. For SQL Server editions other than SQL Server Express, /SQLSYSADMINACCOUNTS is required. For editions of SQL Server Express, use of /SQLSYSADMINACCOUNTS is optional, but either /SQLSYSADMINACCOUNTS or /ADDCURRENTUSERASSQLADMIN is required.
SQL Server Database Engine	/SQLTEMPDBDIR **Optional**	Specifies the directory for the data files for tempdb. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Database Engine	/SQLTEMPDBLOGDIR **Optional**	Specifies the directory for the log files for tempdb. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLUSERDBDIR **Optional**	Specifies the directory for the data files for user databases. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLUSERDBLOGDIR **Optional**	Specifies the directory for the log files for user databases. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
FILESTREAM	/FILESTREAMLEVEL **Optional**	Specifies the access level for the FILESTREAM feature. Supported values: 0 =Disable FILESTREAM support for this instance. (Default value) 1=Enable FILESTREAM for Transact-SQL access. 2=Enable FILESTREAM for Transact-SQL and file I/O streaming access. (Not valid for cluster scenarios) 3=Allow remote clients to have streaming access to FILESTREAM data.
FILESTREAM	/FILESTREAMSHARENAME **Optional** **Required when FILESTREAMLEVEL is greater than 1.**	Specifies the name of the windows share in which the FILESTREAM data will be stored.
SQL Server Full Text	/FTSVCACCOUNT **Optional**	Specifies the account for Full-Text filter launcher service. This parameter is ignored in Windows Server 2008. ServiceSID is used to help secure the communication between SQL Server and Full-text Filter Daemon. If the values are not provided, the Full-text Filter Launcher Service is disabled. You have to use SQL Server Control Manager to change the service account and enable full-text functionality. Default value: Local Service Account

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Full Text	/FTSVCPASSWORD **Optional**	Specifies the password for the Full-Text filter launcher service. This parameter is ignored in Windows Server 2008.
Integration Services	/ISSVCACCOUNT **Required**	Specifies the account for Integration Services. Default value: NT AUTHORITY\NETWORK SERVICE
Integration Services	/ISSVCPASSWORD Required	Specifies the Integration Services password.
Integration Services	/ISSVCStartupType **Optional**	Specifies the startup mode for the Integration Services service.
SQL Server Network Configuration	/NPENABLED **Optional**	Specifies the state of the Named Pipes protocol for the SQL Server service. Supported values: 0=disable the Named Pipes protocol 1=enable the Named Pipes protocol
SQL Server Network Configuration	/TCPENABLED **Optional**	Specifies the state of the TCP protocol for the SQL Server service. Supported values: 0=disable the TCP protocol 1=enable the TCP protocol

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Reporting Services	/RSINSTALLMODE **Optional**	Specifies the Install mode for Reporting Services. Supported Values: SharePointFilesOnlyMode DefaultNativeMode FilesOnlyMode Note: If the installation includes the SQL ServerDatabase Engine, the default RSINSTALLMODE is DefaultNativeMode. If the installation does not include the SQL ServerDatabase Engine, the default RSINSTALLMODE is FilesOnlyMode. If you choose DefaultNativeMode but the installation does not include the SQL ServerDatabase Engine, the installation will automatically change the RSINSTALLMODE to FilesOnlyMode.
Reporting Services	/RSSVCACCOUNT **Required**	Specifies the startup account for Reporting Services.
Reporting Services	/RSSVCPASSWORD Required	Specifies the password for the startup account for the Reporting Services service.
Reporting Services	/RSSVCStartupType **Optional**	Specifies the startup mode for Reporting Services.

Sample Syntax:

To install a new, stand-alone instance with the SQL Server Database Engine, Replication, and Full-Text Search components.

```
setup.exe /q /ACTION=Install /FEATURES=SQL /INSTANCENAME=MSSQLSERVER /SQLSVCACCOUNT="
<DomainName\UserName>" /SQLSVCPASSWORD="<StrongPassword>" /SQLSYSADMINACCOUNTS="<DomainName\UserName>"
/AGTSVCACCOUNT="NT AUTHORITY\Network Service" /IACCEPTSQLSERVERLICENSETERMS
```

SysPrep Parameters

For more information about SQL Server SysPrep, see

Install SQL Server 2014 Using SysPrep.

Prepare Image Parameters

Use the parameters in the following table to develop command-line scripts for preparing an instance of SQL Server without configuring it.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the installation workflow.Supported values: PrepareImage
SQL Server Setup Control	/IACCEPTSQLSERVERLICENSETERMS **Required only when the /Q or /QS parameter is specified for unattended installations.**	Required to acknowledge acceptance of the license terms.
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/UpdateEnabled **Optional**	Specify whether SQL Server setup should discover and include product updates. The valid values are True and False or 1 and 0. By default, SQL Server setup will include updates that are found.
SQL Server Setup Control	/UpdateSource **Optional**	Specify the location where SQL Server setup will obtain product updates. The valid values are "MU" to search Microsoft Update, a valid folder path, a relative path such as .\MyUpdates or a UNC share. By default, SQL Server setup will search Microsoft Update or a Windows Update Service through the Windows Server Update Services.
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.
SQL Server Setup Control	/FEATURES **Required**	Specifies components to install. Supported values are SQLEngine, Replication, FullText, DQ, AS, AS_SPI, RS, RS_SHP, RS_SHPWFE, DQC, SSDTBI, Conn, IS, BC, SDK, BOL, SSMS, Adv_SSMS, DREPLAY_CTLR, DREPLAY_CLT, SNAC_SDK, SQLODBC, SQLODBC_SDK, LocalDB, MDS
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for installation parameters.
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file is piped to the console.
SQL Server Setup Control	/INSTALLSHAREDDIR **Optional**	Specifies a nondefault installation directory for 64-bit shared components. Default is %Program Files%\MicrosoftSQL Server Cannot be set to %Program Files(x86)%\MicrosoftSQL Server
SQL Server Setup Control	/INSTANCEDIR **Optional**	Specifies a nondefault installation directory for instance-specific components.
SQL Server Setup Control	/INSTANCEID Prior to SQL Server 2012 Service Pack 1 Cumulative Update 2 (January 2013) **Required** Beginning with SQL Server 2012 Service Pack 1 Cumulative Update 2 **Required** for instance features.	Specifies an InstanceID for the instance that is being prepared.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/QS **Optional**	Specifies that Setup runs and shows progress through the UI, but does not accept any input or show any error messages.

Sample Syntax:

To prepare a new, stand-alone instance with the SQL Server Database Engine, Replication, and Full-Text Search components, and Reporting Services.

```
setup.exe /q /ACTION=PrepareImage /FEATURES=SQL,RS /InstanceID =<MYINST> /IACCEPTSQLSERVERLICENSETERMS
```

Complete Image Parameters

Use the parameters in the following table to develop command-line scripts for completing and configuring a prepared instance of SQL Server.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the installation workflow. Supported values: CompleteImage

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/IACCEPTSQLSERVERLICENSETERMS **Required only when the /Q or /QS parameter is specified for unattended installations.**	Required to acknowledge acceptance of the license terms.
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.
SQL Server Setup Control	/ERRORREPORTING **Optional**	Specifies the error reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for installation parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file is piped to the console.
SQL Server Setup Control	/INSTANCEID Prior to SQL Server 2012 Service Pack 1 Cumulative Update 2 (January 2013) **Required** Beginning with SQL Server 2012 Service Pack 1 Cumulative Update 2 **Optional**	Use the Instance ID specified during the prepare image step. Supported Values: InstanceID of a Prepared Instance.
SQL Server Setup Control	/INSTANCENAME Prior to SQL Server 2012 Service Pack 1 Cumulative Update 2 (January 2013) **Required** Beginning with SQL Server 2012 Service Pack 1 Cumulative Update 2 **Optional**	Specifies a SQL Server instance name for the instance that is being completed. For more information, see Instance Configuration.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/PID **Optional**	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used. Note: If you are installing SQL Server Express, SQL Server Express with tools or SQL Server Express with Advanced Services, the PID is predefined.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/QS **Optional**	Specifies that Setup runs and shows progress through the UI, but does not accept any input or show any error messages.
SQL Server Setup Control	/SQMREPORTING **Optional**	Specifies feature usage reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.
SQL Server Agent	/AGTSVCACCOUNT **Required**	Specifies the account for the SQL Server Agent service.
SQL Server Agent	/AGTSVCPASSWORD Required	Specifies the password for SQL Server Agent service account.
SQL Server Agent	/AGTSVCSTARTUPTYPE **Optional**	Specifies the startup mode for the SQL Server Agent service. Supported values: Automatic Disabled Manual

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Browser	/BROWSERSVCSTARTUPTYPE **Optional**	Specifies the startup mode for SQL Server Browser service.Supported values: Automatic Disabled Manual
SQL Server Database Engine	/ENABLERANU **Optional**	Enables run-as credentials for SQL Server Express installations.
SQL Server Database Engine	/INSTALLSQLDATADIR **Optional**	Specifies the data directory for SQL Server data files. Default values: For WOW mode on 64-bit:%Program Files(x86)%\MicrosoftSQL Server\ For all other installations: %Program Files%\MicrosoftSQL Server|
SQL Server Database Engine	/SAPWD **Required when /SECURITYMODE=SQL**	Specifies the password for the SQL Serversa account.
SQL Server Database Engine	/SECURITYMODE **Optional**	Specifies the security mode for SQL Server. If this parameter is not supplied, then Windows-only authentication mode is supported. Supported value: SQL
SQL Server Database Engine	/SQLBACKUPDIR **Optional**	Specifies the directory for backup files. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Backup
SQL Server Database Engine	/SQLCOLLATION **Optional**	Specifies the collation settings for SQL Server. The default value is based on the locale of your Windows operating system. For more information, see Collation Settings in Setup.
SQL Server Database Engine	/SQLSVCACCOUNT **Required**	Specifies the startup account for the SQL Server service.
SQL Server Database Engine	/SQLSVCPASSWORD Required	Specifies the password for SQLSVCACCOUNT.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Database Engine	/SQLSVCSTARTUPTYPE **Optional**	Specifies the startup mode for the SQL Server service. Supported values: Automatic Disabled Manual
SQL Server Database Engine	/SQLSYSADMINACCOUNTS **Required**	Use this parameter to provision logins to be members of the sysadmin role.
SQL Server Database Engine	/SQLTEMPDBDIR **Optional**	Specifies the directory for the data files for tempdb. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLTEMPDBLOGDIR **Optional**	Specifies the directory for the log files for tempdb. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLUSERDBDIR **Optional**	Specifies the directory for the data files for user databases. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLUSERDBLOGDIR **Optional**	Specifies the directory for the log files for user databases. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
FILESTREAM	/FILESTREAMLEVEL **Optional**	Specifies the access level for the FILESTREAM feature. Supported values: 0 =Disable FILESTREAM support for this instance. (Default value) 1=Enable FILESTREAM for Transact-SQL access. 2=Enable FILESTREAM for Transact-SQL and file I/O streaming access. (Not valid for cluster scenarios) 3=Allow remote clients to have streaming access to FILESTREAM data.
FILESTREAM	/FILESTREAMSHARENAME **Optional** **Required when FILESTREAMLEVEL is greater than 1.**	Specifies the name of the windows share in which the FILESTREAM data will be stored.
SQL Server Full Text	/FTSVCACCOUNT **Optional**	Specifies the account for Full-Text filter launcher service. This parameter is ignored in Windows Server 2008. ServiceSID is used to help secure the communication between SQL Server and Full-text Filter Daemon. If the values are not provided, the Full-text Filter Launcher Service is disabled. You have to use SQL Server Control Manager to change the service account and enable full-text functionality. Default value: Local Service Account
SQL Server Full Text	/FTSVCPASSWORD **Optional**	Specifies the password for the Full-Text filter launcher service. This parameter is ignored in Windows Server 2008.
SQL Server Network Configuration	/NPENABLED **Optional**	Specifies the state of the Named Pipes protocol for the SQL Server service. Supported values: 0=disable the Named Pipes protocol 1=enable the Named Pipes protocol

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Network Configuration	/TCPENABLED **Optional**	Specifies the state of the TCP protocol for the SQL Server service. Supported values: 0=disable the TCP protocol 1=enable the TCP protocol
Reporting Services	/RSINSTALLMODE **Optional**	Specifies the Install mode for Reporting Services.
Reporting Services	/RSSVCACCOUNT **Required**	Specifies the startup account for Reporting Services.
Reporting Services	/RSSVCPASSWORD Required	Specifies the password for the startup account for the Reporting Services service.
Reporting Services	/RSSVCStartupType **Optional**	Specifies the startup mode for Reporting Services.

Sample Syntax:

To complete a prepared, stand-alone instance that includes SQL Server Database Engine, Replication, and Full-Text Search components.

```
setup.exe /q /ACTION=CompleteImage /INSTANCENAME=MYNEWINST /INSTANCEID=<MYINST> /SQLSVCACCOUNT="
<DomainName\UserName>" /SQLSVCPASSWORD="<StrongPassword>" /SQLSYSADMINACCOUNTS="<DomainName\UserName>"
/AGTSVCACCOUNT="NT AUTHORITY\Network Service" /IACCEPTSQLSERVERLICENSETERMS
```

Upgrade Parameters

Use the parameters in the following table to develop command-line scripts for upgrade.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the installation workflow. Supported values: Upgrade EditionUpgrade The value EditionUpgrade is used to upgrade an existing edition of SQL Server 2014 to a different edition. For more information about the supported version and edition upgrades, see Supported Version and Edition Upgrades.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/IACCEPTSQLSERVERLICENSETERMS **Required only when the /Q or /QS parameter is specified for unattended installations.**	Required to acknowledge acceptance of the license terms.
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/*UpdateEnabled* **Optional**	Specify whether SQL Server setup should discover and include product updates. The valid values are True and False or 1 and 0. By default, SQL Server setup will include updates that are found.
SQL Server Setup Control	/*UpdateSource* **Optional**	Specify the location where SQL Server setup will obtain product updates. The valid values are "MU" to search Microsoft Update, a valid folder path, a relative path such as .\MyUpdates or a UNC share. By default, SQL Server setup will search Microsoft Update or a Windows Update Service through the Windows Server Update Services.
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.
SQL Server Setup Control	/ERRORREPORTING **Optional**	Specifies the error reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for the parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file will be piped to the console.
SQL Server Setup Control	/ INSTANCEDIR **Optional**	Specifies a nondefault installation directory for shared components.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/INSTANCEID **Required when you upgrade from SQL Server 2008** or later. **Optional when you upgrade from SQL Server 2005**	Specifies a nondefault value for an InstanceID.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/PID **Optional**	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/UIMODE **Optional**	Specifies whether to present only the minimum number of dialog boxes during setup. **/UIMode** can only be used with the **/ACTION=INSTALL** and **UPGRADE** parameters. Supported values: **/UIMODE=Normal** is the default for non-Express editions and presents all setup dialog boxes for the selected features. **/UIMODE=AutoAdvance** is the default for Express editions and skips nonessential dialog boxes. The **UIMode** setting cannot be used with the **/Q** or **/QS** parameters.
SQL Server Setup Control	/SQMREPORTING **Optional**	Specifies feature usage reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies the console window would be hidden or closed.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Browser Service	/BROWSERSVCSTARTUPTYPE **Optional**	Specifies the startup mode for SQL Server Browser service. Supported values: Automatic Disabled Manual
SQL Server Full-Text	/FTUPGRADEOPTION **Optional**	Specifies the Full-Text catalog upgrade option. Supported values: REBUILD RESET IMPORT
Integration Services	/ISSVCACCOUNT **Required**	Specifies the account for Integration Services. Default value: NT AUTHORITY\NETWORK SERVICE
Integration Services	/ISSVCPASSWORD Required	Specifies the Integration Services password.
Integration Services	/ISSVCStartupType **Optional**	Specifies the startup mode for the Integration Services service.
Reporting Services	/RSUPGRADEDATABASEACCOUNT **Optional**	The property is only used when upgrading a SharePoint mode Report Server that is version 2008 R2 or earlier. Additional upgrade operations are performed for report servers that use the older SharePoint mode architecture, which was changed in SQL Server 2012 Reporting Services. If this option is not included with the command line installation, the default service account for the old report server instance is used. If this property is used, supply the password for the account using the **/RSUPGRADEPASSWORD** property.
Reporting Services	/RSUPGRADEPASSWORD **Optional**	Password of the existing Report Server service account.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Reporting Services	/ALLOWUPGRADEFORSSRSSHAREPOINTMODE	The switch is required when upgrading a SharePoint Mode installation that is based on the SharePoint shared service architecture. The switch is not needed for upgrading non-shared service versions of Reporting Services.

Sample Syntax:

To upgrade an existing instance or failover cluster node from a previous SQL Server version,

```
setup.exe /q /ACTION=upgrade /INSTANCEID = <INSTANCEID>/INSTANCENAME=MSSQLSERVER
/RSUPGRADEDATABASEACCOUNT="<Provide a SQL Server logon account that can connect to the report server
during upgrade>" /RSUPGRADEPASSWORD="<Provide a password for the report server upgrade account>"
/ISSVCAccount="NT Authority\Network Service" /IACCEPTSQLSERVERLICENSETERMS
```

Repair Parameters

Use the parameters in the following table to develop command-line scripts for repair.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the repair workflow. Supported values: Repair
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/FEATURES **Required**	Specifies components to repair.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.

Sample Syntax:

Repair an instance and shared components.

```
setup.exe /q /ACTION=Repair /INSTANCENAME=<instancename>
```

Rebuild System Database Parameters

Use the parameters in the following table to develop command-line scripts for rebuilding the master, model, msdb, and tempdb system databases. For more information, see Rebuild System Databases.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the rebuild database workflow. Supported values: Rebuilddatabase
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Database Engine	/SQLCOLLATION **Optional**	Specifies a new server-level collation. The default value is based on the locale of your Windows operating system. For more information, see Collation Settings in Setup.
SQL Server Database Engine	/SAPWD **Required when /SECURITYMODE=SQL was specified during Installation of the Instance.**	Specifies the password for SQL SA account.
SQL Server Database Engine	/SQLSYSADMINACCOUNTS **Required**	Use this parameter to provision logins to be members of the sysadmin role.

Uninstall Parameters

Use the parameters in the following table to develop command-line scripts for uninstallation.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the uninstall work flow. Supported values: Uninstall
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/FEATURES **Required**	Specifies components to uninstall.
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for the parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file will be piped to the console.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.

Sample Syntax:

To uninstall an existing instance of SQL Server.

```
setup.exe /Action=Uninstall /FEATURES=SQL,AS,RS,IS,Tools /INSTANCENAME=MSSQLSERVER
```

To remove a named instance, specify the name of the instance instead of "MSSQLSERVER" in the example that was mentioned earlier in this topic.

Failover Cluster Parameters

Before you install a SQL Server failover cluster instance, review the following topics:

- Hardware and Software Requirements for Installing SQL Server 2014
- Security Considerations for a SQL Server Installation
- Before Installing Failover Clustering
- AlwaysOn Failover Cluster Instances (SQL Server)

> **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.

Integrated Install Failover Cluster Parameters

Use the parameters in the following table to develop command-line scripts for failover cluster installation.

For more information about Integrated Installation, see AlwaysOn Failover Cluster Instances (SQL Server).

> **NOTE**
> To add more nodes after the installation, use Add Node action.

SQL SERVER COMPONENT	PARAMETER	DETAILS
SQL Server Setup Control	/ACTION **Required**	Required to indicate the failover cluster installation work flow. Supported values: InstallFailoverCluster
SQL Server Setup Control	/IACCEPTSQLSERVERLICENSETERMS **Required only when the /Q or /QS parameter is specified for unattended installations.**	Required to acknowledge acceptance of the license terms.
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/FAILOVERCLUSTERGROUP **Optional**	Specifies the name of the resource group to be used for the SQL Server failover cluster. It can be the name of an existing cluster group or the name of a new resource group. Default value: SQL Server (<InstanceName>)
SQL Server Setup Control	*/UpdateEnabled* **Optional**	Specify whether SQL Server setup should discover and include product updates. The valid values are True and False or 1 and 0. By default, SQL Server setup will include updates that are found.
SQL Server Setup Control	*/UpdateSource* **Optional**	Specify the location where SQL Server setup will obtain product updates. The valid values are "MU" to search Microsoft Update, a valid folder path, a relative path such as .\MyUpdates or a UNC share. By default, SQL Server setup will search Microsoft Update or a Windows Update Service through the Windows Server Update Services.
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.

SQL SERVER COMPONENT	PARAMETER	DETAILS
SQL Server Setup Control	/ERRORREPORTING **Optional**	Specifies the error reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/FEATURES **Required**	Specifies components to install.
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for the parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file will be piped to the console.
SQL Server Setup Control	/INSTALLSHAREDDIR **Optional**	Specifies a nondefault installation directory for 64-bit shared components. Default is %Program Files%\MicrosoftSQL Server Cannot be set to %Program Files(x86)%\MicrosoftSQL Server
SQL Server Setup Control	/INSTALLSHAREDWOWDIR **Optional**	Specifies a nondefault installation directory for 32-bit shared components. Supported only on a 64-bit system. Default is %Program Files(x86)%\MicrosoftSQL Server Cannot be set to %Program Files%\MicrosoftSQL Server
SQL Server Setup Control	/INSTANCEDIR **Optional**	Specifies nondefault installation directory for instance-specific components.
SQL Server Setup Control	/INSTANCEID **Optional**	Specifies a nondefault value for an InstanceID.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.

SQL SERVER COMPONENT	PARAMETER	DETAILS
SQL Server Setup Control	/PID **Optional**	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/QS **Optional**	Specifies that Setup runs and shows progress through the UI, but does not accept any input or show any error messages.
SQL Server Setup Control	/SQMREPORTING **Optional**	Specifies feature usage reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies the console window would be hidden or closed.
SQL Server Setup Control	/FAILOVERCLUSTERDISKS **Optional**	Specifies the list of shared disks to be included in the SQL Server failover cluster resource group. Default value: The first drive is used as the default drive for all databases.

SQL SERVER COMPONENT	PARAMETER	DETAILS
SQL Server Setup Control	/FAILOVERCLUSTERIPADDRESSES **Required**	Specifies an encoded IP address. The encodings are semicolon-delimited (;) and follow the format <IP Type>;<address>;<network name>;<subnet mask>. Supported IP types include DHCP, IPv4, and IPv6. You can specify multiple failover cluster IP addresses with a space in between. See the following examples: FAILOVERCLUSTERIPADDRESSES=DEFAULT FAILOVERCLUSTERIPADDRESSES=IPv4;DHCP;ClusterNetwork1 FAILOVERCLUSTERIPADDRESSES=IPv4;172.16.0.0;ClusterNetwork1;172.31.255.255 FAILOVERCLUSTERIPADDRESSES=IPv6;DHCP;ClusterNetwork1 FAILOVERCLUSTERIPADDRESSES=IPv6;2001:db8:23:1002:20f:1fff:feff:b3a3;ClusterNetwork1
SQL Server Setup Control	/FAILOVERCLUSTERNETWORKNAME **Required**	Specifies the network name for the new SQL Server failover cluster. This name is used to identify the new SQL Server failover cluster instance on the network.
SQL Server Agent	/AGTSVCACCOUNT **Required**	Specifies the account for the SQL Server Agent service.
SQL Server Agent	/AGTSVCPASSWORD Required	Specifies the password for SQL Server Agent service account.
Analysis Services	/ASBACKUPDIR **Optional**	Specifies the directory for Analysis Services backup files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\<INSTANCEDIR>\<ASInstanceID>\OLAP\Backup. For all other installations:%Program Files%\MicrosoftSQL Server\<INSTANCEDIR>\<ASInstanceID>\OLAP\Backup.

SQL SERVER COMPONENT	PARAMETER	DETAILS
Analysis Services	/ASCOLLATION **Optional**	Specifies the collation setting for Analysis Services. Default value: - Latin1_General_CI_AS
Analysis Services	/ASCONFIGDIR **Optional**	Specifies the directory for Analysis Services configuration files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Config. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Config.
Analysis Services	/ASDATADIR **Optional**	Specifies the directory for Analysis Services data files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Data. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Data.
Analysis Services	/ASLOGDIR **Optional**	Specifies the directory for Analysis Services log files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Log. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Log.
Analysis Services	/ASSYSADMINACCOUNTS **Required**	Specifies the administrator credentials for Analysis Services.

SQL SERVER COMPONENT	PARAMETER	DETAILS
Analysis Services	/ASTEMPDIR **Optional**	Specifies the directory for Analysis Services temporary files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Temp. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Temp.
Analysis Services	/ASPROVIDERMSOLAP **Optional**	Specifies whether the MSOLAP provider can run in-process. Default value: 1=enabled
Analysis Services	/ASSERVERMODE **Optional**	Specifies the server mode of the Analysis Services instance. Valid values in a cluster scenario are MULTIDIMENSIONAL or TABULAR. **ASSERVERMODE** is case-sensitive. All values must be expressed in upper case. For more information about the valid values, see Install Analysis Services in Tabular Mode.
SQL Server Database Engine	/INSTALLSQLDATADIR **Required**	Specifies the data directory for SQL Server data files. The data directory must to specified and on a shared cluster disk.
SQL Server Database Engine	/SAPWD **Required when /SECURITYMODE=SQL**	Specifies the password for the SQL Serversa account.
SQL Server Database Engine	/SECURITYMODE **Optional**	Specifies the security mode for SQL Server. If this parameter is not supplied, then Windows-only authentication mode is supported. Supported value: SQL
SQL Server Database Engine	/SQLBACKUPDIR **Optional**	Specifies the directory for backup files. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Backup.

SQL SERVER COMPONENT	PARAMETER	DETAILS
SQL Server Database Engine	/SQLCOLLATION **Optional**	Specifies the collation settings for SQL Server. The default value is based on the locale of your Windows operating system. For more information, see Collation Settings in Setup.
SQL Server Database Engine	/SQLSVCACCOUNT **Required**	Specifies the startup account for the SQL Server service.
SQL Server Database Engine	/SQLSVCPASSWORD Required	Specifies the password for SQLSVCACCOUNT.
SQL Server Database Engine	/SQLSYSADMINACCOUNTS **Required**	Use this parameter to provision logins to be members of the sysadmin role.
SQL Server Database Engine	/SQLTEMPDBDIR **Optional**	Specifies the directory for the data files for tempdb. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLTEMPDBLOGDIR **Optional**	Specifies the directory for the log files for tempdb. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLUSERDBDIR **Optional**	Specifies the directory for the data files for user databases. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLUSERDBLOGDIR **Optional**	Specifies the directory for the log files for user databases. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data

SQL SERVER COMPONENT	PARAMETER	DETAILS
FILESTREAM	/FILESTREAMLEVEL **Optional**	Specifies the access level for the FILESTREAM feature. Supported values: 0 =Disable FILESTREAM support for this instance. (Default value) 1=Enable FILESTREAM for Transact-SQL access. 2=Enable FILESTREAM for Transact-SQL and file I/O streaming access. (Not valid for cluster scenarios) 3=Allow remote clients to have streaming access to FILESTREAM data.
FILESTREAM	/FILESTREAMSHARENAME **Optional** **Requiredwhen FILESTREAMLEVEL is greater than 1.**	Specifies the name of the windows share in which the FILESTREAM data will be stored.
SQL Server Full Text	/FTSVCACCOUNT **Optional**	Specifies the account for Full-Text filter launcher service. This parameter is ignored in Windows Server 2008. ServiceSID will be used to help secure the communication between SQL Server and Full-text Filter Daemon. If the values are not provided the Full-text Filter Launcher Service will be disabled. You have to use SQL Server Control Manager to change the service account and enable full-text functionality. Default value: Local Service Account
SQL Server Full Text	/FTSVCPASSWORD **Optional**	Specifies the password for the Full-Text filter launcher service. This parameter is ignored in Windows Server 2008.
Integration Services	/ISSVCACCOUNT **Required**	Specifies the account for Integration Services. Default value: NT AUTHORITY\NETWORK SERVICE
Integration Services	/ISSVCPASSWORD Required	Specifies the Integration Services password.

SQL SERVER COMPONENT	PARAMETER	DETAILS
Integration Services	/ISSVCStartupType **Optional**	Specifies the startup mode for the Integration Services service.
Reporting Services	/RSINSTALLMODE **Optional**	Specifies the Install mode for Reporting Services.
Reporting Services	/RSSVCACCOUNT **Required**	Specifies the startup account for Reporting Services.
Reporting Services	/RSSVCPASSWORD Required	Specifies the password for the startup account for Reporting Services service.
Reporting Services	/RSSVCStartupType **Optional**	Specifies the startup mode for Reporting Services.

[1] We recommend that you use Service SID instead of domain groups.

Additional Notes:

The Database Engine and Analysis Services are the only components that are cluster-aware. Other features are not cluster-aware and do not have high availability through failover.

Sample Syntax:

To install a single-node SQL Server failover cluster instance with the Database Engine and Analysis Services, default instance.

```
setup.exe /q /ACTION=InstallFailoverCluster /InstanceName=MSSQLSERVER /INDICATEPROGRESS
/ASSYSADMINACCOUNTS="<DomainName\UserName>" /ASDATADIR=<Drive>:\OLAP\Data /ASLOGDIR=<Drive>:\OLAP\Log
/ASBACKUPDIR=<Drive>:\OLAP\Backup /ASCONFIGDIR=<Drive>:\OLAP\Config /ASTEMPDIR=<Drive>:\OLAP\Temp
/FAILOVERCLUSTERDISKS="<Cluster Disk Resource Name - for example, 'Disk S:'" /FAILOVERCLUSTERNETWORKNAME="
<Insert Network Name>" /FAILOVERCLUSTERIPADDRESSES="IPv4;xx.xxx.xx.xx;Cluster Network;xxx.xxx.xxx.x"
/FAILOVERCLUSTERGROUP="MSSQLSERVER" /Features=AS,SQL /ASSVCACCOUNT="<DomainName\UserName>"
/ASSVCPASSWORD="xxxxxxxxxxx" /AGTSVCACCOUNT="<DomainName\UserName>" /AGTSVCPASSWORD="xxxxxxxxxxx"
/INSTALLSQLDATADIR="<Drive>:\<Path>\MSSQLSERVER" /SQLCOLLATION="SQL_Latin1_General_CP1_CS_AS"
/SQLSVCACCOUNT="<DomainName\UserName>" /SQLSVCPASSWORD="xxxxxxxxxxx" /SQLSYSADMINACCOUNTS="
<DomainName\UserName> /IACCEPTSQLSERVERLICENSETERMS
```

Prepare Failover Cluster Parameters

Use the parameters in the following table to develop command-line scripts for failover cluster prepare. This is the first step in advanced cluster installation, where you have to prepare the failover cluster instances on all the nodes of the failover cluster. For more information, see AlwaysOn Failover Cluster Instances (SQL Server).

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the failover cluster prepare work flow. Supported values: PrepareFailoverCluster

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/IACCEPTSQLSERVERLICENSETERMS **Required only when the /Q or /QS parameter is specified for unattended installations.**	Required to acknowledge acceptance of the license terms.
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/*UpdateEnabled* **Optional**	Specify whether SQL Server setup should discover and include product updates. The valid values are True and False or 1 and 0. By default, SQL Server setup will include updates that are found.
SQL Server Setup Control	/*UpdateSource* **Optional**	Specify the location where SQL Server setup will obtain product updates. The valid values are "MU" to search Microsoft Update, a valid folder path, a relative path such as .\MyUpdates or a UNC share. By default, SQL Server setup will search Microsoft Update or a Windows Update Service through the Windows Server Update Services.
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.
SQL Server Setup Control	/ERRORREPORTING **Optional**	Specifies the error reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/FEATURES **Required**	Specifies components to install.
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for the parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file will be piped to the console.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/INSTALLSHAREDDIR **Optional**	Specifies a nondefault installation directory for 64-bit shared components. Default is %Program Files%\MicrosoftSQL Server Cannot be set to %Program Files(x86)%\MicrosoftSQL Server
SQL Server Setup Control	/INSTALLSHAREDWOWDIR **Optional**	Specifies a nondefault installation directory for 32-bit shared components. Supported only on a 64-bit system. Default is %Program Files(x86)%\MicrosoftSQL Server Cannot be set to %Program Files%\MicrosoftSQL Server
SQL Server Setup Control	/INSTANCEDIR **Optional**	Specifies nondefault installation directory for instance specific components.
SQL Server Setup Control	/INSTANCEID **Optional**	Specifies a nondefault value for an InstanceID.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/PID **Optional**	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/QS **Optional**	Specifies that Setup runs and shows progress through the UI, but does not accept any input or show any error messages.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/SQMREPORTING **Optional**	Specifies feature usage reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.
SQL Server Agent	/AGTSVCACCOUNT **Required**	Specifies the account for the SQL Server Agent service.
SQL Server Agent	/AGTSVCPASSWORD Required	Specifies the password for SQL Server Agent service account.
Analysis Services	/ASSVCACCOUNT **Required**	Specifies the account for the Analysis Services service.
Analysis Services	/ASSVCPASSWORD Required	Specifies the password for the Analysis Services service.
SQL Server Database Engine	/SQLSVCACCOUNT **Required**	Specifies the startup account for the SQL Server service.
SQL Server Database Engine	/SQLSVCPASSWORD Required	Specifies the password for SQLSVCACCOUNT.
FILESTREAM	/FILESTREAMLEVEL **Optional**	Specifies the access level for the FILESTREAM feature. Supported values: 0 =Disable FILESTREAM support for this instance. (Default value) 1=Enable FILESTREAM for Transact-SQL access. 2=Enable FILESTREAM for Transact-SQL and file I/O streaming access. (Not valid for Cluster scenarios) 3=Allow remote clients to have streaming access to FILESTREAM data.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
FILESTREAM	/FILESTREAMSHARENAME **Optional** **Required** when FILESTREAMLEVEL is greater than 1.	Specifies the name of the windows share in which the FILESTREAM data will be stored.
SQL Server Full Text	/FTSVCACCOUNT **Optional**	Specifies the account for Full-Text filter launcher service. This parameter is ignored in Windows Server 2008. ServiceSID will be used to help secure the communication between SQL Server and Full-text Filter Daemon. If the values are not provided the Full-text Filter Launcher Service will be disabled. You have to use SQL Server Control Manager to change the service account and enable full-text functionality. Default value: Local Service Account
SQL Server Full Text	/FTSVCPASSWORD **Optional**	Specifies the password for the Full-Text filter launcher service. This parameter is ignored in Windows Server 2008.
Integration Services	/ISSVCACCOUNT **Required**	Specifies the account for Integration Services. Default value: NT AUTHORITY\NETWORK SERVICE
Integration Services	/ISSVCPASSWORD Required	Specifies the Integration Services password.
Integration Services	/ISSVCStartupType **Optional**	Specifies the startup mode for the Integration Services service.
Reporting Services	/RSINSTALLMODE **Available only on Files only mode.**	Specifies the Install mode for Reporting Services.
Reporting Services	/RSSVCACCOUNT **Required**	Specifies the startup account for Reporting Services.
Reporting Services	/RSSVCPASSWORD Required	Specifies the password for the startup account for Reporting Services service.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Reporting Services	/RSSVCStartupType **Optional**	Specifies the startup mode for Reporting Services.

[1] We recommend that you use Service SID instead of domain groups.

Sample Syntax:

To perform the "Preparation" step of a failover cluster advanced installation scenario for the Database Engine and Analysis Services.

Run the following command at the command prompt to prepare a default instance:

```
setup.exe /q /ACTION=PrepareFailoverCluster /InstanceName=MSSQLSERVER /Features=AS,SQL /INDICATEPROGRESS
/ASSVCACCOUNT="<DomainName\UserName>" /ASSVCPASSWORD="xxxxxxxxxxx" /SQLSVCACCOUNT="<DomainName\UserName>"
/SQLSVCPASSWORD="xxxxxxxxxxx" /AGTSVCACCOUNT="<DomainName\UserName>" /AGTSVCPASSWORD="xxxxxxxxxxx"
/IACCEPTSQLSERVERLICENSETERMS
```

Run the following command at the command prompt to prepare a named instance:

```
setup.exe /q /ACTION=PrepareFailoverCluster /InstanceName="<Insert Instance name>" /Features=AS,SQL
/INDICATEPROGRESS /ASSVCACCOUNT="<DomainName\UserName>" /ASSVCPASSWORD="xxxxxxxxxxx" /SQLSVCACCOUNT="
<DomainName\UserName>" /SQLSVCPASSWORD="xxxxxxxxxxx" /AGTSVCACCOUNT="<DomainName\UserName>"
/AGTSVCPASSWORD="xxxxxxxxxxx" /IACCEPTSQLSERVERLICENSETERMS
```

Complete Failover Cluster Parameters

Use the parameters in the following table to develop command-line scripts for failover cluster complete. This is the second step in the advanced failover cluster install option. After you have run prepare on all the failover cluster nodes, you run this command on the node that owns the shared disks. For more information, see AlwaysOn Failover Cluster Instances (SQL Server).

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the failover cluster complete work flow. Supported values: CompleteFailoverCluster
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/FAILOVERCLUSTERGROUP **Optional**	Specifies the name of the resource group to be used for the SQL Server failover cluster. It can be the name of an existing cluster group or the name of a new resource group. Default value: SQL Server (<InstanceName>)

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.
SQL Server Setup Control	/ERRORREPORTING **Optional**	Specifies the error reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for the parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file will be piped to the console.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/PID **Optional**	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/QS **Optional**	Specifies that Setup runs and shows progress through the UI, but does not accept any input or show any error messages.
SQL Server Setup Control	/SQMREPORTING **Optional**	Specifies feature usage reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/FAILOVERCLUSTERDISKS **Optional**	Specifies the list of shared disks to be included in the SQL Server failover cluster resource group. Default value: The first drive is used as the default drive for all databases.
SQL Server Setup Control	/FAILOVERCLUSTERIPADDRESSES **Required**	Specifies an encoded IP address. The encodings are semicolon-delimited (;) and follow the format <IP Type>;<address>;<network name>;<subnet mask>. Supported IP types include DHCP, IPv4, and IPv6. You can specify multiple failover cluster IP addresses with a space in between. See the following examples: FAILOVERCLUSTERIPADDRESSES=DEFAULT FAILOVERCLUSTERIPADDRESSES=IPv4;DHCP;ClusterNetwork1 FAILOVERCLUSTERIPADDRESSES=IPv4;172.16.0.0;ClusterNetwork1;172.31.255.255 FAILOVERCLUSTERIPADDRESSES=IPv6;DHCP;ClusterNetwork1 FAILOVERCLUSTERIPADDRESSES=IPv6;2001:db8:23:1002:20f:1fff:feff:b3a3;ClusterNetwork1
SQL Server Setup Control	/FAILOVERCLUSTERNETWORKNAME **Required**	Specifies the network name for the new SQL Server failover cluster. This name is used to identify the new SQL Server failover cluster instance on the network.
SQL Server Setup Control	/CONFIRMIPDEPENDENCYCHANGE	Indicates the consent to set the IP address resource dependency to OR for multi-subnet failover clusters. For more information, see Create a New SQL Server Failover Cluster (Setup). Supported Values: 0 = False (Default) 1 = True

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Analysis Services	/ASBACKUPDIR **Optional**	Specifies the directory for Analysis Services backup files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Backup. For all other installations:%Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Backup.
Analysis Services	/ASCOLLATION **Optional**	Specifies the collation setting for Analysis Services. Default value: Latin1_General_CI_AS
Analysis Services	/ASCONFIGDIR **Optional**	Specifies the directory for Analysis Services configuration files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Config. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Config.
Analysis Services	/ASDATADIR **Optional**	Specifies the directory for Analysis Services data files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Data. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Data.
Analysis Services	/ASLOGDIR **Optional**	Specifies the directory for Analysis Services log files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Log. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Log.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Analysis Services	/ASSERVERMODE **Optional**	Specifies the server mode of the Analysis Services instance. Valid values in a cluster scenario are MULTIDIMENSIONAL or TABULAR. **ASSERVERMODE** is case-sensitive. All values must be expressed in upper case. For more information about the valid values, see Install Analysis Services in Tabular Mode.
Analysis Services	/ASSYSADMINACCOUNTS **Required**	Specifies the administrator credentials for Analysis Services.
Analysis Services	/ASTEMPDIR **Optional**	Specifies the directory for Analysis Services temporary files. Default values: For WOW mode on 64-bit: %Program Files(x86)%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Temp. For all other installations: %Program Files%\MicrosoftSQL Server\ <INSTANCEDIR>\ <ASInstanceID>\OLAP\Temp.
Analysis Services	/ASPROVIDERMSOLAP **Optional**	Specifies whether the MSOLAP provider can run in-process. Default value: 1=enabled
SQL Server Database Engine	/INSTALLSQLDATADIR **Required**	Specifies the data directory for SQL Server data files. The data directory must to specified and on a shared cluster disk.
SQL Server Database Engine	/SAPWD **Required when /SECURITYMODE=SQL**	Specifies the password for the SQL Serversa account.
SQL Server Database Engine	/SECURITYMODE **Optional**	Specifies the security mode for SQL Server. If this parameter is not supplied, then Windows-only authentication mode is supported. Supported values: SQL

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Database Engine	/SQLBACKUPDIR **Optional**	Specifies the directory for backup files. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Backup.
SQL Server Database Engine	/SQLCOLLATION **Optional**	Specifies the collation settings for SQL Server. The default value is based on the locale of your Windows operating system. For more information, see Collation Settings in Setup.
SQL Server Database Engine	/SQLSYSADMINACCOUNTS **Required**	Use this parameter to provision logins to be members of the sysadmin role.
SQL Server Database Engine	/SQLTEMPDBDIR **Optional**	Specifies the directory for the data files for tempdb. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data.
SQL Server Database Engine	/SQLTEMPDBLOGDIR **Optional**	Specifies the directory for the l files for tempdb. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLUSERDBDIR **Optional**	Specifies the directory for the data files for user databases. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
SQL Server Database Engine	/SQLUSERDBLOGDIR **Optional**	Specifies the directory for the log files for user databases. Default value: <InstallSQLDataDir>\ <SQLInstanceID>\MSSQL\Data
Reporting Services	/RSINSTALLMODE **Available on files only mode.**	Specifies the Install mode for Reporting Services.

Sample Syntax:

To perform the "Completion" step of a failover cluster advanced installation scenario for the Database Engine and Analysis Services. Run the following command on the computer that will be the active node in the failover cluster to make it usable. You must run the "CompleteFailoverCluster" action on the node that owns the shared disk in the Analysis Services failover cluster.

Run the following command at the command prompt to complete failover cluster installation for a default instance:

```
setup.exe /q /ACTION=CompleteFailoverCluster /InstanceName=MSSQLSERVER /INDICATEPROGRESS
/ASSYSADMINACCOUNTS="<DomainName\Username>" /ASDATADIR=<Drive>:\OLAP\Data /ASLOGDIR=<Drive>:\OLAP\Log
/ASBACKUPDIR=<Drive>:\OLAP\Backup /ASCONFIGDIR=<Drive>:\OLAP\Config /ASTEMPDIR=<Drive>:\OLAP\Temp
/FAILOVERCLUSTERDISKS="<Cluster Disk Resource Name - for example, 'Disk S:'>:"
/FAILOVERCLUSTERNETWORKNAME="<Insert FOI Network Name>"
/FAILOVERCLUSTERIPADDRESSES="IPv4;xx.xxx.xx.xx;Cluster Network;xxx.xxx.xxx.x"
/FAILOVERCLUSTERGROUP="MSSQLSERVER" /INSTALLSQLDATADIR="<Drive>:\<Path>\MSSQLSERVER"
/SQLCOLLATION="SQL_Latin1_General_CP1_CS_AS" /SQLSYSADMINACCOUNTS="<DomainName\UserName>"
```

Run the following command at the command prompt to complete failover cluster installation for a named instance:

```
setup.exe /q /ACTION=CompleteFailoverCluster /InstanceName="<Insert Instance Name>" /INDICATEPROGRESS
/ASSYSADMINACCOUNTS="<DomainName\UserName>" /ASDATADIR=<Drive>:\KATMAI\Data /ASLOGDIR=<drive>:\KATMAI\Log
/ASBACKUPDIR=<Drive>:\KATMAI\Backup /ASCONFIGDIR=<Drive>:\KATMAI\Config /ASTEMPDIR=<Drive>:\KATMAI\Temp
/FAILOVERCLUSTERDISKS="<Cluster Disk Resource Name - for example, 'Disk S:'>"
/FAILOVERCLUSTERNETWORKNAME="CompNamedFOI"
/FAILOVERCLUSTERIPADDRESSES="IPv4;xx.xxx.xx.xx;ClusterNetwork1;xxx.xxx.xxx.x" /FAILOVERCLUSTERGROUP="
<Insert New Group Name>" /INSTALLSQLDATADIR="<Drive>:\<Path>\MSSQLSERVER_KATMAI"
/SQLCOLLATION="SQL_Latin1_General_CP1_CS_AS" /SQLSYSADMINACCOUNTS="<DomainName\Username>"
```

Upgrade Failover Cluster Parameters

Use the parameters in the following table to develop command-line scripts for failover cluster upgrade. For more information, see Upgrade a SQL Server Failover Cluster Instance (Setup) and AlwaysOn Failover Cluster Instances (SQL Server).

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate the installation workflow. Supported values: Upgrade
SQL Server Setup Control	/IACCEPTSQLSERVERLICENSETERMS **Required only when the /Q or /QS parameter is specified for unattended installations.**	Required to acknowledge acceptance of the license terms.
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	/*UpdateEnabled* **Optional**	Specify whether SQL Server setup should discover and include product updates. The valid values are True and False or 1 and 0. By default, SQL Server setup will include updates that are found.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	*/UpdateSource* **Optional**	Specify the location where SQL Server setup will obtain product updates. The valid values are "MU" to search Microsoft Update, a valid folder path, a relative path such as .\MyUpdates or a UNC share. By default, SQL Server setup will search Microsoft Update or a Windows Update Service through the Windows Server Update Services.
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.
SQL Server Setup Control	/ERRORREPORTING **Optional**	Specifies the error reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 1=enabled 0=disabled
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for the parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file will be piped to the console.
SQL Server Setup Control	/ INSTANCEDIR **Optional**	Specifies a nondefault installation directory for shared components.
SQL Server Setup Control	/INSTANCEID **Required when you upgrade from SQL Server 2008 or higher.** **Optional when you upgrade from SQL Server 2005**	Specifies a nondefault value for an InstanceID.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/PID **Optional**	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/SQMREPORTING **Optional**	Specifies feature usage reporting for SQL Server. For more information, see Privacy Statement for the Microsoft Error Reporting Service. Supported values: 0=disabled 1=enabled
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.
SQL Server Setup Control	/FAILOVERCLUSTERROLLOWNERSHIP	Specifies the failover behavior during upgrade.
SQL Server Browser Service	/BROWSERSVCSTARTUPTYPE **Optional**	Specifies the startup mode for SQL Server Browser service. Supported values: Automatic Disabled Manual
SQL Server Full-Text	/FTUPGRADEOPTION **Optional**	Specifies the Full-Text catalog upgrade option. Supported values: REBUILD RESET IMPORT
Integration Services	/ISSVCACCOUNT **Required**	Specifies the account for Integration Services. Default value: NT AUTHORITY\NETWORK SERVICE
Integration Services	/ISSVCPASSWORD Required	Specifies the Integration Services password.
Integration Services	/ISSVCStartupType **Optional**	Specifies the startup mode for the Integration Services service.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
Reporting Services	/RSUPGRADEDATABASEACCOUNT **Optional**	The property is only used when upgrading a SharePoint mode Report Server that is version 2008 R2 or earlier. Additional upgrade operations are performed for report servers that use the older SharePoint mode architecture, which was changed in SQL Server 2012 Reporting Services. If this option is not included with the command line installation, the default service account for the old report server instance is used. If this property is used, supply the password for the account using the **/RSUPGRADEPASSWORD** property.
Reporting Services	/RSUPGRADEPASSWORD **Optional**	Password of the existing Report Server service account.

Add Node Parameters

Use the parameters in the following table to develop command-line scripts for AddNode.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate AddNode work flow. Supported value: AddNode
SQL Server Setup Control	/IACCEPTSQLSERVERLICENSETERMS **Required only when the /Q or /QS parameter is specified for unattended installations.**	Required to acknowledge acceptance of the license terms.
SQL Server Setup Control	/ENU **Optional**	Use this parameter to install the English version of SQL Server on a localized operating system when the installation media includes language packs for both English and the language corresponding to the operating system.
SQL Server Setup Control	*/UpdateEnabled* **Optional**	Specify whether SQL Server setup should discover and include product updates. The valid values are True and False or 1 and 0. By default, SQL Server setup will include updates that are found.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	*/UpdateSource* **Optional**	Specify the location where SQL Server setup will obtain product updates. The valid values are "MU" to search Microsoft Update, a valid folder path, a relative path such as .\MyUpdates or a UNC share. By default, SQL Server setup will search Microsoft Update or a Windows Update Service through the Windows Server Update Services.
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for the parameters.
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file will be piped to the console.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/PID **Optional**	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/QS **Optional**	Specifies that Setup runs and shows progress through the UI, but does not accept any input or show any error messages.
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/FAILOVERCLUSTERIPADDRESSES **Required**	Specifies an encoded IP address. The encodings are semicolon-delimited (;) and follow the format <IP Type>;<address>;<network name>;<subnet mask>. Supported IP types include DHCP, IPv4, and IPv6. You can specify multiple failover cluster IP addresses with a space in between. See the following examples: FAILOVERCLUSTERIPADDRESSES=DEFAULT FAILOVERCLUSTERIPADDRESSES=IPv4;DHCP;ClusterNetwork1 FAILOVERCLUSTERIPADDRESSES=IPv4;172.16.0.0;ClusterNetwork1;172.31.255.255 FAILOVERCLUSTERIPADDRESSES=IPv6;DHCP;ClusterNetwork1 FAILOVERCLUSTERIPADDRESSES=IPv6;2001:db8:23:1002:20f:1fff:feff:b3a3;ClusterNetwork1 For more information, see Add or Remove Nodes in a SQL Server Failover Cluster (Setup).
SQL Server Setup Control	/CONFIRMIPDEPENDENCYCHANGE **Required**	Indicates the consent to set the IP address resource dependency to OR for multi-subnet failover clusters. For more information, see Add or Remove Nodes in a SQL Server Failover Cluster (Setup). Supported values: 0 = False (Default) 1 = True
SQL Server Agent	/AGTSVCACCOUNT **Required**	Specifies the account for the SQL Server Agent service.
SQL Server Agent	/AGTSVCPASSWORD Required	Specifies the password for SQL Server Agent service account.
Analysis Services	/ASSVCACCOUNT **Required**	Specifies the account for the Analysis Services service.
Analysis Services	/ASSVCPASSWORD Required	Specifies the password for the Analysis Services service.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Database Engine	/SQLSVCACCOUNT **Required**	Specifies the startup account for the SQL Server service.
SQL Server Database Engine	/SQLSVCPASSWORD Required	Specifies the password for SQLSVCACCOUNT.
Integration Services	/ISSVCPASSWORD Required	Specifies the Integration Services password.
Reporting Services	/RSINSTALLMODE **Available in Files only mode**	Specifies the Install mode for Reporting Services.
Reporting Services	/RSSVCPASSWORD Required	Specifies the startup account password for the Reporting Services service.

Additional Notes:

The Database Engine and Analysis Services are the only components that are cluster-aware. Other features are not cluster-aware and do not have high availability through failover.

Sample Syntax:

To add a node to an existing failover cluster instance with the Database Engine and Analysis Services.

```
setup.exe /q /ACTION=AddNode /INSTANCENAME="<Insert Instance Name>" /SQLSVCACCOUNT="<SQL account that is
used on other nodes>" /SQLSVCPASSWORD="<password for SQL account>" /AGTSVCACCOUNT="<SQL Server Agent
account that is used on other nodes>", /AGTSVCPASSWORD="<SQL Server Agent account password>"
/ASSVCACCOUNT="<AS account that is used on other nodes>" /ASSVCPASSWORD="<password for AS account>"
/INDICATEPROGRESS /IACCEPTSQLSERVERLICENSETERMS
/FAILOVERCLUSTERIPADDRESSES="IPv4;xx.xxx.xx.xx;ClusterNetwork1;xxx.xxx.xxx.x" /CONFIRMIPDEPENDENCYCHANGE=0
```

Remove Node Parameters

Use the parameters in the following table to develop command-line scripts for RemoveNode. To uninstall a failover cluster, you must run RemoveNode on each failover cluster node. For more information, see AlwaysOn Failover Cluster Instances (SQL Server).

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/ACTION **Required**	Required to indicate RemoveNode work flow. Supported value: RemoveNode
SQL Server Setup Control	/CONFIGURATIONFILE **Optional**	Specifies the ConfigurationFile to use.
SQL Server Setup Control	/HELP, H, ? **Optional**	Displays the usage options for the parameters.

SQL SERVER COMPONENT	PARAMETER	DESCRIPTION
SQL Server Setup Control	/INDICATEPROGRESS **Optional**	Specifies that the verbose Setup log file will be piped to the console.
SQL Server Setup Control	/INSTANCENAME **Required**	Specifies a SQL Server instance name. For more information, see Instance Configuration.
SQL Server Setup Control	/Q **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
SQL Server Setup Control	/QS **Optional**	Specifies that Setup runs and shows progress through the UI, but does not accept any input or show any error messages.
SQL Server Setup Control	/HIDECONSOLE **Optional**	Specifies that the console window is hidden or closed.
SQL Server Setup Control	/CONFIRMIPDEPENDENCYCHANGE **Required**	Indicates the consent to set the IP address resource dependency from OR to AND for multi-subnet failover clusters. For more information, see Add or Remove Nodes in a SQL Server Failover Cluster (Setup). Supported values: 0 = False (Default) 1 = True

Sample Syntax:

To remove a node from an existing failover cluster instance with the Database Engine and Analysis Services.

```
setup.exe /q /ACTION=RemoveNode /INSTANCENAME="<Insert Instance Name>" [/INDICATEPROGRESS]
/CONFIRMIPDEPENDENCYCHANGE=0
```

Service Account Parameters

You can configure the SQL Server services by using a built-in account, local account, or domain account.

> **NOTE**
> When you use a managed service account, virtual account, or a built-in account, you should not specify the corresponding password parameters. For more information about these service accounts, see **New Account Types Available with Windows 7 and Windows Server 2008 R2** section in Configure Windows Service Accounts and Permissions.

For more information about service account configuration, see Configure Windows Service Accounts and Permissions.

SQL SERVER COMPONENT	ACCOUNT PARAMETER	PASSWORD PARAMETER	STARTUP TYPE
SQL Server Agent	/AGTSVCACCOUNT	/AGTSVCPASSWORD	/AGTSVCSTARTUPTYPE
Analysis Services	/ASSVCACCOUNT	/ASSVCPASSWORD	/ASSVCSTARTUPTYPE
SQL Server Database Engine	/SQLSVCACCOUNT	/SQLSVCPASSWORD	/SQLSVCSTARTUPTYPE
Integration Services	/ISSVCACCOUNT	/ISSVCPASSWORD	/ISSVCSTARTUPTYPE
Reporting Services	/RSSVCACCOUNT	/RSSVCPASSWORD	/RSSVCSTARTUPTYPE

Feature Parameters

To install specific features, use the /FEATURES parameter and specify the parent feature or feature values in the following table. For a list of features that are supported by the editions of SQL Server, see Features Supported by the Editions of SQL Server 2014.

Parent feature parameter	Feature parameter	Description
SQL		Installs the SQL Server Database Engine, Replication, Fulltext, and Data Quality Server.
	SQLEngine	Installs just the SQL Server Database Engine.
	Replication	Installs the Replication component along with SQL Server Database Engine.
	FullText	Installs the FullText component along with SQL Server Database Engine.
	DQ	Copies the files required for completing the Data Quality Server installation. After completing SQL Server installation, you must run the DQSInstaller.exe file to complete the Data Quality Server installation. For more information, see Run DQSInstaller.exe to Complete Data Quality Server Installation. This also installs SQL Server Database Engine.
AS		Installs all Analysis Services components.
RS		Installs all Reporting Services components.
DQC		Installs Data Quality Client.

IS		Installs all Integration Services components.
MDS		Installs Master Data Services.
Tools		Installs client tools and SQL Server Books Online components.
	BC	Installs backward compatibility components.
	BOL	Installs SQL Server Books Online components to view and manage help content.
	Conn	Installs connectivity components.
	SSMS	Installs SQL Server Management Tools - Basic. This includes the following: SQL Server Management Studio support for the SQL Server Database Engine, SQL Server Express, **sqlcmd** utility, and the SQL Server PowerShell provider
	ADV_SSMS	Installs SQL Server Management Tools - Complete. This includes the following components in addition to the components in the Basic version: SQL Server Management Studio support for Reporting Services, Analysis Services, and Integration Services SQL Server Profiler Database Engine Tuning Advisor SQL Server Utility management
	DREPLAY_CTLR	Installs Distributed Replay controller
	DREPLAY_CLT	Installs Distributed Replay client
	SNAC_SDK	Installs SDK for MicrosoftSQL Server Native Client
	SDK	Installs the software development kit.
	LocalDB[1]	Installs LocalDB, an execution mode of SQL Server Express targeted to program developers.

[1] LocalDB is an option when installing any SKU of SQL Server 2014 Express. For more information, see SQL Server 2014 Express LocalDB.

Feature parameter examples:

PARAMETER AND VALUES	DESCRIPTION
/FEATURES=SQLEngine	Installs the Database Engine without replication and full-text.
/FEATURES=SQLEngine, FullText	Installs the Database Engine and full-text.
/FEATURES=SQL, Tools	Installs the complete Database Engine and all tools.
/FEATURES=BOL	Installs SQL Server Books Online components to view and manage help content.

Role Parameters

The setup role or /Role parameter is used to install a preconfigured selection of features. The Analysis Services roles install an Analysis Services instance in either an existing SharePoint farm, or a new un-configured farm. Two setup roles are provided to support each scenario. You can only choose one setup role to install at a time. If you choose a setup role, Setup installs the features and components that belong to the role. You cannot vary the features and components that are designated for that role. For more information about how to use the feature role parameter, see Install PowerPivot from the Command Prompt.

The AllFeatures_WithDefaults role is the default behavior for editions of SQL Server Express and reduces the number of dialog boxes presented to the user. It can be specified from the command line when installing a SQL Server edition that is not SQL Server Express.

ROLE	DESCRIPTION	INSTALLS...
SPI_AS_ExistingFarm	Installs Analysis Services as a PowerPivot named instance on an existing SharePoint Server 2010 farm or standalone server.	Analysis Services calculation engine, preconfigured for in-memory data storage and processing. PowerPivot solution packages Installer program for the PowerPivot for Excel SQL Server Books Online
SPI_AS_NewFarm	Installs Analysis Services and Database Engine as a PowerPivot named instance on a new, un-configured Office SharePoint Server 2010 farm or standalone server. SQL Server Setup will configure the farm during feature role installation.	Analysis Services calculation engine, preconfigured for in-memory data storage and processing. PowerPivot solution packages SQL Server Books Online Database Engine Configuration Tools SQL Server Management Studio

ROLE	DESCRIPTION	INSTALLS...
AllFeatures_WithDefaults	Installs all features that are available with the current edition. Adds the current user to the SQL Server `sysadmin` fixed server role. On Windows Server 2008 or higher and when the operating system is not a domain controller, the Database Engine, and Reporting Services are defaulted to use the NTAUTHORITY\NETWORK SERVICE account, and Integration Services is defaulted to use the NTAUTHORITY\NETWORK SERVICE account. This role is enabled by default in editions of SQL Server Express. For all other editions, this role is not enabled but can be specified through the UI or with command line parameters.	For editions of SQL Server Express, installs only those features available in the edition. For other editions, installs all SQL Server features. The **AllFeatures_WithDefaults** parameter can be combined with other parameters which override the **AllFeatures_WithDefaults** parameter settings. For example, using the **AllFeatures_WithDefaults** parameter and the **/Features=RS** parameter overrides the command to install all features and only installs Reporting Services, but honors the **AllFeatures_WithDefaults** parameter to use the default service account for Reporting Services. When using the **AllFeatures_WithDefaults** parameter along with the **/ADDCURRENTUSERASSQLADMIN=FALSE** the provisioning dialog is not auto populated with the current user. Add **/AGTSVCACCOUNT** and **/AGTSVCPASSWORD** to specify a service account and password for the SQL Server Agent.

Controlling Failover Behavior using the /FAILOVERCLUSTERROLLOWNERSHIP Parameter

To upgrade a SQL Server failover cluster to SQL Server 2014, you must run the Setup on one failover cluster node at a time, starting with the passive nodes. Setup determines when to fail over to the upgraded node, depending on the total number of nodes in the failover cluster instance, and the number of nodes that have already been upgraded. When half of the nodes or more have already been upgraded, Setup by default will cause a failover to an upgraded node.

To control the failover behavior of cluster nodes during the upgrade process, run the upgrade operation at the command prompt and use the /FAILOVERCLUSTERROLLOWNERSHIP parameter to control the failover behavior before the upgrade operation takes the node offline. Use of this parameter is as follows:

- /FAILOVERCLUSTERROLLOWNERSHIP=0 will not roll cluster ownership (move group) to upgraded nodes, and does 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 (move group) 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 (move group) as needed.

Instance ID or InstanceID Configuration

The Instance ID or /InstanceID parameter is used for specifying where you can install the instance components and the registry path of the instance. The value of "INSTANCEID" is a string and should be unique.

- SQL Instance ID:MSSQL12.<INSTANCEID>
- AS Instance ID:MSAS12.<INSTANCEID>
- RS Instance ID:MSRS12.<INSTANCEID>

The instance-aware components are installed to the following locations:

%Program Files%\MicrosoftSQL Server\<SQLInstanceID>

%Program Files%\MicrosoftSQL Server\<ASInstanceID>

%Program Files%\MicrosoftSQL Server\<RSInstanceID>

> **NOTE**
> If INSTANCEID is not specified on the command line, then by default Setup substitute <INSTANCEID> with the <INSTANCENAME>.

See Also

Install SQL Server 2014 from the Installation Wizard (Setup)
SQL Server Failover Cluster Installation
Install SQL Server 2014 BI Features

Install SQL Server 2014 Using a Configuration File

11/5/2018 • 5 minutes to read • Edit Online

SQL Server Setup provides the ability to generate a configuration file based upon the system default and run-time inputs. You can use the configuration file to deploy SQL Server throughout the enterprise with the same configuration. You can also standardize manual installations throughout the enterprise, by creating a batch file that launches Setup.exe.

Setup supports the use of the configuration file only through the command prompt. The processing order of the parameters while using the configuration file is outlined below:

- The configuration file overwrites the defaults in a package
- Command-line values overwrite the values in the configuration file

The configuration file can be used to track the parameters and values for each installation. This makes the configuration file useful for verifying and auditing the installations.

Configuration File Structure

The ConfigurationFile.ini file is a text file with parameters (name/value pair) and descriptive comments.

The following is an example of a ConfigurationFile.ini file:

```
; Microsoft SQL Server Configuration file
[OPTIONS]
; Specifies a Setup work flow, like INSTALL, UNINSTALL, or UPGRADE.
; This is a required parameter.
ACTION="Install"
; Specifies features to install, uninstall, or upgrade.
; The list of top-level features include SQL, AS, RS, IS, and Tools.
; The SQL feature will install the database engine, replication, and full-text.
; The Tools feature will install Management Tools, Books online,
; SQL Server Data Tools, and other shared components.
FEATURES=SQL,Tools
```

How to generate a configuration file

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.

   > **NOTE**
   >
   > SQL Server Express Edition setup does not create a configuration file automatically. The following command will start setup and create a configuration file.
   >
   > SETUP.exe /UIMODE=Normal /ACTION=INSTALL

2. Follow the wizard through to the **Ready to Install** page. The path to the configuration file is specified in the **Ready to Install** page in the configuration file path section. For more information about how to install SQL Server, see Install SQL Server 2014 from the Installation Wizard (Setup).

3. Cancel the setup without actually completing the installation, to generate the INI file.

> **NOTE**
>
> The setup infrastructure writes out all the appropriate parameters for the actions that were run, with the exception of sensitive information such as passwords. The /IAcceptSQLServerLicenseTerms parameter is also not written out to the configuration file and requires either a modification of the configuration file or a value to be supplied at the command prompt. For more information, see Install SQL Server 2014 from the Command Prompt. In addition, a value is included for Boolean parameters where a value is usually not supplied through the command prompt.

Using the Configuration File to Install SQL Server

You can only use the configuration file on command-line installations.

> **NOTE**
>
> If you need to make changes to the configuration file, we recommend that you make a copy and work with the copy.

How to use a configuration file to install a stand-alone SQL Server instance

- Run the installation through the command prompt and supply the ConfigurationFile.ini using the *ConfigurationFile* parameter.

How to use a configuration file to prepare and complete an image of a stand-alone SQL Server instance (SysPrep)

1. To prepare one or more instances of SQL Server and configure them on the same machine.
   - Run **Image preparation of a stand-alone instance of SQL Server** from the **Advanced** page of the Installation Center and capture the prepare image configuration file.
   - Use the same prepare image configuration file as a template to prepare more instances of SQL Server.
   - Run **Image completion of a prepared stand-alone instance of SQL Server** from the **Advanced** page of the Installation Center to configure a prepared instances on the machine.
2. To prepare an image of the operating system including an unconfigured prepared instance of SQL Server, using Windows SysPrep tool.
   - Run the **Image preparation of a stand-alone instance of SQL Server** from the Advanced page of the Installation Center and capture the prepare image configuration file.
   - Run the **Image completion of a prepared stand-alone instance of SQL Server** from the **Advanced** page of the Installation Center, but cancel it on the **Ready to Complete** page after capturing the complete configuration file.
   - The complete image configuration file can be stored with the Windows image for automating the configuration of the prepared instances.

How to install a SQL Server failover cluster using the configuration file

1. Integrated Install option (create a single node failover cluster on a node and for additional nodes, run AddNode on them):
   - Run the "Install a Failover Cluster" option and capture the configuration file that lists all the installation settings.
   - Run the command-line failover cluster install by supplying the *ConfigurationFile* parameter.
   - On an additional node to be added, run AddNode to capture the ConfigurationFile.ini file applicable to the existing failover cluster.

- Run the command-line AddNode on all the additional nodes that will join the failover cluster, by supplying the same configuration file using the ConfigurationFile parameter.

2. Advanced install option (prepare failover cluster on all failover cluster nodes, then, after preparing all the nodes, run complete on the node that owns the shared disk):
   - Run **Prepare** on one of the nodes, and capture the ConfigurationFile.ini file.
   - Supply the same ConfigurationFile.ini file to Setup on all the nodes that will be prepared for the failover cluster.
   - After all the nodes have been prepared, run a complete failover cluster operation on the node that owns the shared disk and capture the ConfigurationFile.ini file.
   - You can then supply this ConfigurationFile.ini file to complete the failover cluster.

How to add or remove a node to a SQL Server failover cluster using the configuration file

- If you have a configuration file that was previously used to add a node to or remove a node from a failover cluster, you can reuse that same file to add or remove additional nodes.

How to upgrade a SQL Server failover cluster using the configuration file

1. Run upgrade on the passive node and capture the ConfigurationFile.ini file. You can do this either by performing the actual upgrade, or exiting at the end without doing the actual upgrade.
2. On all the additional nodes to be upgraded, supply the ConfigurationFile.ini file to complete the process.

Sample Syntax

Following are some examples on how to use the configuration file:

- To specify the configuration file at the command prompt:

```
Setup.exe /ConfigurationFile=MyConfigurationFile.INI
```

- To specify passwords at the command prompt instead of in the configuration file:

```
Setup.exe /SQLSVCPASSWORD="************" /AGTSVCPASSWORD="************" /ASSVCPASSWORD="************"
/ISSVCPASSWORD="************" /RSSVCPASSWORD="************" /ConfigurationFile=MyConfigurationFile.INI
```

See Also

Install SQL Server 2014 from the Command Prompt
SQL Server Failover Cluster Installation
Upgrade a SQL Server Failover Cluster

Install SQL Server 2014 Using SysPrep

3/29/2019 • 17 minutes to read • Edit Online

SQL Server SysPrep related setup actions can be accessed through the Installation Center. The **Advanced** Page of the **Installation Center** has two options - **Image preparation of a stand-alone instance of SQL Server** and **Image completion of a prepared stand-alone instance of SQL Server**. The Prepare and Complete sections describe the installation process in detail. For more information, see Considerations for Installing SQL Server Using SysPrep.

You can also prepare and complete an instance of SQL Server using the command prompt or a configuration file. For more information, see:

- Install SQL Server 2014 from the Command Prompt
- Install SQL Server 2014 Using a Configuration File

Prerequisites

Before you install SQL Server, review the topics in Planning a SQL Server Installation.

For more information about SQL Server editions and the hardware and software requirements, see Hardware and Software Requirements for Installing SQL Server 2014.

> **IMPORTANT**
>
> The following is not supported by SQL Server SysPrep:
>
> WOW 64 installations: a feature of 64-bit edition of Windows that enables 32-bit applications to run natively in 32-bit mode.

This section includes the following procedures:

- SQL Server SysyPrep Cluster Support
- Prepare a stand-alone instance of SQL Server
- Complete a prepared instance of SQL Server
- Add features to a prepared instance of SQL Server
- Remove features from a prepared instance of SQL Server
- Uninstalling a Prepared Instance
- Modifying or Uninstalling a Completed Instance of SQL Server.

SQL Server SysPrep Cluster Support

Beginning in SQL Server 2014, SysPrep supports clustered SQL Server instances in command line installations. For more information, see What is Sysprep?.

To Prepare a SQL Server Failover Cluster (Unattended)

1. Prepare the image (as discussed in Considerations for Installing SQL Server Using SysPrep) and capture the windows image through SysPrep Generalization. The following sample prepares the image:

```
Setup.exe /q /ACTION=PrepareImage l /FEATURES=SQLEngine /InstanceID =<MYINST>
/IACCEPTSQLSERVERLICENSETERMS
```

Then run Windows SysPrep Generalization.

2. Deploy the image by running Windows SysPrep Specialize.
3. Create the Windows Failover Cluster.
4. Run setup.exe with `/ACTION=PrepareFailoverCluster` all nodes. For example:

```
setup.exe /q /ACTION=PrepareFailoverCluster /InstanceName=<InstanceName> /Features=SQLEngine
/SQLSVCACCOUNT="<DomainName\UserName>" /SQLSVCPASSWORD="xxxxxxxxxxx"  /IACCEPTSQLSERVERLICENSETERMS
```

Complete a SQL Server Failover Cluster (Unattended)

1. Run setup.exe with `/ACTION=CompleteFailoverCluster` on the node that owns the available storage group:

```
setup.exe /q /ACTION=CompleteFailoverCluster /InstanceName=<InstanceName>  /FAILOVERCLUSTERDISKS="
<Cluster Disk Resource Name - for example, 'Disk S:'>:" /FAILOVERCLUSTERNETWORKNAME="<Insert FOI
Network Name>" /FAILOVERCLUSTERIPADDRESSES="IPv4;xx.xxx.xx.xx;Cluster Network;xxx.xxx.xxx.x"
/FAILOVERCLUSTERGROUP="MSSQLSERVER" /INSTALLSQLDATADIR="<Drive>:\<Path>\MSSQLSERVER"
/SQLCOLLATION="SQL_Latin1_General_CP1_CS_AS" /SQLSYSADMINACCOUNTS="<DomainName\UserName>"
```

Adding a Node to an Existing SQL Server Failover Cluster (Unattended)

1. Deploy the image by running Windows SysPrep Specialize.
2. Join the Windows Failover Cluster.
3. Run setup.exe with `/ACTION=AddNode` on all nodes:

```
setup.exe /q /ACTION=AddNode /InstanceName=<InstanceName> /Features=SQLEngine  /SQLSVCACCOUNT="
<DomainName\UserName>" /SQLSVCPASSWORD="xxxxxxxxxxx"  /IACCEPTSQLSERVERLICENSETERMS
```

Prepare Image

Prepare a stand-alone instance of SQL Server.

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.
2. The Installation Wizard runs the SQL Server Installation Center. To prepare an instance of SQL Server, click **Image preparation of a stand-alone instance of SQL Server** on the **Advanced** page.
3. The System Configuration Checker runs a discovery operation on your computer. To continue, click **OK**. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.
4. On the Product Updates page, the latest available SQL Server product updates are displayed. If you don't want to include the updates, clear the **Include SQL Server product updates** check box. If no product updates are discovered, SQL Server Setup does not display this page and auto advances to the **Install Setup Files** page.
5. On the Install Setup files page, Setup provides the progress of downloading, extracting, and installing the Setup files. If an update for SQL Server Setup is found, and is specified to be included, that update will also be installed.
6. The System Configuration Checker verifies the system state of your computer before Setup continues. You

can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.

7. On the **Prepare Image Type** page, select **Prepare a new instance of SQL Server**.

   The **Prepare Image Type** page is displayed only when you have an existing un-configured prepared instance of SQL Server on the machine. You can choose to prepare a new instance of SQL Server or add sys prep supported features to an existing prepared instance of SQL Server on the machine. For more information on how to add features to a prepared instance of SQL Server see Add Features to a prepared instance.

8. On the **License Terms** page, read the license agreement, and then select the check box to accept the license terms and conditions. To help improve SQL Server, you can also enable the feature usage option and send reports to Microsoft.

9. On the **Feature Selection** page, select the components for your installation:

SQL Server 2014 SysPrep	Database Engine SQL Server Replication Full-Text Features Data Quality Services Reporting Services in Native mode Analysis Services Redistributable Features Shared Features

   A description for each component group appears in the right pane when you highlight the feature name. You can select any combination of check boxes. For more information, see Editions and Components of SQL Server 2014 and Features Supported by the Editions of SQL Server 2014.

   The prerequisites for the selected features are displayed on the right-hand pane. SQL Server Setup will install the prerequisite that are not already installed during the installation step described later in this procedure.

10. On the **Prepare Image Rules** page, the System Configuration Checker verifies the system state of your computer before Setup continues. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.

11. On the Instance Configuration page, specify the Instance ID for the Instance. Click **Next** to continue.

    Instance ID - The Instance ID 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. If the prepared instance is completed as a default Instance during the Complete step, the instance name is overwritten as MSSQLSERVER. The Instance ID remains the same as specified.

    Instance root directory - By default, the instance root directory is C:\Program Files\Microsoft SQL Server\120\. To specify a non-default root directory, use the field provided, or click **Browse** to locate an installation folder. The directory specified in the prepare step will be used during configuration in the Complete step.

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

Installed Instances - The grid shows instances of SQL Server that are on the computer where Setup is running.

12. The **Disk Space Requirements** page calculates the required disk space for the features that you specify. Then it compares the required space to the available disk space.
13. The System Configuration Checker will run prepare image rules to validate your computer configuration with the SQL Server features that you have specified. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.
14. The **Ready to Prepare Image** page shows a tree view of installation options that were specified during Setup. On this page, Setup indicates whether the Product Update feature is enabled or disabled and the final update version. To continue, click **Prepare**. SQL Server Setup will first install the required prerequisites for the selected features followed by the feature installation.
15. During installation, the **Prepare Image Progress** page provides status so that you can monitor installation progress as Setup continues.
16. 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**.
17. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you have finished with Setup. For more information, see View and Read SQL Server Setup Log Files.
18. This completes the prepare step. You may complete the image or deploy the prepared image as described in Considerations for Installing SQL Server Using SysPrep.

Complete Image

Complete a Prepared Instance of SQL Server

1. If you have a prepared instance of SQL Server included in the image of you machine, you will see a shortcut in the Start Menu. You can also launch the Installation Center and click **Image completion of a prepared stand-alone instance** on the **Advanced** page.
2. The System Configuration Checker runs a discovery operation on your computer. To continue, click **OK**. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.
3. On the **Setup Support Files** page, click **Install** to install the Setup support files.
4. The System Configuration Checker verifies the system state of your computer before Setup continues. After the check is complete, click **Next** to continue. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.
5. On the **Product Key** page, select an option 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 Editions and Components of SQL Server 2014. If you are installing Evaluation edition the 180-day trial period starts when you complete this step.
6. On the **License Terms** page, read the license agreement, and then select the check box to accept the license terms and conditions. To help improve SQL Server, you can also enable the feature usage option and send reports to Microsoft.
7. On the **Select a Prepared Instance** page select the prepared instance you want to complete from the drop down box. Select the Un-configured instance from the **Instance ID** list.

   Installed instances: This grid displays all the instances including any prepared instance on this machine.

8. On the **Feature Review** page, you will see the selected features and components included in the install during the prepare step. If you wish to add more features to your SQL Server Instance not included in the prepared instance, you must first complete this step to complete the SQL Server Instance, then add the features from the **Add Features** on the **Installation Center**.

   > **NOTE**
   > You can add features that are available for the product version that you are installing. For more information, see Features Supported by the Editions of SQL Server 2014

9. On the Instance Configuration page, specify the Instance name for the prepared Instance. This is the name of the instance once you have completed the configuration of SQL Server. Click **Next** to continue.

   Instance ID - The Instance ID 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. If the prepared instance is completed as a default Instance during the Complete step, the instance name is overwritten as MSSQLSERVER. The Instance ID remains the same as specified during the Prepare step.

   Instance root directory -The directory specified in the prepare step will be used and cannot be modified in this step.

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

   Installed instances - The grid shows instances of SQL Server that are on the computer where Setup is running.

10. Work flow for the rest of this topic depends on the features that were selected during the prepare step. You might not see all the pages, depending on the selections.

11. 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 that 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 have to have to complete their tasks. For more information, see Server Configuration - Service Accounts and Configure Windows Service Accounts and Permissions.

    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**.

12. Use the **Server Configuration - Collation** tab to specify non-default collations for the Database Engine and Analysis Services. For more information, see Server Configuration - Collation.

13. 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 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**.

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

> **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 Database Engine Configuration - Data Directories.

15. Use the Database Engine Configuration - FILESTREAM page to enable FILESTREAM for your instance of SQL Server. For more information, see Database Engine Configuration - Filestream.

16. Use the Reporting Services Configuration page to specify the kind of Reporting Services installation to create. For more information about Reporting Services configuration modes, see Reporting Services Configuration Options (SSRS).

17. On the **Error Reporting** page, specify the information that you want to send to Microsoft that will help improve SQL Server. By default, options for error reporting is enabled.

18. On the **Complete Image Rules** page, the System Configuration Checker will run the complete image rules to validate your computer configuration with the SQL Server configurations that you have specified. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.

19. The **Ready to Complete Image** page shows a tree view of installation options that were specified during Setup. To continue, click **Install**.

20. During installation, the **Complete Image Progress** page provides status so that you can monitor installation progress as Setup continues.

21. 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**.

22. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you have finished with Setup. For more information, see View and Read SQL Server Setup Log Files.

23. This step completes the configuration of the prepared instance of SQL Server and you have completed the installation of SQL Server.

Add Features to a Prepared Instance

Add Features to a Prepared Instance of SQL Server

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.

2. The Installation Wizard runs the SQL Server Installation Center. To add features to a prepared instance of

SQL Server, click **Image preparation of a stand-alone instance of SQL Server** on the **Advanced** page.

3. The System Configuration Checker runs a discovery operation on your computer. To continue, click **OK**. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.
4. On the Setup Support Files page, click **Install** to install the Setup support files.
5. On the **Prepare Image Type** page, select **Add features to an existing prepared instance of SQL Server** option. Select the specific prepared instance you want to add features to from the drop down list of available prepared instances.
6. On the **Feature Selection** page, specify the features you want to add to the specified prepared instance.

   The prerequisites for the selected features are displayed on the right-hand pane. SQL Server Setup will install the prerequisite that are not already installed during the installation step described later in this procedure.
7. On the **Prepare Image Rules** page, the System Configuration Checker verifies the system state of your computer before Setup continues. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.
8. The Disk Space Requirements page calculates the required disk space for the features that you specify. Then it compares the required space to the available disk space.
9. On the **Prepare Image Rules** page, the System Configuration Checker will run prepare image rules to validate your computer configuration with the SQL Server features that you have specified. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.
10. The **Ready to Prepare Image** page shows a tree view of installation options that were specified during Setup. To continue, click **Install**. SQL Server Setup will first install the required prerequisites for the selected features followed by the feature installation.
11. During installation, the **Prepare Image Progress** page provides status so that you can monitor installation progress as Setup continues.
12. 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**.
13. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you have finished with Setup. For more information, see View and Read SQL Server Setup Log Files.

Remove Features from a Prepare Instance

Removing features from a prepared instance of SQL Server

1. To begin the uninstall process, from the **Start** menu click **Control Panel** and double click **Program and Features**.
2. Double click the SQL Server component to uninstall and click **Remove**.
3. Setup support rules will run to verify your computer configuration. Click **OK** to continue.
4. On the **Select Instance** page, select the prepared instance you want to modify. The name of the prepared instance will be displayed as "Unconfigured PreparedInstanceID" where PreparedInstanceID is the instance you select.

5. On the **Select Features** page, specify the features to remove from the SQL Server instance you specified. Click **Next** to continue.
6. Removal rules will run to verify that the operation can complete successfully.
7. On the **Ready to Remove** page, review the list of components and features that will be uninstalled.
8. The **Remove Progress** page will display the status of the operation.
9. On the **Complete** page you can review the completion status of the operation. Click **close** to exit the installation wizard.

Uninstalling a Prepared Instance

Uninstall a prepared instance of SQL Server

1. To begin the uninstall process, from the **Start** menu click **Control Panel** and double click **Program and Features**.
2. Double click the SQL Server component to uninstall and click **Remove**.
3. Setup support rules will run to verify your computer configuration. Click **OK** to continue.
4. On the **Select Instance** page, select the prepared instance you want to modify. The name of the prepared instance will be displayed as "Unconfigured PreparedInstanceID" where PreparedInstanceID is the instance you select.
5. On the **Select Features** page, specify the features to remove from the SQL Server instance you specified. Click **Next** to continue.
6. On the **Removal Rules** page, Setup will run rules to verify that the operation can complete successfully.
7. On the **Ready to Remove** page, review the list of components and features that will be uninstalled.
8. The **Remove Progress** page will display the status of the operation.
9. On the **Complete** page you can review the completion status of the operation. Click **close** to exit the installation wizard.
10. Repeat steps 1 to 9 until all components of SQL Server 2014 have been removed.

Modifying or Uninstalling a Completed Instance of SQL Server.

The process to add or remove features or to uninstall a completed instance of SQL Server is similar to the process to an installed instance of SQL Server. For more information, see the following topics:

- Add Features to an Instance of SQL Server 2014 (Setup)
- Uninstall an Existing Instance of SQL Server (Setup)

See Also

What is Windows SysPrep
How does Windows SysPrepWork

Add Features to an Instance of SQL Server 2014 (Setup)

11/28/2018 • 9 minutes to read • Edit Online

This topic provides a step-by-step procedure for adding features to an instance of SQL Server 2014. Some SQL Server components or services are specific to an instance of SQL Server. These are also known as instance-aware. They share the same version as the instance that hosts them, and are used exclusively for that instance. You can add the instance-aware components to an instance SQL Server, along with the shared components of if they are not already installed. For a list of features that are supported by the editions of SQL Server, see Features Supported by the Editions of SQL Server 2014.

To add features to an instance of SQL Server from the command prompt, see Install SQL Server 2014 from the Command Prompt.

Prerequisites

Before you continue, review topics in Planning a SQL Server Installation.

> **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 permissions on the remote share.

> **NOTE**
> When you add features to an instance of SQL Server 2014, the existing usage report settings are applied to the newly-added features. To change these settings, use the **SQL Server Error and Usage Reporting** tool on the SQL Server**Configuration Tools** menu.

Procedures

To add features to an instance of SQL Server 2014

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 SQL Server 2012 Setup dialog box appears, Click **OK**. to install the prerequisites, then click **Cancel** to quit SQL Server 2014 installation.

2. The Installation Wizard will launch the SQL Server Installation Center. To add a new feature to an existing instance of SQL Server 2014, click **Installation** in the left-hand navigation area, and then click **New SQL Server stand-alone installation or add features to an existing installation**.

3. The System Configuration Checker will run a discovery operation on your computer. To view the verification details click **View Details**. To continue, Click **OK**..

4. On the Product Updates page, the latest available SQL Server product updates are displayed. If you don't want to include the updates, clear the **Include SQL Server product updates** check box. If no product updates are discovered, SQL Server Setup does not display this page and auto advances to the **Install Setup Files** page.

5. On the Install Setup files page, Setup provides the progress of downloading, extracting, and installing the Setup files. If an update for SQL Server Setup is found, and is specified to be included, that update will also be installed. Click **Install** to install Setup Support files.

6. The System Configuration Checker will verify the system state of your computer before Setup continues.

7. On the Installation Type page, select the option **Add features to an existing instance of SQL Server 2014**, and select the instance you would like to update.

8. 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 2014. 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 prerequisites for the selected features are displayed on the right-hand pane. SQL Server Setup will install the prerequisite that are not already installed during the installation step described later in this procedure.

   The System Configuration Checker will verify the system state of your computer before Setup continues. Click **Next** to continue.

9. 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.

10. 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.

11. 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 Server Configuration - Service Accounts and Configure Windows Service Accounts and Permissions.

    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**.

12. Use the **Server Configuration - Collation** tab to specify non-default collations for the Database Engine and Analysis Services. For more information, see Server Configuration - Collation.

13. 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 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**.

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

    > **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 Database Engine Configuration - Data Directories.

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

16. Use the Analysis Services Configuration - Account Provisioning page to specify the server mode and the users or accounts that will have administrator permissions for Analysis Services. Server mode determines which memory and storage subsystems are used on the server. Different solution types run in different server modes. If you plan to run multidimensional cube databases on the server, choose the default option, Multidimensional and Data Mining server mode. Regarding administrator permissions, 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 about server mode and administrator permissions, see Analysis Services 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 Analysis Services Configuration - Data Directories page to specify non-default installation directories. To install to default directories, click **Next**.

    > **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 Analysis Services Configuration - Data Directories.

18. Use the Reporting Services Configuration page to specify the type of Reporting Services installation to create. For more information about Reporting Services configuration modes, see Reporting Services Configuration Options (SSRS).

19. Use the Distributed Replay Controller Configuration page to specify the users you want to grant administrative permissions to for the Distributed Replay controller service. Users that have administrative permissions will have unlimited access to the Distributed Replay controller service.

Click the **Add Current User** button to add the users to whom you want to grant access permissions for the Distributed Replay controller service. Click the **Add** button to add access permissions for the Distributed Replay controller service. Click the **Remove** button to remove access permissions from the Distributed Replay controller service.

To continue, click **Next**.

20. Use the Distributed Replay Client Configuration page to specify the users you want to grant administrative permissions to for the Distributed Replay client service. Users that have administrative permissions will have unlimited access to the Distributed Replay client service.

    Controller Name is an optional parameter, and the default value is <*blank*>. Enter the name of the controller that the client computer will communicate with for the Distributed Replay client service. Note the following:

    - If you have already set up a controller, enter the name of the controller while configuring each client.
    - If you have not yet set up a controller, you can leave the controller name blank. However, you must manually enter the controller name in the **client configuration** file.

    Specify the **Working Directory** for the Distributed Replay client service. The default working directory is <*drive letter*>:\Program Files\MicrosoftSQL Server\DReplayClient\WorkingDir\.

    Specify the **Result Directory** for the Distributed Replay client service. The default result directory is <*drive letter*>:\Program Files\MicrosoftSQL Server\DReplayClient\ResultDir\.

    To continue, click **Next**.

21. On the Error 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 is enabled.

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

23. The Ready to Install page displays a tree view of installation options that were specified during Setup. On this page, Setup indicates whether the Product Update feature is enabled or disabled and the final update version. After you click install, SQL Server Setup will first install the required prerequisites for the selected features followed by the feature installation.

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

25. 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**.

26. 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 View and Read SQL Server Setup Log Files.

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 Surface Area Configuration.

See Also

View and Read SQL Server Setup Log Files

Validate a SQL Server Installation
Drop a SQL Server 2014 Installation
Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)
Install SQL Server 2014 from the Command Prompt

Drop a SQL Server 2014 Installation

6/13/2019 • 2 minutes to read • Edit Online

Repair operation can be used in the following scenarios:

- Repair an instance of SQL Server that is corrupted after it was successfully installed.
- Repair an instance of SQL Server if the upgrade operation is cancelled or fails after the instance name is mapped to the newly-upgraded instance.
  - If you see the following message in the summary log, you can repair the failed upgrade instance:

    "SQL Server upgrade failed. To continue, investigate the reason for the failure, correct the problem, and then repair your installation."
  - If you see the following message in the summary log, you need to uninstall and reinstall SQL Server. You cannot repair the SQL Server instance.

    "SQL Server upgrade failed. To continue, investigate the reason for the failure, correct the problem."

When you repair an instance of SQL Server:

- All missing or corrupt files are replaced.
- All missing or corrupt registry keys are replaced.
- All missing or invalid configuration values are set to default values.

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. For more information, see Add or Remove Nodes in a SQL Server Failover Cluster (Setup).

To repair a failed installation of SQL Server from the Installation Center

1. Launch the SQL Server Setup program (setup.exe) from SQL Server 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.

   > **TIP**
   > If the Installation Center was launched using the start menu, you will need to provide the location of the installation media at this time.

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.

To repair a failed installation of SQL Server using Command Prompt

1. Run the following command at a command prompt:

```
Setup.exe /q /ACTION=Repair /INSTANCENAME=instancename
```

See Also

View and Read SQL Server Setup Log Files
Installation How-to Topics

Rename a Computer that Hosts a Stand-Alone Instance of SQL Server

12/13/2018 • 4 minutes to read • Edit Online

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 the 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 Log Shipping and Replication (SQL Server).
- 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 Rename 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.

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:

```
sp_dropserver <old_name>;
GO
sp_addserver <new_name>, local;
GO
```

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>;
GO
sp_addserver <new_name\instancename>, local;
GO
```

Restart the instance of SQL Server.

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.

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';
```

Additional Considerations

Remote Logins - If the computer has any remote logins, running **sp_dropserver** might generate an error similar to the following:

```
Server: Msg 15190, Level 16, State 1, Procedure sp_dropserver, Line 44 There are still remote logins for the server 'SERVER1'.
```

To resolve the error, you must drop remote logins for this server.

To drop remote logins

- For a default instance, run the following procedure:

```
sp_dropremotelogin old_name;
GO
```

- For a named instance, run the following procedure:

```
sp_dropremotelogin old_name\instancename;
GO
```

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 sp_addlinkedserver (Transact-SQL) or sp_setnetname (Transact-SQL).

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 `\\SRVR1\pipe\sql\query`. 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 Creating a Valid Connection String Using Named Pipes.

See Also

Install SQL Server 2014

Install SQL Server 2014 Servicing Updates

3/29/2019 • 2 minutes to read • Edit Online

This topic provides information about installing updates for SQL Server 2014. This section provides information about the following:

- Installing updates for SQL Server 2014 during a new installation
- Installing updates for SQL Server 2014 after it has already been installed.

We recommend that customers evaluate and install latest SQL Server updates in a timely manner to make sure that systems are up-to-date with the most recent security updates.

Installing Updates for SQL Server 2014 During a New Installation

SQL Server setup integrates the latest product updates with the main product installation so that the main product and its applicable updates are installed at the same time. Product Update can search for the applicable updates from:

- Microsoft Update
- Windows Server Update Services (WSUS)
- A local folder
- A network share

After Setup finds the latest versions of the applicable updates, it downloads and integrates them with the current SQL Server setup process. Product Update can include a cumulative update, service pack, or service pack plus cumulative update. Product Update functionality is an extension of the Slipstream Functionality that was available in SQL Server 2008 PCU1.

Installing Updates for SQL Server 2014 after it has already been installed

On an installed instance of SQL Server 2014, we recommend that you apply all available updates: General Distribution Releases (GDR - security/critical updates), Service Packs (SP), as well as the latest available Cumulative Update (CU).

Depending on the type of servicing release, SQL Server updates are available via Microsoft Update (MU), the Microsoft Download Center, and/or the Customer Support Services hotfix Server. Security and Critical updates for SQL Server are automatically provided by Microsoft Update (to be able to see these updates you need to opt-in to MU through Windows Update in Control panel). Service Packs are available on MU as Optional/Important downloads as well as the Download Center. Cumulative updates are available on the Microsoft hotfix download server provided in CU Knowledge Base articles.

See Also

Install SQL Server 2014 from the Installation Wizard (Setup)
Installing updates from the command prompt Add Features to an Instance of SQL Server 2014 (Setup)
Drop a SQL Server 2014 Installation

View and Read SQL Server Setup Log Files

11/16/2018 • 3 minutes to read • Edit Online

Each execution of Setup creates log files are created with a new timestamped log folder at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\. The time-stamped log folder name format is YYYYMMDD_hhmmss. When Setup is run in an unattended mode, the logs are created at % temp%\sqlsetup*.log. All files in the logs folder are archived into the Log*.cab file in their respective log folder.

A typical Setup request goes through three execution phases:

1. Global rules text
2. Component update
3. User-requested action

In each phase, Setup generates detail and summary logs with additional logs created as appropriate. Setup is called at least three times per user-requested Setup action.

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 subfolder under the time-stamped log folder, as follows:

- Datastore_GlobalRules
- Datastore_ComponentUpdated
- Datastore

The following sections describe SQL Server Setup log files.

Summary Text

Overview

This file shows the SQL Server components that were detected during Setup, the operating system environment, command-line parameter values if they are specified, and the overall status of each MSI/MSP that was executed.

The log is organized into the following sections:

- An overall summary of the execution
- Properties and the configuration of the computer where SQL Server Setup was run
- SQL Server product features previously installed on the computer
- Description of the installation version and installation package properties
- Runtime input settings that are provided during install
- Location of the configuration file
- Details of the execution results
- Global rules
- Rules specific to the installation scenario

- Failed rules
- Location of the rules report file

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\.

To find errors in the summary text file, search the file by using the "error" or "failed" keywords.

Summary_engine-base_YYYYMMDD_HHMMss.txt

Overview

The summary_engine base file is similar to the summary file and is generated during the main workflow.

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\.

Summary_engine-base_YYYYMMDD_HHMMss_ComponentUpdate.txt

Overview

The component update summary log file is similar to the summary file and is generated during the component update workflow.

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\.

Summary_engine-base_<VersionNumber>MMDD_HHMMss_GlobalRules.txt

Overview

The global rules summary log file is similar to the summary file generated during the global rules workflow.

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\.

Detail.txt

Overview

Detail.txt is generated for the main workflow such as install or upgrade, and provides the details of the execution. The logs in the file are generated based on the time when each action for the installation was invoked, and show the order in which the actions were executed, and their dependencies.

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup

Bootstrap\Log\<YYYYMMDD_HHMM>\Detail.txt.

If an error occurs during the Setup process, the exception or error are logged at the end of this file. To find the errors in this file, first examine the end of the file followed by a search of the file for the "error" or "exception" keywords.

Detail_ComponentUpdate.txt

Overview

The Detail_ComponentUpdate.txt file is generated for the component update workflow and is similar to Detail.txt.

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\.

Detail_GlobalRules.txt

Overview

Detail_GlobalRules.txt is generated for the global rules execution and is similar to Detail.txt.

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\.

MSI log files

Overview

The MSI log files provide details of the installation package process. They are generated by the MSIEXEC during the installation of the specified package.

Types of MSI log files:

- <Feature>_<Architecture>_<Interaction>.log
- <Feature>_<Architecture>_<Language>_<Interaction>.log
- <Feature>_<Architecture>_<Interaction>_<workflow>.log

Location

The MSI log files are located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\<Name>.log.

At the end of the file is a summary of the execution which includes the success or failure status and properties. To find the error in the MSI file, search for "value 3" and usually the errors can be found close to the string.

ConfigurationFile.ini

Overview

The configuration file contains the input settings that are provided during installation. It can be used to restart the installation without having to enter the settings manually. However, passwords for the accounts, PID, and some parameters are not saved in the configuration file. The settings can be either added to the file or provided by using the command line or the Setup user interface. For more information, see Install SQL Server 2014 Using a Configuration File.

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\.

SystemConfigurationCheck_Report.htm

Overview

The system configuration check report contains a short description for each executed rule, and the execution status.

Location

It is located at %programfiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\<YYYYMMDD_HHMM>\.

See Also

Installation How-to Topics
Install SQL Server 2014

Install SQL Server 2014 BI Features

10/3/2019 • 2 minutes to read • Edit Online

SQL Server features that are part of the Microsoft Business Intelligence platform include Analysis Services, Integration Services, Master Data Services, Reporting Services, and several client applications used for creating or working with analytical data. This section of the SQL Server Setup documentation explains how to install these features.

Analysis Services and Reporting Services can be installed as standalone servers, in scale-out configurations, or as shared service applications in a SharePoint farm. Installing the services in a farm enables BI features that are only available in SharePoint, including PowerPivot for SharePoint and Power View, the Reporting Services ad hoc interactive report designer that runs on PowerPivot or Analysis Services tabular model databases.

If you are already familiar with the installation steps for Reporting Services, Analysis Services, or PowerPivot for SharePoint, you can skip ahead to the checklists for guidance on how to enable specific scenarios. For more information, see Checklists for Installing BI Features with SharePoint.

Contents

In this section:

LINK	TASK
Checklists for Installing BI Features with SharePoint	If you already know what you want to install and are familiar with the installation steps for Reporting Services, Analysis Services, or PowerPivot for SharePoint, you can use the checklists in this section for guidance on installation order, account and permission requirements, and steps deploying advanced topologies, including multi-server and multi-feature deployments.
Install SQL Server BI Features with SharePoint (PowerPivot and Reporting Services)	This section explains how to install SQL Server features in a SharePoint environment. It identifies which SQL Server features are available given a specific version and edition of SharePoint. It also includes installation procedures for PowerPivot for SharePoint and Reporting Services in SharePoint mode.
Install Analysis Services in Multidimensional and Data Mining Mode Install Analysis Services in Tabular Mode Install Data Quality Services Install Integration Services Install Master Data Services Install Reporting Services Native Mode Report Server	This section provides instructions for installing Analysis Services, Integration Services, Master Data Services, and Reporting Services, where Analysis Services and Reporting Services are installed without SharePoint. This is sometimes referred to as *native mode*, and it is the most common installation scenario for both Reporting Services and Analysis Services. In this section, you will learn about installation options that directly determine the operational context of your server. For Reporting Services, this might be a server that is pre-configured or one that requires multiple configuration steps before you can use it. For Analysis Services, the installation options you select will determine the type of project you can deploy to the server.
Verify or Troubleshoot SQL Server BI Feature Installation Problems	This section includes steps for verifying an installation. It also provides links to problem resolution information on the web.

Related content

LINK	TASK
Upgrade to SQL Server 2014 Upgrade Analysis Services Upgrade PowerPivot for SharePoint Upgrade and Migrate Reporting Services	Use the instructions in this section to upgrade servers and content from a previous release to SQL Server 2014.
Uninstall SQL Server 2014 Uninstall PowerPivot for SharePoint Uninstall Reporting Services	Use the instructions in this section to uninstall BI features.

See Also

- What's New (Reporting Services)
- What's New in Analysis Services and Business Intelligence
- Install SQL Server 2014
- Upgrade to SQL Server 2014

Install Analysis Services in Multidimensional and Data Mining Mode

3/4/2020 • 2 minutes to read • Edit Online

Analysis Services provides online analytical processing (OLAP) and data mining functionality for business intelligence applications. In this release, support for OLAP databases and data mining models is available when you install Analysis Services in *Multidimensional mode*. Multidimensional mode is one of three server modes that Analysis Services runs in. It is the default mode. If you install Analysis Services using default values, you will get an instance that runs multidimensional databases and data mining models.

Analysis Services is a multi-instance feature, which means that you can install more than one instance Analysis Services on a single computer, or run a new instance of Analysis Services side-by-side an earlier version. Server mode is specific to an instance. Using other modes requires that you install additional instances of the server.

You can install Analysis Services by itself or with other components. If you install just Analysis Services, the following features are installed when you select **Analysis Services** on the Feature Selection page of the SQL Server Installation Wizard:

- Analysis Services server for running Analysis Services databases and data mining models
- Data providers used for Analysis Services data access to source databases
- SQL Server Configuration Manager

Choosing additional features

Analysis Services OLAP and data warehouse solutions will require the installation of additional SQL Server components to enable the development, deployment, and administration of Analysis Services databases. The following additional features are options for many typical user scenarios:

- SQL Server Data Tools (SSDT), used to create and view Analysis Services data structures and data mining models.
- Client tools connectivity components, used for communication between clients and servers, including network libraries for DB-Library, ODBC, and OLE DB.
- Integration Services, a set of graphical and programmable objects for moving, copying, and transforming data.
- Management tools, including SQL Server Configuration Manager, SQL Server Management Studio, SQL Server Profiler, and Replication Monitor.

Installation Tasks

Installation tasks include the following:

LINKS	TASKS
Hardware and Software Requirements for Installing SQL Server 2014 and Configure Windows Service Accounts and Permissions.	Before you run Setup, check prerequisites for installing Analysis Services and determine which account you will use to provision the server.

LINKS	TASKS
Install SQL Server 2014 from the Installation Wizard (Setup).	Run SQL Server Setup to install the software.
Configure the Windows Firewall to Allow Analysis Services Access	After Setup is finished, you must configure firewall settings to allow remote connections to the server.
Authorizing access to objects and operations (Analysis Services)	Users who access Analysis Services databases must have Read permission on at least one database on the server.

Related Content

Additional setup content can be found in the following topics:

Install Analysis Services in Tabular Mode

PowerPivot for SharePoint 2010 Installation

Determine the Server Mode of an Analysis Services Instance

SQL Server Data Mining Add-ins

By default, sample databases, sample code, and client application add-ins are not installed as part of SQL Server Setup. To install sample databases and sample code, see the CodePlex Web site.

See Also

Features Supported by the Editions of SQL Server 2012
Languages and Collations (Analysis Services)
Upgrade Analysis Services

Install Analysis Services in Tabular Mode

5/22/2019 • 2 minutes to read • Edit Online

If you are installing Analysis Services to use the new tabular modeling features, you must install Analysis Services in a server mode that supports that type of model. The server mode is Tabular, and it is configured during installation.

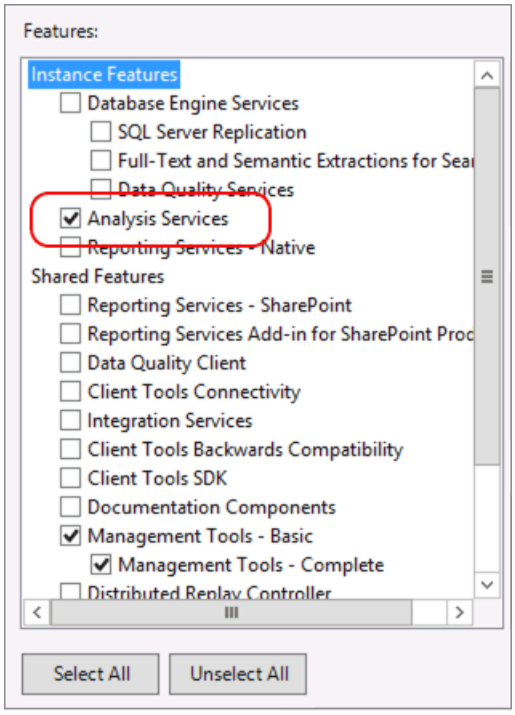
After you install the server in this mode, you can use it host solutions that you build in tabular model designer. A tabular mode server is required if you want tabular model data access over the network.

You can specify Tabular mode in the Installation Wizard or in a command line setup. The following sections describe each approach.

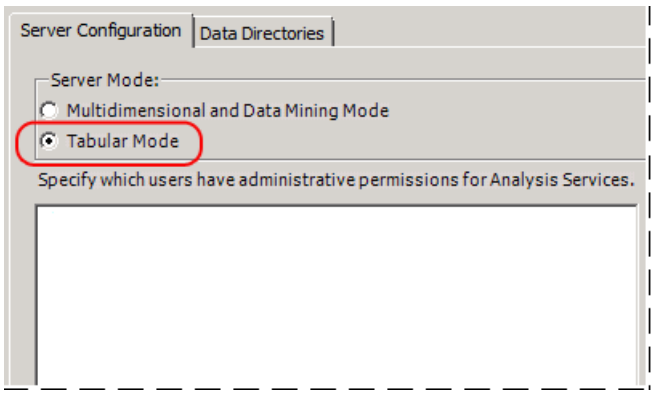
Installation Wizard

The following list shows you which pages in the SQL Server Installation wizard are used to install Analysis Services in Tabular mode:

1. Select **Analysis Services** from the Feature Tree in Setup.



2. On the Analysis Services Configuration page, be sure to select **Tabular Mode**.



Tabular mode uses the xVelocity in-memory analytics engine (VertiPaq), which is the default storage for tabular models that you deploy to Analysis Services. After you deploy tabular model solutions to the server, you can selectively configure tabular solutions to use DirectQuery disk storage as an alternative to memory-bound storage.

Command Line Setup

SQL Server Setup includes a new parameter (`ASSERVERMODE`) that specifies the server mode. The following example illustrates a command line setup that installs Analysis Services in Tabular server mode.

```
Setup.exe /q /IAcceptSQLServerLicenseTerms /ACTION=install /FEATURES=AS /ASSERVERMODE=TABULAR
/INSTANCENAME=ASTabular /INDICATEPROGRESS/ASSVCACCOUNT=<DomainName\UserName> /ASSVCPASSWORD=<StrongPassword>
/ASSYSADMINACCOUNTS=<DomainName\UserName>
```

`INSTANCENAME` must be less than 17 characters.

All placeholder account values must be replaced with valid accounts and password.

Tools such as SQL Server Management Studio or SQL Server Data Tools (SSDT) are not installed using the example command line syntax that is provided. For more information about adding features, see Install SQL Server 2014 from the Command Prompt.

`ASSERVERMODE` is case-sensitive. All values must be expressed in upper case. The following table describes the valid values for `ASSERVERMODE`.

VALUE	DESCRIPTION
MULTIDIMENSIONAL	This is the default value. If you do not set `ASSERVERMODE`, the server is installed in Multidimensional server mode.
POWERPIVOT	This value is optional. In practice, if you set the `ROLE` parameter, the server mode is automatically set to 1, making `ASSERVERMODE` optional for a PowerPivot for SharePoint installation. For more information, see Install PowerPivot from the Command Prompt.
TABULAR	This value is required if you are installing Analysis Services in Tabular mode using command line setup.

See Also

Determine the Server Mode of an Analysis Services Instance
Configure In-Memory or DirectQuery Access for a Tabular Model Database
Tabular Modeling (SSAS Tabular)

Install Data Quality Services

5/9/2019 • 5 minutes to read • Edit Online

SQL Server Data Quality Services (DQS) contains the following two components: **Data Quality Server** and **Data Quality Client**.

DQS COMPONENT	DESCRIPTION
Data Quality Server	Data Quality Server is installed on top of the SQL Server 2014 Database Engine, and includes three databases: DQS_MAIN, DQS_PROJECTS, and DQS_STAGING_DATA. DQS_MAIN contains DQS stored procedures, the DQS engine, and published knowledge bases. DQS_PROJECTS contains the data quality project information. DQS_STAGING_DATA is the staging area where you can copy your source data to perform DQS operations, and then export your processed data.
Data Quality Client	Data Quality Client is a standalone application that enables you to connect to Data Quality Server, and provides you with a highly-intuitive graphical user interface to perform data-quality operations, and other administrative tasks related to DQS.

IMPORTANT

Apart from the above two DQS components, you can also:

- Use the DQS Cleansing Transformation in Integration Services that performs data cleansing within the Integration Services packaging process, and is automatically installed when you install Integration Services. For information about installing Integration Services, see Install Integration Services.
- Enable DQS integration in Master Data Services to use the DQS matching functionality in the Master Data Services Add-in for Excel. For more information, see Enable Data Quality Services Integration with Master Data Services.

DQS installation is a three-part process:

- Pre-Installation Tasks: Verify system requirements before you install DQS.
- Data Quality Services Installation Tasks: Install DQS by using SQL Server Setup.
- Post-Installation Tasks: Perform these tasks after finishing with the SQL Server Setup to finish installing DQS.

NOTE

This topic does not include instructions for running Setup from the command line. For information about command-line options for installing Data Quality Server and client, see Feature Parameters in Install SQL Server 2014 from the Command Prompt.

Pre-Installation Tasks

Before installing DQS, make sure that your computer meets the minimum system requirements. The following

table provides information about the minimum system requirements for the DQS components:

DQS COMPONENT	MINIMUM SYSTEM REQUIREMENTS
Data Quality Server	Memory (RAM): -Minimum: 2 GB -Recommended: 4 GB or more SQL Server 2014 Database Engine. For more information, see About the SQL Server Database Engine.
Data Quality Client	.NET Framework 4.0 (installed during the Data Quality Client installation, if not already installed) Internet Explorer 6.0 SP1 or later

IMPORTANT

- Data Quality Server and Data Quality Client can be installed on the same computer, or different computers. Both the components can be installed independently of each other, and in any sequence. However, to use Data Quality Client, you will need a Data Quality Server installed to connect to.
- You can connect to the SQL Server 2014 version of Data Quality Server using either the current or earlier version of Data Quality Client and DQS Cleansing Transformation. For information about upgrading your existing version of DQS to SQL Server 2014, see Upgrade Data Quality Services.
- Although Microsoft Excel is not a prerequisite for installing Data Quality Client, Microsoft Excel 2003 or later must be installed on the Data Quality Client computer to perform various operations in the client application such as importing domain values from an excel file or mapping to the source data in an Excel file for knowledge discovery, cleansing, or matching activities.

For detailed information about the minimum system requirements for installing SQL Server 2014, see Hardware and Software Requirements for Installing SQL Server 2014.

Data Quality Services Installation Tasks

You have to use the SQL Server 2014 Setup to install DQS components. When you run the SQL Server Setup, you have to go through a series of installation wizard pages to select appropriate options based on your requirements. The following table lists only those pages in the installation wizard where the options that you select will affect your installation of DQS:

PAGE	ACTION

PAGE	ACTION
Feature Selection	Select: **Data Quality Services** under **Database Engine Services** to install the Data Quality Server. If you select the **Data Quality Services** check box, SQL Server Setup will copy an installer file, DQSInstaller.exe, under the SQL Server instance directory on your computer. You must run this file after you have completed the SQL Server Setup to *complete* the Data Quality Server installation. Also, you must perform some additional steps to configure your Data Quality Server before you can use it. For more information, see Post-Installation Tasks. **Data Quality Client** to install Data Quality Client. (Recommended) **Management Tools - Complete** under **Management Tools - Basic** to install SQL Server Management Studio. It provides you a graphical user interface to manage your SQL Server instance, and will aid you in performing additional tasks post installation as listed in the next section.
Database Engine Configuration	Click **Add Current User** to add your user Windows account to the sysadmin fixed server role. This is required for you to be able to run the DQSInstaller.exe file later to complete the Data Quality Server installation.

Post-Installation Tasks

After you complete the SQL Server installation wizard, you must perform additional steps mentioned in this section to complete your Data Quality Server installation, configure it, and then use it.

ACTION	DESCRIPTION	RELATED TOPICS

ACTION	DESCRIPTION	RELATED TOPICS
Complete the Data Quality Server installation	Run the DQSInstaller.exe file. On running the DQSInstaller.exe file: The DQS_MAIN, DQS_PROJECTS, and DQS_STAGING_DATA databases are created. The ##MS_dqs_db_owner_login## and ##MS_dqs_service_login## logins are created. The dqs_administrator, dqs_kb_editor, and dqs_kb_operator roles are created in the DQS_MAIN database. The DQInitDQS_MAIN stored procedure is created in the master database. DQS_install.log file is typically created in the C:\Program Files\Microsoft SQL Server\MSSQL12.*<instance_name>*\MSSQL\Log folder. The file contains information about the actions performed on running the DQSInstaller.exe file. If a Master Data Services database is present in the same SQL Server instance as Data Quality Server, a user mapped to the Master Data Services login is created, and is granted the dqs_administrator role on the DQS_MAIN database. This completes the Data Quality Server installation.	Run DQSInstaller.exe to Complete Data Quality Server Installation
Grant DQS Roles to Users	To log on to Data Quality Server using Data Quality Client, a user must have any of the following three roles on the DQS_MAIN database: **dqs_administrator**, **dqs_kb_editor**, or **dqs_kb_operator**. By default, if your user account is a member of the sysadmin fixed server role, you can log on to Data Quality Server using Data Quality Client even if none of the DQS roles are granted to your user account. For information about the three DQS roles, see DQS Security. Note: The three DQS roles are not available for the DQS_PROJECTS and DQS_STAGING_DATA databases.	Grant DQS Roles to Users

ACTION	DESCRIPTION	RELATED TOPICS
Make your data available for DQS operations	Make sure that you can access your source data for the DQS operations, and can export the processed data to a table in a database.	Access Data for the DQS Operations

See Also

Video: Install and Configure DQS
Upgrade SQLCLR Assemblies After .NET Framework Update
Export and Import DQS Knowledge Bases Using DQSInstaller.exe
Upgrade Data Quality Services
Remove Data Quality Server Objects
Install SQL Server 2014 BI Features
Uninstall SQL Server 2014
Data Quality Services
Troubleshoot Installation and Configuration Issues in DQS

Run DQSInstaller.exe to Complete Data Quality Server Installation

5/9/2019 • 6 minutes to read • Edit Online

To complete the Data Quality Server installation, you must run the DQSInstaller.exe file after installing SQL Server 2014. This topic describes how to run the DQSInstaller.exe from the **Start** screen, **Start** menu, Windows Explorer, or Command Prompt; you can choose any of the ways to run the DQSInstaller.exe file.

Prerequisites

- You must have selected **Data Quality Services** under **Database Engine Services** on the Feature Selection page in the SQL Server setup while installing SQL Server, and must have completed the SQL Server installation. For more information, see Install Data Quality Services.
- Your Windows user account must be a member of the sysadmin fixed server role in the SQL Server database engine instance.
- You must be logged on as a member of the Administrators group on the computer where you are running DQSInstaller.exe.

Run DQSInstaller.exe from Start Screen, Start Menu or Windows Explorer

1. On the computer where you chose to install Data Quality Server, run the DQSInstaller.exe file using any of the following as applicable:
   - **Start screen**: On the **Start** screen, click **Data Quality Server Installer.**
   - **Start menu**: On the taskbar, click **Start**, point to **All Programs**, click **Microsoft SQL Server 2014**. Under **Microsoft SQL Server 2014**, click **Data Quality Services**, and then click **Data Quality Server Installer.**
   - **Windows Explorer**: Locate the DQSInstaller.exe file. If you installed the default instance of SQL Server, the DQSInstaller.exe file will be available at C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn. Double-click the DQSInstaller.exe file.
2. A command prompt window appears that shows the status of the installation. You will notice the following three things:
   a. The installer creates an installation log file, DQS_install.log, which contains information about the actions performed on running the DQSInstaller.exe file. If you installed the default instance of SQL Server, the DQS_install.log file will be available at C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Log.
   b. The installer uses the default server collation, SQL_Latin1_General_CP1_CI_AS, for installing Data Quality Server.

> **IMPORTANT**
> You can provide a different server collation value for installing Data Quality Server only if you are running DQSInstaller.exe from command prompt. For more information, see Run DQSInstaller.exe from Command Prompt later in this topic.

c. The installer checks for any pending restarts on your computer owing to any updates that have been recently installed on your computer. If any pending restarts are found, a message appears to notify you about the same, and you can choose to continue or abort the installation by pressing Y or N respectively. We recommend that if there are any pending restarts, you must abort the installation, restart your computer, and then run the DQSInstaller.exe again.

3. You are prompted to type a password for the database master key. The database master key is required to encrypt the reference data service provider keys that will be stored in the DQS_MAIN database when you set up reference data providers in Data Quality Services (DQS) later.

> **IMPORTANT**
> The password must be at least 8 characters long, and must contain characters from three of the following four categories: English upper case letter (A, B, C,... Z), English lowercase letter (a, b, c,... z), Numeral (0, 1, 2,... 9), and nonalphanumeric or special character (~!@#$%^&*()_-+=|\{}[]:;"'<>,.?/). For example: P@ssword. The installer will prompt you to enter another password if the current password does not match the requirement.

4. Provide a password, confirm the password, and then press ENTER to continue with the installation.

> **IMPORTANT**
> You must retain the password specified for the database master key because you will require it while restoring the DQS databases from a backup in future, if you choose to do so. For more information about restoring DQS databases, see Backing Up and Restoring DQS Databases.

5. If a Master Data Services database is present in the same SQL Server instance as Data Quality Server, the installer creates a user mapped to the Master Data Services login, and grants it the dqs_administrator role on the DQS_MAIN database. For information about installing Master Data Services and creating a Master Data Services database, see Install Master Data Services.

6. A completion message is displayed after the installation has completed successfully. Press any key to close the command prompt window.

Run DQSInstaller.exe from Command Prompt

You can run DQSInstaller.exe from command prompt using the following command-line parameters:

DQSINSTALLER.EXE PARAMETER	DESCRIPTION	SAMPLE SYNTAX

DQSINSTALLER.EXE PARAMETER	DESCRIPTION	SAMPLE SYNTAX
-collation	The server collation to be used for installing Data Quality Server. DQS supports only case-insensitive collation. If you specify a case-sensitive collation, the installer attempts to use the case-insensitive version of the specified collation. If there is no case-insensitive version, or if the collation is unsupported by SQL, the Data Quality Server installation fails. If a server collation is not specified, the default collation, SQL_Latin1_General_CP1_CI_AS, is used.	`dqsinstaller.exe -collation <collation_name>`
-upgradedlls	Skips recreating the DQS databases (DQS_MAIN, DQS_PROJECTS, and DQS_STAGING_DATA), and only updates the SQL Common Language Runtime (SQLCLR) assemblies used by DQS in the SQL Server 2014 database. For more information, see Upgrade SQLCLR Assemblies After .NET Framework Update	`dqsinstaller.exe -upgradedlls`

-exportkbs	Export all the knowledge bases to a DQS backup file (.dqsb). You must also specify the full path and file name where you want to export all the knowledge bases.
	
	For more information, see Export and Import DQS Knowledge Bases Using DQSInstaller.exe.
		`dqsinstaller.exe -exportkbs <path><filename>`
		
		For example,
		`dqsinstaller.exe -exportkbs c:\DQSBackup.dqsb`

-importkbs	Import all the knowledge bases from a DQS backup file (.dqsb) after completing the Data Quality Server installation. You must also specify the full path and file name from where you want to import all the knowledge bases.
	
	For more information, see Export and Import DQS Knowledge Bases Using DQSInstaller.exe.
		`dqsinstaller.exe -importkbs <path><filename>`
		
		For example,
		`dqsinstaller.exe -importkbs c:\DQSBackup.dqsb`

-upgrade	Upgrades DQS databases schema. You must use this parameter after you have installed a SQL Server update on a previously configured DQS instance. For more information, see Upgrade DQS Databases Schema After Installing SQL Server Update.	`dqsinstaller.exe -upgrade`

DQSINSTALLER.EXE PARAMETER	DESCRIPTION	SAMPLE SYNTAX

-uninstall	Uninstalls Data Quality Server from the current SQL Server instance.
	
	You can also export all the knowledge bases in the existing Data Quality Server installation to a DQS backup file (.dqsb), and then uninstall Data Quality Server. For more information, see Export and Import DQS Knowledge Bases Using DQSInstaller.exe.
	
	** **Important** ** If you uninstall Data Quality Server from a SQL server instance using the `-uninstall` command line parameter, all the DQS objects are deleted as part of the uninstall process. You do not have to delete them manually after uninstalling Data Quality Server as mentioned in Remove Data Quality Server Objects.
		To just uninstall Data Quality Server:
		`dqsinstaller.exe -uninstall`
		
		To export all the knowledge bases to a file, and then uninstall Data Quality Server:
		`dqsinstaller.exe -uninstall <path><filename>`
		For example,
		`dqsinstaller.exe -uninstall c:\DQSBackup.dqsb`

To run DQSInstaller.exe from Command Prompt:

1. Start Command Prompt.
2. At the command prompt, change your directory to the location where DQSInstaller.exe is available. If you installed the default instance of SQL Server, the DQSInstaller.exe file will be available at C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn:

   ```
   cd C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn
   ```

3. At the command prompt, run DQSInstaller.exe with or without command-line parameters:
   - **Without Command-Line Parameter**: Type `dqsinstaller.exe`, and then press ENTER.
   - **With Command-Line Parameter**: Type the required command as mentioned in the table above, and then press ENTER.
4. The required actions are performed based on the specified command. If you just chose to install Data Quality Server without any command-line parameters, rest of the steps are same as described in steps 2-6 in the previous section, Run DQSInstaller.exe from Start Screen, Start Menu or Windows Explorer.

Next Steps

- Grant appropriate DQS roles to users based on their work profile. See Grant DQS Roles to Users.
- If Data Quality Server will be accessed remotely from Data Quality Client, enable the TCP/IP protocol using SQL Server Configuration Manager on this computer.
- Make sure that you can access your source data for the DQS operations, and can export the processed data to a table in a database. See Access Data for the DQS Operations.

See Also

Install Data Quality Services
Upgrade SQLCLR Assemblies After .NET Framework Update

Export and Import DQS Knowledge Bases Using DQSInstaller.exe

Grant DQS Roles to Users

5/9/2019 • 2 minutes to read • Edit Online

This topic describes how to create SQL logins based on a Windows principal, and grant Data Quality Services (DQS) roles on the DQS_MAIN database.

Prerequisites

- You must have completed the Data Quality Server installation by running the DQSInstaller.exe file. For more information, see Run DQSInstaller.exe to Complete Data Quality Server Installation.
- Your Windows user account must be a member of the appropriate fixed server role (such as securityadmin, serveradmin, or sysadmin) to create SQL login, and grant them DQS roles.

To create SQL login and grant DQS roles

1. Start SQL Server Management Studio.
2. In SQL Server Management Studio, expand your SQL Server instance, and then expand **Security**.
3. Right-click the **Security** folder, point to **New**, and then click **Login**.
4. In the **Login - New** dialog box, specify the name of a Windows user in the **Login name** box, specify the type of authentication as **Windows authentication**, and click **Search** to validate the user.
5. After the user is validated, click the **User Mapping** page in the left pane.
6. In the right pane, select the check box under the **Map** column for the **DQS_MAIN** database, and then select the **dqs_administrator**, **dqs_kb_editor**, or **dqs_kb_operator** check box in the **Database role membership for: DQS_MAIN** pane, depending on the access level needed for the user. For information about the three DQS roles, see DQS Security.
7. In the **Login - New** dialog box, click **OK** to apply the changes.

> **NOTE**
> If you grant the **dqs_administrator** role to a user, apply the changes, and then recheck the user permissions, the other two DQS roles check boxes (**dq_kb_editor** and **dqs_kb_operator**) are also selected.

Next Steps

Try logging on to Data Quality Server by using the Windows user account for which you just created a SQL login, and granted a DQS role.

See Also

Install Data Quality Services
Create a Login

Access Data for the DQS Operations

5/9/2019 • 2 minutes to read • Edit Online

To use your source data for Data Quality Services (DQS) operations, and export your processed data, you can do either of the following:

- Copy your source data to a table/view in the DQS_STAGING_DATA database, and then use it for DQS operations. You can also export the processed data to a new table in the DQS_STAGING_DATA database. To do so, your Windows user account must be granted read/write access to the DQS_STAGING_DATA database.
- Use your own database as the source data for the DQS operations, and destination for exporting the processed data. To do so, ensure that your database is in the same SQL Server instance as the Data Quality Server databases. Otherwise, the database will not be available in the Data Quality Client for DQS operations. Also, your Windows user account must be granted access on the DQS_STAGING_DATA database for exporting the matching results because matching results are exported in two phases: first, the matching results are exported to the temporary tables in the DQS_STAGING_DATA database, and then moved to the table in your destination database.

Prerequisites

- You must have completed the Data Quality Server installation by running the DQSInstaller.exe file. For more information, see Run DQSInstaller.exe to Complete Data Quality Server Installation.
- Your Windows user account must be a member of the appropriate fixed server role (such as securityadmin, serveradmin, or sysadmin) in the database engine instance to grant/modify access to SQL login on databases.

To grant read/write access to a User on the DQS_STAGING_DATA Database

1. Start Microsoft SQL Server Management Studio.
2. In Microsoft SQL Server Management Studio, expand your SQL Server instance, and expand **Security**, and then expand **Logins**.
3. Right-click a SQL login, and click **Properties**.
4. In the **Login Properties** dialog box, click the **User Mapping** page in the left pane.
5. In the right pane, select the check box under the **Map** column for the **DQS_STAGING_DATA** database, and then select the following roles in the **Database role membership for: DQS_STAGING_DATA** pane:
   - **db_datareader**: Read data from tables/views.
   - **db_datawriter**: Add, delete, or change data in tables.
   - **db_ddladmin**: Create, modify, or delete tables/views.
6. In the **Login Properties** dialog box, click **OK** to apply the changes.

Next Steps

Try performing DQS operations that accesses the database as data source for DQS operation, and then exports the processed data to the database.

See Also

Install Data Quality Services

Export and Import DQS Knowledge Bases Using DQSInstaller.exe

5/9/2019 • 3 minutes to read • Edit Online

For an existing installation of DQS, you can export all the knowledge bases in your Data Quality Server to a DQS backup file (.dqsb) at once, and then later use the .dqsb file to import all the knowledge bases to a different Data Quality Server at once by running the DQSInstaller.exe file from the command prompt. For more information about running DQSInstaller.exe from the command prompt, see Run DQSInstaller.exe from Command Prompt in Run DQSInstaller.exe to Complete Data Quality Server Installation.

This feature enables you to take a backup of *all* your knowledge bases in Data Quality Server at once without having to individually export each knowledge base to a .dqs file by using Data Quality Client. Similarly, you can import *all* the knowledge bases from the backup file into another Data Quality Server at once without having to individually import each knowledge base from a .dqs file by using Data Quality Client. This is particularly useful for backing up and restoring your knowledge bases when you are uninstalling Data Quality Server on a computer, and then reinstalling it on a different computer. You can easily export all the knowledge bases in an existing installation of Data Quality Server to a DQS backup file (.dqsb), and then import all the knowledge bases from the backup file after installing Data Quality Server on a different computer.

Exporting Knowledge Bases to .dqsb File

You can export all the knowledge bases in the existing Data Quality Server at any time or while uninstalling Data Quality Server.

- To export all the knowledge bases in a Data Quality Server to a DQS backup file (.dqsb), run DQSInstaller.exe with the `exportkbs` parameter from the command prompt, along with the full path and file name where you want to export the knowledge bases. For example, to export all the knowledge bases to the DQSBackup.dqsb file in the C: drive:

  ```
  dqsinstaller.exe -exportkbs c:\DQSBackup.dqsb
  ```

  > **NOTE**
  > If the provided file name already exists at the specified location, the installer prompts you whether to overwrite the file.

- To export all the knowledge bases to a DQS backup file while uninstalling Data Quality Server, run DQSInstaller.exe with the `uninstall` parameter from the command prompt, along with the full path and file name where you want to export the knowledge bases. For example, to export all the knowledge bases to the DQSBackup.dqsb file in the C: drive, and then uninstall Data Quality Server:

  ```
  dqsinstaller.exe -uninstall c:\DQSBackup.dqsb
  ```

> **NOTE**
>
> If you do not specify the full path and file name of the DQS backup file while uninstalling Data Quality Server from the command prompt using the `uninstall` parameter, a message is displayed stating that all the knowledge bases will be deleted, and allows you to choose whether to continue with the uninstall process.

Importing Knowledge Bases from .dqsb File

You can import all the knowledge bases from a DQS backup file (.dqsb) after completing the Data Quality Server installation. To import knowledge bases, you must have a valid DQS backup file (.dqsb).

Run the DQSInstaller.exe file with the `importkbs` parameter from the command prompt, along with the full path and file name from where you want to import the knowledge bases. For example, to import all the knowledge bases from the DQSBackup.dqsb file in the C: drive:

```
dqsinstaller.exe -importkbs c:\DQSBackup.dqsb
```

If there are existing knowledge bases in your Data Quality Server with the same name as the ones you are importing, the names of the imported knowledge bases will be appended with an underscore (_) followed by an integer value starting with 1. For example, if the "CompanyName" domain is duplicate, the imported domain name will be "CompanyName_1."

See Also

Run DQSInstaller.exe to Complete Data Quality Server Installation
Install Data Quality Services
Export a Knowledge Base to a .dqs File
Import a Knowledge Base from a .dqs File

Upgrade SQLCLR Assemblies After .NET Framework Update

5/9/2019 • 2 minutes to read • Edit Online

Data Quality Services (DQS) is a collection of SQL Common Language Runtime (SQLCR) routines that reference Microsoft .NET Framework 4 assemblies. When you install any .NET Framework updates on your computer that affect any such referenced .NET Framework assembly, it leads to a change in the Module Version ID (MVID) of the assembly in the Global Assembly Cache (GAC). This causes a mismatch between the MVIDs of the referenced assembly in GAC and the assembly in SQL Server 2014.

If the .NET Framework update requires you to restart the Data Quality Server computer, the affected SQLCLR assemblies are upgraded automatically to fix the MVID mismatch issue on restarting the Data Quality Server computer. However, for .NET Framework updates that do not require you to restart your Data Quality Server computer, an error occurs due to the mismatch in the MVIDs of the assemblies when you try to connect to a Data Quality Server using a Data Quality Client:

```
A new version of .NET was installed on this machine. In order to continue to work with DQS please run
dqsinstaller.exe -upgradedlls.
```

To fix this issue, the affected SQLCLR assemblies in SQL Server 2014 must be upgraded. You can do so by running the DQSInstaller.exe file with the **upgradedlls** command line parameter to skip recreating the DQS databases, and just upgrade the affected assemblies. This ensures that your knowledge bases, data quality projects, and any other data in DQS are preserved.

Prerequisites

- You must be logged on as a member of the Administrators group on the Data Quality Server computer.
- Your Windows user account must be a member of the sysadmin fixed server role in the SQL Server instance where Data Quality Server is installed.

To upgrade SQLCLR Assemblies

1. Start Command Prompt.
2. At the command prompt, change your directory to the location where DQSInstaller.exe is available. If you installed the default instance of SQL Server, the DQSInstaller.exe file will be available at C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn:

   ```
   cd C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn
   ```

3. At the command prompt, type the following command, and press ENTER:

   ```
   dqsinstaller.exe -upgradedlls
   ```

4. Rest of the steps are same as steps 2-6 in the Run DQSInstaller.exe from Start Screen, Start Menu or Windows Explorer section in Run DQSInstaller.exe to Complete Data Quality Server Installation.

See Also

Install Data Quality Services

Upgrade DQS Databases Schema After Installing SQL Server Update

Upgrade DQS Databases Schema After Installing SQL Server Update

5/9/2019 • 2 minutes to read • Edit Online

After you have installed a SQL Server update (patch, hotfix, or cumulative update) on a previously configured DQS instance, you might have to upgrade the DQS databases schema by running the DQSInstaller.exe file with the **upgrade** command line parameter. Otherwise, you might receive the following error while trying to connect to Data Quality Server using your Data Quality Client:

```
An error occurred in the Microsoft .NET Framework while trying to load assembly id 65581.
```

Upgrading DQS databases schema does not impact your existing data in the DQS databases (knowledge bases, data quality projects, and exported results in the DQS_STAGING_DATA database). However, you must back up your DQS databases before upgrading DQS databases schema to prevent any accidental data loss during the schema upgrade. For information about backing up DQS databases, see Backing Up and Restoring DQS Databases.

> **NOTE**
> Most of the SQL Server updates will require an upgrade to the DQS databases schema. For information about the SQL Server updates that will require an upgrade to the DQS databases schema, see the chart in step 1.A in Upgrade DQS: Installing Cumulative Updates or Hotfix Patches on Data Quality Services.

Prerequisites

- You must be logged on as a member of the Administrators group on the Data Quality Server computer.
- Your Windows user account must be a member of the sysadmin fixed server role in the SQL Server instance where Data Quality Server is installed.

To upgrade DQS databases schema

1. (Recommended) Back up your DQS databases before you proceed with the schema upgrade. For information about backing up DQS databases, see Backing Up and Restoring DQS Databases.
2. Start Command Prompt.
3. At the command prompt, change your directory to the location where DQSInstaller.exe is available. If you installed the default instance of SQL Server, the DQSInstaller.exe file will be available at C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn:

   ```
   cd C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn
   ```

4. At the command prompt, type the following command, and press ENTER:

   ```
   dqsinstaller.exe -upgrade
   ```

5. The installer prompts you for backing up the DQS databases before proceeding. If you have already backed up your DQS databases, type `Y` or `Yes` and press ENTER to continue with the upgrade.

6. A completion message is displayed after successful upgrade of the DQS databases schema.

Next Steps

Log on to the upgraded Data Quality Server from a Data Quality Client application.

For more information about upgrading DQS databases schema after installing SQL Server updates and associated troubleshooting steps, see Upgrade DQS: Installing Cumulative Updates or Hotfix Patches on Data Quality Services.

See Also

Install Data Quality Services
Upgrade SQLCLR Assemblies After .NET Framework Update

Install Integration Services

3/22/2019 • 7 minutes to read • Edit Online

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 Quick-Start Installation of SQL Server 2014. For information about command-line options for installing Integration Services, see Install SQL Server 2014 from the Command Prompt.

Preparing to Install Integration Services

Before you install Microsoft SQL Server Integration Services, review the following requirements:

- Hardware and Software Requirements for Installing SQL Server 2014
- Check Parameters for the System Configuration Checker
- Security Considerations for a SQL Server Installation

Selecting an Integration Services Configuration

You can install SQL ServerIntegration Services in the following configurations:

- You can install SQL ServerIntegration Services on a computer that has no previous instances of SQL Server.
- You can install SQL Server 2014 Integration Services (SSIS) side-by-side with an existing instance of SQL Server 2008 Integration Services (SSIS) and SQL Server 2012 Integration Services (SSIS).

  When you upgrade to SQL Server 2014 Integration Services (SSIS) on a machine that has one of these earlier versions of Integration Services already installed, SQL Server 2014 Integration Services (SSIS) is installed side-by-side with the earlier version.

  For more information on upgrading Integration Services, see Upgrade Integration Services. For information about backward compatibility with earlier versions of Integration Services, see Integration Services Backward Compatibility.

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.

NOTE

In previous versions of SQL Server, by default when you installed SQL Server all users in the Users group had access to the Integration Services service. When you install SQL Server 2014, users do not have access to the Integration Services service. The service is secure by default. After SQL Server is installed, the SQL Server administrator must run the DCOM Configuration tool (Dcomcnfg.exe) to grant specific users access to **SQL Server Integration Services 12.0**.

For instructions on how to grant permissions, see Grant Permissions to Integration Services Service.

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 are pages in the Setup Wizard where the options that you select affect your installation of Integration Services with selection recommendations:

- **Feature Selection**

  Select **Integration Services** 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 **Integration Services** and the following **Shared Features**:

  - **SQL Server Data Tools** to install the tools for designing packages.
  - **Management Tools - Complete** to install SQL Server Management Studio for managing packages.
  - **Client Tools SDK** to install managed assemblies for Integration Services programming.

  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 **SQL Server Data Tools** 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 SQL Server Data Tools. To ensure a complete installation of Integration Services, you must select **Integration Services** on the **Feature Selection** page.

  Installing on a 64-bit Computer On a 64-bit computer, selecting **Integration Services** installs only the 64-bit runtime and tools. If you have to run packages in 32-bit mode, you must also select an additional option to install the 32-bit runtime and tools:

  - If the 64-bit computer is running the x86 operating system, select **SQL Server Data Tools** or **Management Tools - Complete**.
  - If the 64-bit computer is running the Itanium operating system, select **Management Tools - Complete**.

  Installing on a Dedicated Server for ETL 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 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

- **Instance Configuration**

  Any selection that you make on the **Instance Configuration** 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.

  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 (SSIS Service).

- **Server Configuration**

  Review the settings for the Integration Services service on the **Service Accounts** tab of the **Server Configuration** page.

  If Windows 7 or Windows Server 2008 R2 is installed, the Integration Services service is registered to run under the NT Services\MsDtsServer120 virtual account, and the **Startup Type** is **Automatic**. You do not have to enter a password for the virtual account. If Microsoft Vista or Windows Server 2008 is installed, the Integration Services service is registered to run under the built-in Network Service account, and the **Startup Type** is **Automatic**. You do not have to enter a password for the built-in Network Service account.

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 2014. 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\120\SSIS node, and change the DWORD value of the LogPackageExecutionToEventLog setting from 0 to 1.

Understanding the Integration Services Service

Integration Services installs the Integration Services service.

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.

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 SQL Server Data Tools (SSDT).
- 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 SQL Server Data Tools (SSDT).

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.)

> **IMPORTANT**
> SQL Server Data Tools (SSDT), 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.

Interoperability and Coexistence (Integration Services)

6/14/2019 • 4 minutes to read • Edit Online

SQL Server 2014 Integration Services (SSIS) can co-exist side-by-side with SQL Server 2008 Integration Services and SQL Server 2012 Integration Services.

Features and Differences

The following table lists some of the differences between the current and earlier versions of Integration Services.

FEATURE	SQL SERVER 2014 INTEGRATION SERVICES (SSIS)	SQL SERVER 2012 INTEGRATION SERVICES (SSIS)	SQL SERVER 2008 INTEGRATION SERVICES (SSIS)
Development environment	Previous releases of SQL Server Data Tools (SSDT and SSDT-BI) SQL Server 2014 Data Tools - Business Intelligence for Visual Studio 2013	SQL Server Data Tools for Visual Studio 2010 SQL Server Data Tools - Business Intelligence for Visual Studio 2012	Business Intelligence Development Studio (Microsoft Visual Studio 2008)
Management environment	SQL Server Management Studio	SQL Server Management Studio	SQL Server Management Studio
Main system table in msdb for storing packages	sysssispackages	sysssispackages	sysssispackages
Main command prompt utility for running packages	**dtexec** (dtexec.exe), 2014 version	**dtexec** (dtexec.exe), 2012 version	**dtexec** (dtexec.exe), 2008 version
Default root file system folder	C:\Program Files\Microsoft SQL Server\120\DTS	C:\Program Files\Microsoft SQL Server\110\DTS	C:\Program Files\Microsoft SQL Server\100\DTS
Default root Registry key	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\120\SSIS	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\110\SSIS	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\100\SSIS

Side-by-Side Compatibility Issues

When you have SQL Server 2014 Integration Services installed side-by-side with SQL Server 2008 Integration Services and SQL Server 2012 Integration Services., you can perform the following tasks:

- **Design packages in SQL Server Data Tools**. You use the following tools to develop and maintain packages based on corresponding versions of SQL Server.
  - Use the SQL Server 2008 version of Business Intelligence Development Studio to develop and maintain packages that are based on SQL Server 2008 Integration Services (SSIS)
  - Use the SQL Server 2012 version of SQL Server Data Tools (SSDT) to develop and maintain packages that are based on SQL Server 2014 Integration Services (SSIS).

- Use the SQL Server 2014 version of SQL Server Data Tools (SSDT) to develop and maintain packages that are based on SQL Server 2014 Integration Services (SSIS).

- **Load and run packages**. You can load and run packages that were developed in SQL Server 2008 and SQL Server 2012, in the SQL Server 2014 version of SQL Server Data Tools (SSDT). When you add the package to an existing project, the package is permanently upgraded to the format that SQL Server 2014 Integration Services uses. When you open the package file in SQL Server Data Tools (SSDT), the package is temporarily upgraded to the format that SQL Server 2014 Integration Services uses. If you save the change to the package, the package is permanently upgraded. Once saved in the format that SQL Server 2014 Integration Services uses, packages can no longer be opened in the corresponding SQL Server 2008 or SQL Server 2012 version of Business Intelligence Development Studio, nor run by the corresponding SQL Server 2008 Integration Services or SQL Server 2012 Integration Services tools.

- **Manage packages in SQL Server Management Studio**. You cannot connect to an instance of the SQL Server 2014 version of the Integration Services service, from the SQL Server 2008 or the SQL Server 2012 version of Management Studio. You cannot connect to an instance of the SQL Server 2008 or the SQL Server 2012 version of the Integration Services service from the SQL Server 2014 version of Management Studio. You can use the SQL Server 2014 version of Management Studio to manage Integration Services packages that are stored in an instance of SQL Server 2012 or SQL Server 2014. You need to modify the service configuration file to add the instance of SQL Server 2012 to the list of locations managed by the service. For more information, see Configuring the Integration Services Service (SSIS Service).

- **Store packages in SQL Server**. You can store packages in the following databases.

PACKAGE FORMAT	DATABASE
SQL Server 2008 Integration Services	msdb database of an instance of SQL Server 2008
SQL Server 2012 Integration Services	msdb database of an instance of SQL Server 2012
SQL Server 2014 Integration Services	msdb database of an instance of SQL Server 2014

  On an instance of SQL Server 2014, you can import packages from an instance of SQL Server 2008 or from an instance of SQL Server 2012, but you cannot export packages to an instance of SQL Server 2008 or to an instance of SQL Server 2012.

  On an instance of SQL Server 2008 or an instance of SQL Server 2012, you cannot import packages from, nor export packages to, an instance of SQL Server 2014.

- **Run packages**. You can run SQL Server 2008 Integration Services and SQL Server 2012 Integration Services packages by using the SQL Server 2014 version of the **dtexec** utility or of SQL Server Agent. Whenever a SQL Server 2014 Integration Services tool loads a package that was developed in SQL Server 2008 or SQL Server 2012, the tool temporarily converts, in memory, the package to the package format that SQL Server 2014 Integration Services (SSIS) uses. If the SQL Server 2008 or SQL Server 2012 package has issues that prevent a successful conversion, the SQL Server 2014 Integration Services tool cannot run the package until those issues are resolved. For more information, see Upgrade Integration Services Packages.

Install Master Data Services

5/9/2019 • 4 minutes to read • Edit Online

The following workflow provides an overview of how to install and configure Master Data Services. Master Data Services installation is a three-part process:

- Pre-Installation Tasks: Verify system requirements before you install Master Data Services.
- Installation Operations: Install Master Data Services by using SQL Server Setup or the command prompt.
- Post-Installation Tasks: Open Master Data Services Configuration Manager to complete post-installation operations. Create and configure the Master Data Services database, Master Data Manager web application, and web services, and deploy a sample model.

Pre-Installation Tasks

ACTION	DETAILS	RELATED TOPICS
Verify installation requirements	The computer where you run SQL Server Setup must meet minimum requirements for: SQL Server Setup. The Master Data Manager web application and web services. The Master Data Services database, if you host the database on the same computer as the web application. Note that you can separate the web server computer and database server computer by running Setup on only the web server computer and creating the Master Data Services database on a remote computer that runs a supported version and edition of SQL Server.	Features Supported by the Editions of SQL Server 2014 Hardware and Software Requirements for Installing SQL Server 2014 Web Application Requirements (Master Data Services) Database Requirements (Master Data Services)
Configure the required roles, role services, and features	Before you run Setup, configure the computer with the required Windows roles, role services, and features. Note: Although you can perform this step later in the workflow, it is helpful to configure this prior to running Setup so that you can perform web configuration tasks immediately following the installation.	Web Application Requirements (Master Data Services)
Review language support considerations	Determine the language that you want to install and run Master Data Services in.	Multi-Lingual and Global Deployments (Master Data Services)

Installation Operations

ACTION	DETAILS	RELATED TOPICS
Run SQL Server Setup	On the computer that will host the Master Data Manager web application and Master Data Services web services, use SQL Server Setup or a command prompt to install Master Data Services. When you use SQL Server Setup, Master Data Services is available on the **Feature Selection** page under **Shared Features**. When you use a command prompt, Master Data Services is available as a feature parameter. Note that the command-line setup process installs Master Data Services, but does not configure it. You must configure it using the Master Data Services Configuration Manager. The installation process: Installs Master Data Services folders and files at the location you specify for shared features, and assigns permissions to these objects. Registers Master Data Services assemblies in the Global Assembly Cache (GAC). Installs Master Data Services Configuration Manager.	Install SQL Server 2014 from the Installation Wizard (Setup) Folder and File Permissions (Master Data Services)

Post-Installation Tasks

ACTION	DETAILS	RELATED TOPICS

ACTION	DETAILS	RELATED TOPICS
Open Master Data Services Configuration Manager to complete post-installation operations	After Setup completes, open Master Data Services Configuration Manager. Master Data Services Configuration Manager performs the following post-installation operations on the local computer: Creates a Windows group, **MDS_ServiceAccounts**, to contain Master Data Services service accounts for application pools. Under the Master Data Services installation path, creates the MDSTempDir folder and assigns permissions for **MDS_ServiceAccounts**. This folder is where temporary compilation files are compiled for the Master Data Manager web application. In the Master Data Services Web.config file, configures the `tempDirectory` attribute of the **<compilation>** element with the path to the MDSTempDir folder. If you script the installation process, you can open Master Data Services Configuration Manager to register the Master Data Services snap-in, but you must manually perform the other steps to complete the configuration. Master Data Services Configuration Manager provides you with a wizard-driven configuration process. There is no command-line process for configuring Master Data Services.	Folder and File Permissions (Master Data Services) Web Configuration Reference (Master Data Services)
Create a Master Data Services database	Use Master Data Services Configuration Manager to create a Master Data Services database for your master data.	Create a Master Data Services Database
Create a Master Data Manager web application	Use Master Data Services Configuration Manager to create and configure a web application to host Master Data Manager.	Create a Master Data Manager Web Application (Master Data Services)
Associate a Master Data Services database with a web application	Use Master Data Services Configuration Manager to associate your Master Data Manager web application with your Master Data Services database.	Associate a Master Data Services Database and Web Application

ACTION	DETAILS	RELATED TOPICS
Configure Internet Explorer Enhanced Security	When you install Master Data Services on a Windows Server 2008 or Windows Server 2008 R2 computer, you might have to configure Internet Explorer Enhanced Security to allow scripting for the Master Data Manager application site. Otherwise, browsing to the Master Data Manager application site on the server computer will fail.	Internet Explorer: Enhanced Security Configuration
Install the Master Data ServicesAdd-in for Excel	Users who will work with master data can install the Add-in for Excel.	https://go.microsoft.com/fwlink/?LinkId=219530
Enable Data Quality Services (DQS) integration	For users of the Master Data ServicesAdd-in for Excel, enable integration with the DQS feature, which can be used to match similar data.	Enable Data Quality Services Integration with Master Data Services
Deploy a sample model	Sample model packages are installed with Master Data Services, and can be deployed using MDSModelDeploy.exe.	Deploying MDS Samples in SQL Server

If you encounter issues during the installation process or initial configuration, see Troubleshooting Installation and Configuration Issues on TechNet Wiki.

If you no longer need Master Data Services on a computer, you can uninstall Master Data Services and determine whether to remove items that are not affected by the uninstall process. For more information, see Uninstall and Remove Master Data Services.

See Also

Install SQL Server 2014

Web Application Requirements (Master Data Services)

5/9/2019 • 4 minutes to read • Edit Online

Master Data Services is a web application hosted by Internet Information Services (IIS). Master Data Services works only in Internet Explorer (IE) 7 or later. IE 7 and earlier versions, Microsoft Edge and Chrome are not supported.

Use Master Data Services Configuration Manager to create and configure the Master Data Manager web application. Master Data Services Configuration Manager configures IIS on the local computer, so it is best for initial web configuration tasks. For example, configure a Master Data Services environment with a single Master Data Manager web application, or configure the first web application in a scale-out deployment of Master Data Services. Use IIS tools to perform more complex tasks, such as configuring multiple web servers in a scale-out deployment.

> **NOTE**
> Any computer on which you install components of Master Data Services must be licensed. For more information, refer to the End User License Agreement (EULA).

Requirements

Operating System

The following Windows operating systems include the Internet Information Services (IIS) functionality required for the Master Data Services web application and web service.

SQL SERVER 2014 DEVELOPER (64-BIT) X64	SQL SERVER 2014 ENTERPRISE (64-BIT) X64	SQL SERVER 2014 BUSINESS INTELLIGENCE (64-BIT) X64
Windows Server 2008 SP2	Windows Server 2008 SP2	Windows Server 2008 SP2
Windows Server 2008 R2 SP1	Windows Server 2008 R2 SP1	Windows Server 2008 R2 SP1
Windows 7 Professional, Enterprise, and Ultimate	Windows Server 2012	Windows Server 2012
Windows 8.0 Professional, Enterprise, and Ultimate		

For a full list of the Windows operating systems that are supported for your edition of SQL Server, see Hardware and Software Requirements for Installing SQL Server 2014.

Microsoft Silverlight

To work in the Master Data Manager web application, Silverlight 5 must be installed on the client computer. If you do not have the required version of Silverlight, you will be prompted to install it when you navigate to an area of the web application that requires it. You can install Silverlight 5 from here.

Role and Role Services (Windows Server 2008 or Windows Server 2008 R2, Windows 7 operating systems)

On Windows Server 2008 R2, you can use **Server Manager**, which is available in the Microsoft Management Console (MMC), to install the **Web Server (IIS)** role and the following required role services.

> **NOTE**
> On Windows Vista and Windows 7 operating systems, use **Programs and Features** in Control Panel to enable these options in the **Windows Features** dialog box.

Web Server
Common HTTP Features
Static Content
Default Document
Directory Browsing
HTTP Errors
Application Development
ASP.NET
.NET Extensibility
ISAPI Extensions
ISAPI Filters
Health and Diagnostics
HTTP Logging
Request Monitor
Security
Windows Authentication
Request Filtering
Performance
Static Content Compression
Management Tools
IIS Management Console

Role and Role Services (Windows Server 2012 or Windows 8 operating systems)

On Windows Server 2012, you can use **Server Manager**, which is available in the Microsoft Management Console (MMC), to install the **Web Server (IIS)** role and the following required role services.

> **NOTE**
> On the Windows 8 operating system, use **Programs and Features** in Control Panel to enable these options in the **Windows Features** dialog box.

Internet Information Services
Web Management Tools
IIS Management Console
World Wide Web Services
Application Development
.NET Extensibility 3.5
.NET Extensibility 4.5
ASP.NET 3.5
ASP.NET 4.5
ISAPI Extensions
ISAPI Filters
Common HTTP Features
Default Document
Directory Browsing
HTTP Errors
Static Content
[Note: Do not install WebDAV Publishing]
Health and Diagnostics
HTTP Logging
Request Monitor
Performance
Static Content Compression
Security
Request Filtering
Windows Authentication

Features (Windows Server 2008 or Windows Server 2008 R2, Windows 7 operating systems)

On Windows Server 2008 or Windows Server 2008 R2, you can use **Server Manager** to install the following required features.

> **NOTE**
> On Windows Vista and Windows 7 operating systems, use **Programs and Features** in Control Panel to enable these options in the **Windows Features** dialog box.

.NET Framework 3.0 Features
WCF Activation
HTTP Activation
Non-HTTP Activation
Windows Process Activation Service
Process Model
.NET Environment
Configuration APIs

Features (Windows Server 2012 or Windows 8 operating systems)

On Windows Server 2012, you can use **Server Manager** to install the following required features.

> **NOTE**
> On the Windows 8 operating system, use **Programs and Features** in Control Panel to enable these options in the **Windows Features** dialog box.

.NET Framework 3.5 (includes .NET 2.0 and 3.0)
.NET Framework 4.5 Advanced Services
ASP.NET 4.5
WCF Services
HTTP Activation [Note: This is required.]
TCP Port Sharing
Windows Process Activation Service
Process Model
.NET Environment
Configuration APIs

Accounts and Permissions

TYPE	DESCRIPTION
Windows account	You must log on to the web server computer with a Windows account that has permission to configure Windows roles, role services, and features, and to create and manage application pools, web sites, and web applications in IIS on the local computer.

TYPE	DESCRIPTION
Service account	When you create the Master Data Manager web application in Master Data Services Configuration Manager, you must specify an identity for the application pool that the application runs in. This account can be different from the service account that was specified when the Master Data Services database was created. This identity must be a domain user account, and it is added to the mds_exec database role in the Master Data Services database for database access. For more information, see Database Logins, Users, and Roles (Master Data Services). This account is also added to a Master Data Services Windows group, **MDS_ServiceAccounts**, which is granted permission to the temporary compilation directory, **MDSTempDir**, in the file system. For more information, see Folder and File Permissions (Master Data Services). The application pool account needs the VIEW SERVER STATE permission, to avoid server errors. For example, the MDS Validate Version command fails with a server error. For more information, see MDS Validate Version command fails with a server error in SQL Server 2012 and SQL Server 2014

See Also

Install Master Data Services
Create a Master Data Manager Web Application (Master Data Services)
Web Configuration Page (Master Data Services Configuration Manager)

Database Requirements (Master Data Services)

5/9/2019 • 2 minutes to read • Edit Online

All master data is stored in a Master Data Services database. The computer that hosts this database must run an instance of SQL Server Database Engine.

Use Master Data Services Configuration Manager to create and configure the Master Data Services database on either a local or a remote computer. If you move the database from one environment to another, you can maintain the information in a new environment by associating the Master Data Services web service and Master Data Manager to the database in its new location.

> **NOTE**
> Any computer on which you install components of Master Data Services must be licensed. For more information, refer to the End User License Agreement (EULA).

Requirements

Before you create a Master Data Services database, ensure the following requirements are met.

SQL Server Edition

The Master Data Services database can be hosted on the following editions of SQL Server:

- SQL Server 2014 Business Intelligence (64-bit) x64
- SQL Server 2014 Enterprise (64-bit) x64
- SQL Server 2014 Developer (64-bit) x64
- SQL Server 2012 Business Intelligence (64-bit) x64
- SQL Server 2012 Enterprise (64-bit) x64 - Upgrade from SQL Server 2008 R2 Enterprise only
- SQL Server 2012 Developer (64-bit) x64
- Microsoft SQL Server 2008 R2 Enterprise (64-bit) x64
- Microsoft SQL Server 2008 R2 Developer (64-bit) x64

For a list of features that are supported by the editions of SQL Server, see Features Supported by the Editions of SQL Server 2014.

Operating System

For information about the supported Windows operating systems and other requirements for SQL Server Database Engine, see Hardware and Software Requirements for Installing SQL Server 2014.

Accounts and Permissions

TYPE	DESCRIPTION

TYPE	DESCRIPTION
User account	In Master Data Services Configuration Manager, you can use a Windows account or a SQL Server account to connect to the Database Engine instance of SQL Server to host the Master Data Services database. The user account must belong to the **sysadmin** server role on the instance of SQL Server Database Engine. For more information about the **sysadmin** role, see Server-Level Roles.
Master Data Manager administrator account	When you create a Master Data Services database, you must specify a domain user account to be the Master Data Services system administrator. For all Master Data Manager web applications associated with this database, this user can update all models and all data in all functional areas. For more information, see Administrators (Master Data Services).

Database Backup

As a best practice, back up the full database daily at a time of low activity and back up transaction logs more frequently depending on the needs of your environment. For more information about database backups, see Backup Overview (SQL Server).

See Also

Install Master Data Services
Create a Master Data Services Database
Master Data Services Database
Connect to a Master Data Services Database Dialog Box
Create Database Wizard (Master Data Services Configuration Manager)

Create a Master Data Services Database

5/9/2019 • 2 minutes to read • Edit Online

Create a Master Data Services database when you need a new database to support the Master Data Manager web application and Master Data Services web service.

Prerequisites

- For information about the requirements for the computer that hosts the database, see Database Requirements (Master Data Services).

To create a Master Data Services database

1. Open Master Data Services Configuration Manager.
2. In the left pane, click **Database Configuration**.
3. On the **Database Configuration** page, click **Create Database**.
4. Complete the **Create Database** wizard to create and configure the database. For information about the user interface (UI) options in the wizard, see Create Database Wizard (Master Data Services Configuration Manager).

Next Steps

- Configure system settings for the database and web application. For more information, see System Settings (Master Data Services).
- Create a Master Data Manager web application to associate with this database. For more information, see Create a Master Data Manager Web Application (Master Data Services).
- Configure a maintenance plan to back up the database and transaction logs. For more information, see Database Requirements (Master Data Services).

See Also

Install Master Data Services

Create a Master Data Manager Web Application (Master Data Services)

5/9/2019 • 2 minutes to read • Edit Online

The Master Data Manager web application provides an interface for users to work with master data and for administrators to configure and administer MDS.

A web application must always be contained by a website. To create a web application, you must either:

- Use the Default website and then create the web application,
- Use an existing website and then create the web application, or
- Create a new website, which automatically creates a web application.

After you create the web application, you associate it with the Master Data Services database.

Prerequisites

- For information about the requirements for the computer that hosts the web application, see Web Application Requirements (Master Data Services).

To create a Master Data Manager web application in a new website

When you create a new website, the root web application is the Master Data Manager web application. The web application is also added to a new application pool.

> **NOTE**
> If you follow this procedure, you cannot specify a virtual path and alias of the Master Data Manager web application. If you want to specify a virtual path and alias for Master Data Manager, you must create a web application in an existing website that is not already configured as a Master Data Manager web application.

Additionally, Master Data Services Configuration Manager supports creating sites with HTTP bindings only. To add an HTTPS binding, create a new site and application in Master Data Services Configuration Manager and then see Secure a Master Data Manager Web Application for more information.

To create a Master Data Manager web application in a new website

1. Open Master Data Services Configuration Manager.
2. In the left pane, click **Web Configuration**.
3. On the **Web Configuration** page, in the Website list, select **Create new website**.
4. On the **Create Website** dialog box, specify information for a new website. For more information about the user interface (UI) options in the dialog box, see Create Website Dialog Box (Master Data Services Configuration Manager).
5. Click **OK**.

To create a Master Data Manager web application in an existing website

When you create a web application in an existing website, you can choose the virtual path and alias of the web application. The web application is added to a new application pool.

To create a Master Data Manager web application in an existing website

1. Open Master Data Services Configuration Manager.
2. In the left pane, click **Web Configuration**.
3. On the **Web Configuration** page, from the **Website** list, select the website in which you want to create the Master Data Manager web application.
4. Click **Create Application**.
5. On the **Create Web Application** dialog box, specify information for a new web application. For more information about the user interface (UI) options in the dialog box, see Create Web Application Dialog Box (Master Data Services Configuration Manager).
6. Click **OK**.

Next Steps

- Associate the web application with a Master Data Services database. For more information, see Associate a Master Data Services Database and Web Application.
- Optionally, configure the website that hosts the Master Data Manager web application to use an HTTPS binding if you want to encrypt content by using Secure Sockets Layer (SSL). You must use an Internet Information Services (IIS) tool, such as IIS Manager, to configure the server certificate for the web server, and to configure an HTTPS binding and the SSL settings for the site. For more information, see Secure a Master Data Manager Web Application.

See Also

Install Master Data Services

Secure a Master Data Manager Web Application

5/9/2019 • 2 minutes to read • Edit Online

You can secure the Master Data Manager web application with HTTPS.

> **NOTE**
> The Master Data Manager web application can use either HTTP or HTTPS, but not both.

Prerequisites

To perform the procedure:

- You must be an administrator on the web server where Master Data Manager is installed.
- MDS must be installed on the web server, and a web application must exist. For more information, see Install Master Data Services and Create a Master Data Manager Web Application (Master Data Services).

To secure the Master Data Manager web application with HTTPS

1. After you have confirmed that the Master Data Manager web application is configured correctly with HTTP, create a certificate in IIS. For more information, see Configuring Server Certificates in IIS 7.
2. In the **Connections** pane, under **Sites**, click the site that hosts the Master Data Manager web application.
3. In the **Actions** pane, click **Bindings**.
4. Click **Add**.
5. From the list, select **https**.
6. Select the SSL certificate.
7. Click **OK**.
8. Optional. To remove HTTP so that users can access the site with HTTPS only, from the list, click the row with **http**. Click **Remove** and on the confirmation dialog box, click **Yes**.

   > **IMPORTANT**
   > You must change basicHttp and wsHttpBinding configurations after removing HTTP.

9. To close the **Site Bindings** dialog box, click **Close**.
10. Now open the web.config file from *drive*:\Program Files\Microsoft SQL Server\120\Master Data Services\WebApplication.
11. Find the string `<security mode="Message">` and change it to `<security mode="Transport">`.
12. Save and close the file. If you get an error, it could be because you have UAC enabled. For more information, see Turn off User Account Control. Users should now be able to use HTTPS to access the site.

See Also

Create a Master Data Manager Web Application (Master Data Services)

Associate a Master Data Services Database and Web Application

5/9/2019 • 2 minutes to read • Edit Online

Associate your Master Data Manager web application with a Master Data Services database to specify the database to use for web operations.

Prerequisites

- Master Data Services Configuration Manager must be installed on the local computer. For more information, see Install Master Data Services.
- A local Master Data Manager web application must exist. For more information, see Create a Master Data Manager Web Application (Master Data Services).
- Either a local or remote Master Data Services database must exist. For more information, see Create a Master Data Services Database.

To associate a Master Data Services database and web application

1. Open Master Data Services Configuration Manager.
2. In the left pane, click **Web Configuration**.
3. On the **Web Configuration** page, under **Web application**, from the **Website** list, select the website that contains your Master Data Manager web application.
4. In the **Web application** box, select the web application that hosts Master Data Manager.
5. Under **Associate Application with Database**, click **Select**. The **Connect to Database** dialog box opens.
6. Specify connection information for the instance of SQL Server that hosts the Master Data Services database, and click **Connect**.
7. From the **Master Data Services database** list, select the database you want to associate the web application with and then click **OK**.
8. Under **Associate Application with Database**, verify that the instance and database information are correct, and then click **Apply**.

Next Steps

- Programmatic access to Master Data Services web services is automatically enabled when the web application is created. To let developers access the service metadata to easily generate proxy classes for programmatic access, enable metadata publishing. For more information, see Create Master Data Manager Web Service Proxy Classes.
- Add users and groups to Master Data Manager. If no users or groups have been granted access to Master Data Manager, you must open Master Data Manager using the Master Data Services system administrator credentials. For more information, see Administrators (Master Data Services) and Users and Groups (Master Data Services).

See Also

Install Master Data Services
Web Configuration Page (Master Data Services Configuration Manager)

Enable Data Quality Services Integration with Master Data Services

5/9/2019 • 2 minutes to read • Edit Online

In the Master Data ServicesAdd-in for Excel, matching functionality is provided by Data Quality Services (DQS). This functionality must be enabled to be used.

Prerequisites

- A Master Data Services web application and database must exist.
- The Data Quality Services feature and the Data Quality Client must be installed on the same SQL Server instance that hosts the MDS database. For more information, see Install Data Quality Services.

To enable Data Quality Services integration

1. Open Master Data Services Configuration Manager.
2. In the left pane, click **Web Configuration**.
3. On the **Web Configuration** page, select the website and web application.
4. In the **Enable DQS Integration** section, click **Enable integration with Data Quality Services**.
5. On the confirmation dialog box, click **OK**.

See Also

Data Quality Matching in the MDS Add-in for Excel
Master Data Services Add-in for Microsoft Excel
Install Master Data Services

Multi-Lingual and Global Deployments (Master Data Services)

5/9/2019 • 2 minutes to read • Edit Online

Master Data Services supports deployment of components and tools in all languages supported by SQL Server. For more information, see Local Language Versions in SQL Server.

How languages are used

The following table describes the language support for the Master Data Services components and tools.

COMPONENT OR TOOL	DESCRIPTION
Master Data Services Setup	Select the English Setup program when you want the Master Data Manager web application to be available and supported in languages that differ from the Setup language. For more information, see the Master Data Manager description below.
Master Data Services Configuration Manager	The Setup language determines the Master Data Services Configuration Manager language. For example, if you choose German for the Setup language, Master Data Services Configuration Manager is available in German on that computer.
Master Data Manager	When you run Setup in English, the Master Data Manager web application is available and supported in all application languages. Master Data Manager can display in any of those application languages and accept locale-specific input based on the language preferences of the client web browser. If the language preferences are configured for a non-supported application language, Master Data Manager defaults to English. When you run Setup in a language other than English, resources are included for the all other application languages but it is not a supported scenario for clients to use Master Data Manager in a language other than the selected Setup language. If you try to access Master Data Manager in a language different from the Setup language, you might experience problems with data display and input in the application.
Master Data Services database	Information in the Master Data Services database is not specific to any locale. This enables Master Data Manager to determine how to display information, such as dates and numbers, in the format determined by the language preferences of the client web browser.

See Also

Install Master Data Services

Install Reporting Services Native Mode Report Server

2/28/2020 • 6 minutes to read • Edit Online

A Reporting Services native mode report server can be installed from the SQL Server installation wizard or from the command line. In the setup wizard, you can select to either 1) install files and configure the server with default settings or to 1) only install the files and the server is not configured by the installation wizard. This topic reviews the *Default configuration for native mode* 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 server mode.

Applies to: Reporting Services Native mode

In this topic

- What is the Default Configuration?
- When to Install the Default Configuration for Native Mode
- Requirements
- Default URL Reservations
- Install Native Mode with the SQL Server Installation Wizard
- Install Native Mode with the Command Line

What is the Default Configuration?

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)
- The Reporting Services Configuration Manager
- 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 SQL Server Data Tools (SSDT), 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 Manager to configure these properties. For more information, see Reporting Services Configuration Manager (Native Mode).

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.

Requirements

The default configuration option uses default values to configure the core settings required to make a report server operational. It has the following requirements:

- You hardware should meet minimum hardware and software requirements for running Microsoft SQL Server. For more information, see Hardware and Software Requirements for Installing SQL Server 2014.
- 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 **ReportServer** and **ReportServerTempDB**. 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 Manager 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.

Default URL Reservations

URL reservations are composed of a prefix, host name, port, and virtual directory:

PART	DESCRIPTION

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 http://<IPAddress>/reportserver.
Port	The default port is 80. 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_<*instance_name*> for the Report Server Web service and Reports_<*instance_name*> for Report Manager. For the Report Server Web service, the default virtual directory is **reportserver**. For Report Manager, the default virtual directory is **reports**.

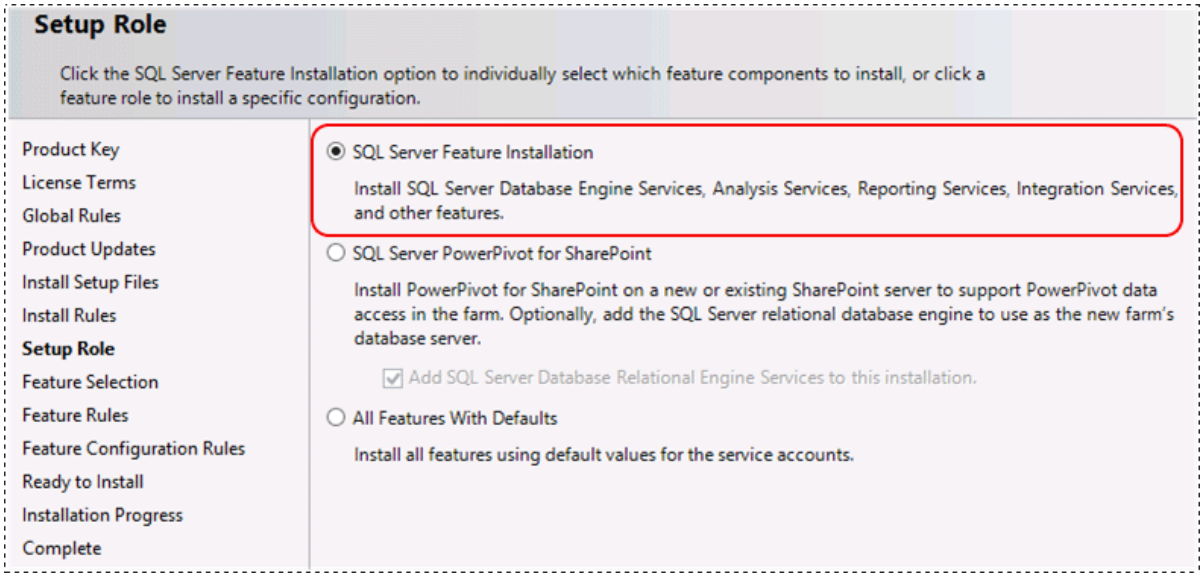
An example of the complete URL string might be as follows:

- http://+:80/reportserver, provides access to the report server.
- http://+:80/reports, provides access to Report Manager.

Install Native Mode with the SQL Server Installation Wizard

The following list describes the Reporting Services specific steps and options you select in the SQL Server Installation Wizard. The list does not described each page you will see in the installation wizard, only the Reporting Services related pages that are part of a Native mode installation.

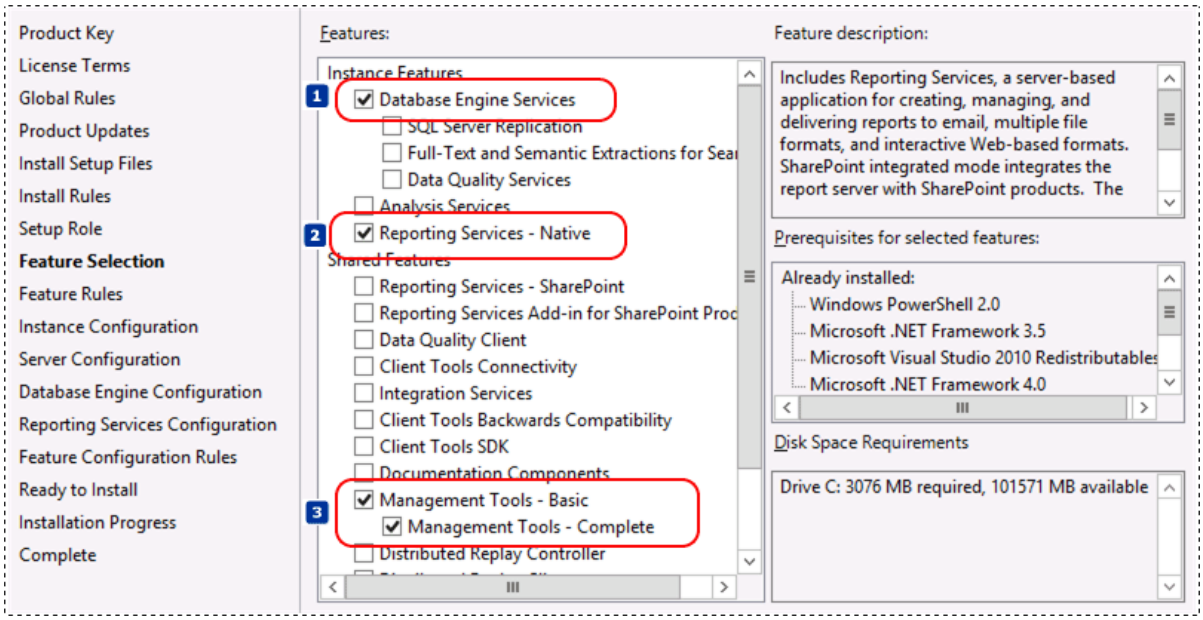
1. On the **Setup Role** page, Select **SQL Server Feature Installation**.



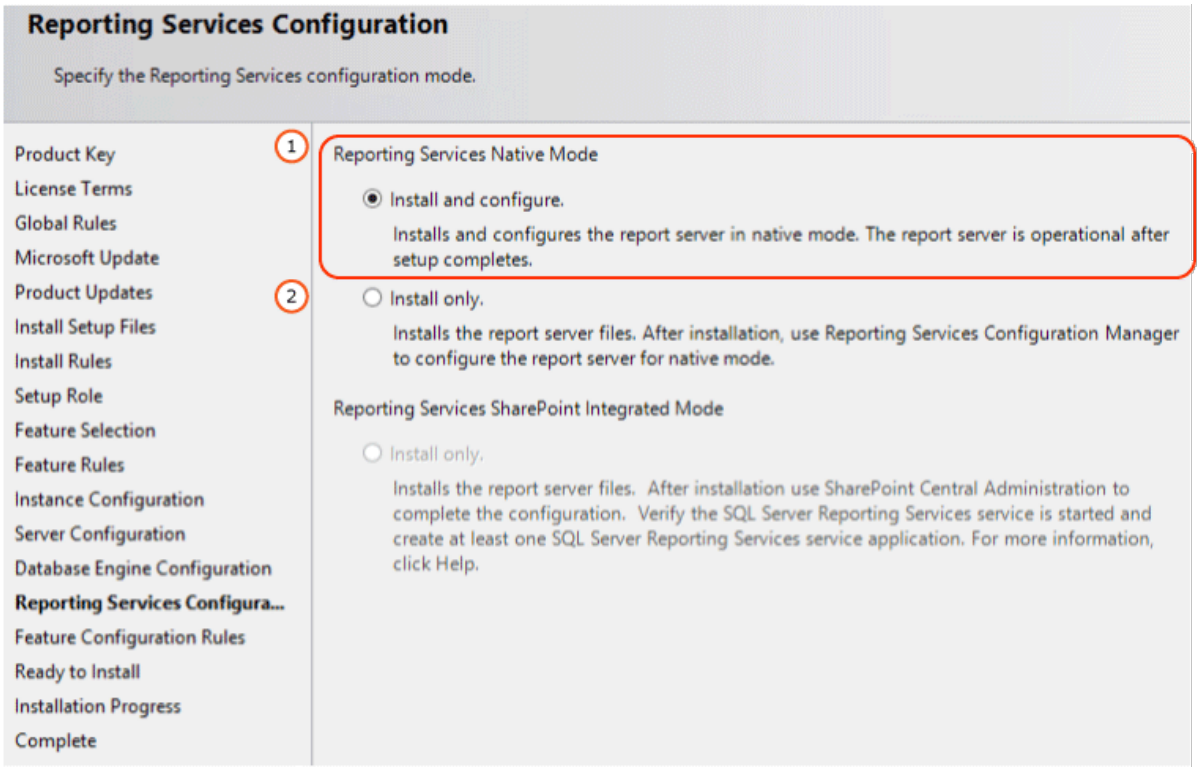
2. On the **Feature Selection** page, select the following:

   - **Database Engine Services**, unless an instance of the database engine is already installed.

- **Reporting Services-Native**.
- **Management Tools - Basic**. The management tools are not required but it is recommended unless you have some other installation of management tools. The default configuration option will result a functioning report server but you may want to change configuration options at a later date. Some options such as 'My Reports' are managed through Management Studio



3. If you plan to use the Reporting Services subscription feature, then on the **Server Configuration** page, you want to verify SQL Server Agent is configured for **Automatic** Startup type.
4. On the **Reporting Services Configuration** page select **Install and Configure**.



5. After the SQL Server installation wizard is complete, verify the default Native mode installation using the following basic steps.

- Open Reporting Services Configuration Manager and confirm you can connect to the report server.
- Open your browser with administrative privileges and connect to Reporting Services Report Manager, for example `http://loclahost/Reports`.

- Open your browser with administrative privileges and connect to the Reporting Services report server page. For example, `http://loclahost/ReportServer`

For more information, see the Native section of the following two topics:

Verify a Reporting Services Installation

Troubleshoot a Reporting Services Installation

Install Native Mode with the Command Line

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=MSSQLSERVER
/SQLSYSADMINACCOUNTS="BUILTIN\ADMINISTRATORS"
/RSSVCACCOUNT="NT AUTHORITY\NETWORK SERVICE" /SQLSVCACCOUNT="NT AUTHORITY\NETWORK SERVICE" /AGTSVCACCOUNT="NT
AUTHORITY\NETWORK
SERVICE" /RSSVCSTARTUPTYPE="Manual" /RSINSTALLMODE="DefaultNativeMode"
```

For more information and examples, see Command Prompt Installation of Reporting Services SharePoint Mode and Native Mode and Install SQL Server 2014 from the Command Prompt

See Also

Troubleshoot a Reporting Services Installation Verify a Reporting Services Installation Configure the Report Server Service Account (SSRS Configuration Manager) Configure Report Server URLs (SSRS Configuration Manager) Configure a Report Server Database Connection (SSRS Configuration Manager) Files-Only Installation (Reporting Services) Initialize a Report Server (SSRS Configuration Manager) Configure SSL Connections on a Native Mode Report Server Configure Report Server URLs (SSRS Configuration Manager) Configure Windows Service Accounts and Permissions Quick-Start Installation of SQL Server 2014

Files-Only Installation (Reporting Services)

5/22/2019 • 2 minutes to read • Edit Online

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, 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 SQL Server Data Tools (SSDT), 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 files-only 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 Manager (Native Mode).

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.

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:

- Install SQL Server 2014 from the Installation Wizard (Setup).
- Install SQL Server 2014 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 /RSSVCACCOUNT="NT AUTHORITY\NETWORK SERVICE"
/RSINSTALLMODE="FilesOnlyMode"
```

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

Verify a Reporting Services Installation
Configure the Report Server Service Account (SSRS Configuration Manager)
Configure Report Server URLs (SSRS Configuration Manager)
Configure a Report Server Database Connection (SSRS Configuration Manager)
Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013)
Install Reporting Services Native Mode Report Server
Reporting Services Tools

Install Reporting Services and Internet Information Services Side-by-Side (SSRS Native Mode)

6/20/2019 • 4 minutes to read • Edit Online

You can install and run SQL Server 2014 Reporting Services (SSRS) and Internet Information Services (IIS) on the same computer. The version of IIS that you are using determines the interoperability issues you must address.

Applies to: Reporting Services Native mode

IIS VERSION	ISSUES	DESCRIPTION
IIS 6.0, 7.0, 8.0, 8.5	Requests intended for one application are accepted by a different application. HTTP.SYS enforces precedence rules for URL reservations. Requests that are sent to applications that have the same virtual directory name and that jointly monitor port 80 might not reach the intended target if the URL reservation is weak relative to the URL reservation of another application.	Under certain conditions, a registered endpoint that supersedes another URL endpoint in the URL reservation scheme might receive HTTP requests intended for the other application. Using unique virtual directory names for the Report Server Web service and Report Manager helps you avoid this conflict. Detailed information about this scenario is provided in this topic.

Precedence Rules for URL Reservations

Before you can address interoperability issues between IIS and Reporting Services, you must understand URL reservation precedence rules. Precedence rules can be generalized into the following statement: a URL reservation that has more explicitly defined values is first in line to receive requests that match the URL.

- A URL reservation that specifies a virtual directory is more explicit than one that omits a virtual directory.
- A URL reservation that specifies a single address (by way of an IP address, a fully qualified domain name, a network computer name, or a host name) is more explicit than a wildcard.
- A URL reservation that specifies a strong wildcard is more explicit than a weak wildcard.

The following examples show a range of URL reservations, ordered from most explicit to least explicit:

EXAMPLE	REQUEST
http://123.234.345.456:80/reports	Receives all requests that are sent to http://123.234.345.456/reports or http://<computername>/reports if a domain name service can resolve the IP address to that host name.
http://+:80/reports	Receives any requests that are sent to any IP address or host name that is valid for that computer as long as the URL contains the "reports" virtual directory name.

EXAMPLE	REQUEST
http://123.234.345.456:80	Receives any request that specifies http://123.234.345.456 or http://<computername> if a domain name service can resolve the IP address to that host name.
http://+:80	Receives requests that are not already received by other applications, for any application endpoints that are mapped to **All Assigned**.
http://*:80	Receives requests that are not already received by other applications, for application endpoints that are mapped to **All Unassigned**.

One indication of a port conflict is that you will see the following error message: 'System.IO.FileLoadException: The process cannot access the file because it is being used by another process. (Exception from HRESULT: 0x80070020).'

URL Reservations for IIS 6.0, 7.0, 8.0, 8.5 with SQL Server 2014 Reporting Services

Given the precedence rules outlined in the previous section, you can begin to understand how URL reservations defined for Reporting Services and IIS promote interoperability. Reporting Services receives requests that explicitly specify the virtual directory names for its applications; IIS receives all remaining requests, which can then be directed to applications that run within the IIS process model.

APPLICATION	URL RESERVATION	DESCRIPTION	REQUEST RECEIPT
Report Server	http://+:80/ReportServer	Strong wildcard on port 80, with report server virtual directory.	Receives all requests on port 80 that specify the report server virtual directory. The Report Server Web service receives all requests to http://<computername>/reportserver.
Report Manager	http://+:80/Reports	Strong wildcard on port 80, with Reports virtual directory.	Receives all requests on port 80 that specify the reports virtual directory. Report Manager receives all requests to http://<computername>/reports.
IIS	http://*:80/	Weak wildcard on port 80.	Receives any remaining requests on port 80 that are not received by another application.

Side-by-Side Deployments of SQL Server 2014 and SQL Server 2005 Reporting Services on IIS 6.0, 7.0, 8.0, 8.5

Interoperability issues between IIS and Reporting Services occur when IIS Web sites have virtual directory names that are identical to those used by Reporting Services. For example, suppose you have the following configuration:

- A Web site in IIS that is assigned to port 80 and a virtual directory named "Reports".

- A SQL Server 2014 report server instance installed in the default configuration, where the URL reservation also specifies port 80 and the Report Manager application also uses "Reports" for the virtual directory name.

Given this configuration, a request that is sent to http://<computername>:80/reports will be received by Report Manager. The application that is accessed through the Reports virtual directory in IIS will no longer receive requests after SQL Server 2014 report server instance is installed.

If you are running side-by-side deployments of older and newer versions of Reporting Services, you are likely to encounter the routing problem just described. This is because all versions of Reporting Services use "ReportServer" and "Reports" as virtual directory names for the report server and Report Manager applications, increasing the likelihood that you will have a "reports" and "reportserver" virtual directories in IIS.

To ensure that all applications receive requests, follow these guidelines:

- For Reporting Services installations, use virtual directory names that are not already used by an IIS Web site on the same port as Reporting Services. If there is a conflict, install Reporting Services in "files-only" mode (using the Install but do not configure the server option in the Installation Wizard) so that you can configure the virtual directories after Setup is finished. One indication that your configuration has a conflict is you will see the error message: System.IO.FileLoadException: The process cannot access the file because it is being used by another process. (Exception from HRESULT: 0x80070020).
- For installations that you configure manually, adopt the default naming conventions in the URLs that configure. If you install SQL Server 2014 Reporting Services (SSRS) as a named instance, include the instance name when creating a virtual directory.

See Also

Configure Report Server URLs (SSRS Configuration Manager)
Configure a URL (SSRS Configuration Manager)
Install Reporting Services Native Mode Report Server

Host a Report Server Database in a SQL Server Failover Cluster

5/22/2019 • 2 minutes to read • Edit Online

SQL Server provides failover clustering support so that you can use multiple disks for one or more SQL Server instances. Failover clustering is supported only for the report server database; you cannot run the Report Server service as part of a failover cluster.

To host a report server database on a SQL Server failover cluster, the cluster must already be installed and configured. You can then select the failover cluster as the server name when you create the report server database in the Database Setup page of the Reporting Services Configuration tool.

Although the Report Server service cannot participate in a failover cluster, you can install Reporting Services on a computer that has a SQL Server failover cluster installed. The report server runs independently of the failover cluster. If you install a report server on a computer that is part of a SQL Server failover instance, you are not required to use the failover cluster for the report server database; you can use a different SQL Server instance to host the database.

See Also

Report Server Database (SSRS Native Mode)
Create a Report Server Database (SSRS Configuration Manager)

Reporting Services Configuration Manager (Native Mode)

10/3/2019 • 6 minutes to read • Edit Online

Use the Reporting Services Configuration Manager to configure a Reporting Services Native Mode installation. If you installed a report server by using the files-only installation option, you must use the Configuration Manager to configure the server before you can use it. If you installed a report server by using the default configuration installation option, you can use the Configuration Manager to verify or modify the settings that were specified during setup. Reporting Services Configuration Manager can be used to configure a local or remote report server instance.

Applies to: Reporting Services Native mode.

> **NOTE**
> Starting with the SQL Server 2012 release, the Reporting Services Configuration Manager is not designed to manage SharePoint mode report servers. SharePoint mode is managed and configured by using SharePoint Central Administration and PowerShell scripts. For information, see Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013).

In this section:

Configure the Report Server Service Account (SSRS Configuration Manager)
Provides recommendations and steps on how to modify the service account and password.

Configure Report Server URLs (SSRS Configuration Manager)
Describes how to configure URLs used to access the Report Server Web service and Report Manager.

Create a Report Server Database (SSRS Configuration Manager)
Describes how to create a report server database, required for storing server metadata and objects.

Configure a Report Server Database Connection (SSRS Configuration Manager)
Describes how to modify the connection string used by the report server to connect to the report server database.

Configure the Unattended Execution Account (SSRS Configuration Manager)
Describes how to configure a user account to process reports in unattended mode.

Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager)
Describes how to configure a report server to support e-mail report distribution.

Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager)
Provides information about configuring multiple report server instances to use a shared report server database.

Configure and Manage Encryption Keys (SSRS Configuration Manager)
Explains how to use and manage encryption keys that are used when storing sensitive data.

Manage a Reporting Services Native Mode Report Server
Provides step-by-step instruction for common tasks.

Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Provides help topics for the pages in the Reporting Services Configuration tool.

In this topic:

- Scenarios to use Reporting Services Configuration Manager
- Requirements
- To Start the Reporting Services Configuration Manager

Scenarios to use Reporting Services Configuration Manager

You can use the Reporting Services Configuration Manager to perform the following tasks:

- Configure the Report Server service account. The account is initially configured during setup, but can be modified by using the Reporting Services Configuration Manager if you update the password or want to use a different account.
- Create and configure URLs. The report server and Report Manager are ASP.NET applications accessed through URLs. The report server URL provides access to the SOAP endpoints of the report server. The Report Manager URL is used to open Report Manager. You can configure a single URL or multiple URLs for each application.
- Create and configure the report server database. The report server is a stateless server that requires a SQL Server database for internal storage. You can use the Reporting Services Configuration Manager to create and configure a connection to the report server database. You can also select an existing report server database that already contains the content you want to use.
- Configure a Native mode scale-out deployment. Reporting Services supports a deployment topology that allows multiple report server instances use a single, shared report server database. To deploy a report server scale-out deployment, you use the Reporting Services Configuration Manager to connect each report server to the shared report server database.
- Backup, restore, or replace the symmetric key that is used to encrypt stored connection strings and credentials. You must have a backup of the symmetric key if you change the service account, or move a report server database to another computer.
- Configure the unattended execution account. This account is used for remote connections during scheduled operations or when user credentials are not available.
- Configure report server e-mail. Reporting Services includes a report server e-mail delivery extension that uses a Simple Mail Transfer Protocol (SMTP) to deliver reports or report processing notification to an electronic mailbox. You can use the Reporting Services Configuration Manager to specify which SMTP server or gateway on your network to use for e-mail delivery.

The Reporting Services Configuration Manager does not help you manage report server content, enable additional features, or grant access to the server. Full deployment requires that you also use SQL Server Management Studio to enable additional features or modify default values, and Report Manager to grant user access to the server.

Requirements

The Reporting Services Configuration manager is version-specific. The Reporting Services Configuration Manager that installs with this version of SQL Server 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 manager that comes with each version to configure each instance.

To use the Reporting Services Configuration manager, you must have the following:

- Local system administrator permissions on the computer that hosts the report server you want to configure. If you are configuring a remote computer, you must have local system administrator permissions on that computer as well.
- You must have permission to create databases on the SQL Server Database Engine used to host the report server database.
- Windows Management Instrumentation (WMI) service must be enabled and running on any report server you are configuring. The Reporting Services Configuration Manager uses the report server WMI provider to connect to local and remote report servers. If you are configuring a remote report server, the computer must allow remote WMI access. For more information, see Configure a Report Server for Remote Administration.
- 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 Configure a Report Server for Remote Administration in SQL Server Books Online.

The Reporting Services Configuration Manager is installed automatically when you install Reporting Services

To Start the Reporting Services Configuration Manager

To start Reporting Services Configuration

1. Use the following step that is appropriate for your version of Microsoft Windows:
   - From the Windows start screen, type **Reporting** and select **Reporting Services Configuration Manager** from the search results.
   - Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2014**, 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 SQL Server 2008 R2 instead of **Microsoft SQL Server 2014** to open the configuration tools for SQL Server 2008 R2 server components.

     Click **Reporting Services Configuration Manager**.
2. The **Reporting Services Configuration Connection** dialog box appears so that you can select the report server instance you want to configure. Click **Connect**.
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 report server instances for this version of SQL Server appear in the list. You cannot configure earlier versions of Reporting Services.
6. Click **Connect**.

See Also

Report Manager (SSRS Native Mode)
Reporting Services Tools
Configure a Report Server Database Connection (SSRS Configuration Manager)
SQL Server Configuration Manager
Configure and Administer a Report Server (SSRS Native Mode)

Configure the Report Server Service Account (SSRS Configuration Manager)

4/18/2019 • 5 minutes to read • Edit Online

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.

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 Server Configuration - Service Accounts page of the Installation Wizard becomes the initial account of the Report Server service.

> **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.

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 Administer a Report Server Database (SSRS Native Mode) in SQL Server Books Online.

  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.

> **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 Configure a Service Account (SSRS Configuration Manager).

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 Register a Service Principal Name (SPN) for a Report Server.

The following guidelines and links in this section can help you decide on an approach that is best for your deployment.

- Service Account (SSRS Native Mode).
- Configure Windows Service Accounts and Permissions in SQL Server Books Online.
- The Services and Service Accounts Security Planning Guide.

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.

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 2010 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**.

> **NOTE**
> SharePoint products and technologies require domain accounts for service configuration like reporting services SharePoint mode.

Next Steps

Configure a Service Account (SSRS Configuration Manager) Service Account (SSRS Native Mode) Configure Report Server URLs (SSRS Configuration Manager) Reporting Services Configuration Manager (Native Mode)

Configure a Service Account (SSRS Configuration Manager)

10/3/2019 • 3 minutes to read • Edit Online

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 SharePoint Foundation 2010 or SharePoint Server 2010.

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.

> **NOTE**
> Built-in Windows service accounts (Local Service or Network Service) are not supported as report server service accounts on a computer that is a domain controller.

Procedures

To configure the Report Server service account

1. Start the Reporting Services Configuration manager 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 Configure the Report Server Service Account (SSRS Configuration Manager).
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 Register a Service Principal Name (SPN) for a Report Server.
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.

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 Back Up and Restore Reporting Services Encryption Keys.
- 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

Reporting Services Configuration Manager (Native Mode)
Configure the Report Server Service Account (SSRS Configuration Manager)
Configure a Report Server Database Connection (SSRS Configuration Manager)
Service Account (SSRS Native Mode)
Configure and Manage Encryption Keys (SSRS Configuration Manager)

Configure Report Server URLs (SSRS Configuration Manager)

5/22/2019 • 5 minutes to read • Edit Online

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 Configure a URL (SSRS Configuration Manager).

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
Host name	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 card installed in a computer. If the IP address resolves to a host header, 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.
Port	A TCP port is an endpoint on the device. The report server will listen for requests on a designated port.
Virtual directory	A port is often shared by multiple Web services or applications. For this reason, a report server URL always includes a virtual 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.
SSL settings	URLs in Reporting Services can be configured to use an existing SSL certificate that you previously installed on the computer. For more information, see Configure SSL Connections on a Native Mode Report Server in SQL Server Books Online.

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 long as the IP address is specified in the URL that you 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 About URL Reservations and Registration (SSRS Configuration Manager).

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 URL Reservation Syntax (SSRS Configuration Manager).

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 Authentication with the Report Server in SQL Server Books Online.

In This Section

Configure a URL (SSRS Configuration Manager)
This topic provides instructions for setting and modifying a URL reservation in the Reporting Services Configuration tool.

About URL Reservations and Registration (SSRS Configuration Manager)
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 (SSRS Configuration Manager)
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 (SSRS Configuration Manager)
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 (SSRS Configuration Manager)
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

Configure a URL (SSRS Configuration Manager)
Web Service URL (SSRS Native Mode)

About URL Reservations and Registration (SSRS Configuration Manager)

5/22/2019 • 8 minutes to read • Edit Online

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.

> **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 HTTP Server API on MSDN.

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.

> **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 Access Report Server Items Using URL Access 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 Namespace Reservations, Registration, and Routing 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 URL you use to access the application:

URL RESERVATION IN HTTP.SYS	URL	EXPLANATION
http://+:80/reportserver	http://<computername>/reportserver http://<IPAddress>/reportserver http://localhost/reportserver	The URL reservation specifies a wildcard (+) on port 80. This puts into the report server queue any incoming request that specifies a host that resolves to the report server computer on port 80. Notice that with this URL reservation, any number of URLs can be used to access the report server. This is the default URL reservation for a Reporting Services report server for most operating systems.
http://123.45.67.0:80/reportserver	http://123.45.67.0/reportserver	This URL reservation specifies an IP address and is much more restrictive than the wildcard URL reservation. Only URLs that include the IP address can be used to connect to the report server. Given this URL reservation, a request to a report server at http://<computername>/reportserver or http://localhost/reportserver would fail.

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.

> **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:

1. Windows Server 2012
2. Windows Server 2008 R2
3. Windows Server 2008
4. Windows 7
5. Windows Vista

INSTANCE TYPE	APPLICATION	DEFAULT URL	ACTUAL URL RESERVATION IN HTTP.SYS
Default instance	Report Server Web service	http://<servername>/reportserver	http://<servername>:80/reportserver
Default instance	Report Manager	http://<servername>/reportserver	http://<servername>:80/reportserver
Named instance	Report Server Web service	http://<servername>/reportserver_<instancename>	http://<servername>:80/reportserver_<instancename>
Named instance	Report Manager	http://<servername>/reports_<instancename>	http://<servername>:80/reports_<instancename>
SQL Server Express	Report Server Web service	http://<servername>/reportserver_SQLExpress	http://<servername>:80/reportserver_SQLExpress
SQL Server Express	Report Manager	http://<servername>/reports_SQLExpress	http://<servername>:80/reports_SQLExpress

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 about authentication, see Authentication with the Report Server in SQL Server Books Online.

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 2008 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 Configure a Native Mode Report Server for Local Administration (SSRS).

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.

See Also

Configure a URL (SSRS Configuration Manager)
URL Reservation Syntax (SSRS Configuration Manager)

Configure a URL (SSRS Configuration Manager)

3/27/2020 • 9 minutes to read • Edit Online

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 and later versions.

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 About URL Reservations and Registration (SSRS Configuration Manager) and Install Reporting Services and Internet Information Services Side-by-Side (SSRS Native Mode)in SQL Server Books Online. To review examples of URLs often used in a Reporting Services installation, see Examples of URLs in this topic.

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 Configure Report Server URLs (SSRS Configuration Manager).

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.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>:*nnnn:nnnn:nnnn:nnnn:nnnn:nnnn*

     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 Vista and Windows Server 2008 because it can be shared with other applications. 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, Information about TCP/IP port assignments, 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 Configure a Firewall for Report Server Access.

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 Create a Native Mode Report Server Database (SSRS Configuration Manager).
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 Configuration and Administration of a Report Server (Reporting Services SharePoint Mode) and Reporting Services Report Server (SharePoint Mode).

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 Configure Report Manager (Native Mode) in SQL Server Books Online.
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.

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.

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 URL Reservations for Multi-Instance Report Server Deployments (SSRS Configuration Manager).

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`

See Also

Reporting Services Configuration Manager (Native Mode)
Configure Report Server URLs (SSRS Configuration Manager)

URL Reservation Syntax (SSRS Configuration Manager)

5/22/2019 • 2 minutes to read • Edit Online

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.

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.

PROPERTY	VALID VALUES	DESCRIPTION
Hostname	(+) Strong wildcard, equates to **(All Assigned)** value for the IP address. (*) Weak wildcard, equates to an IP address of **(All Unassigned)**. Fully qualified domain name Machine name IP address (IPV4) IP address (IPV6)	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 virtual directory combination. The report server will accept any request on the port. (*) 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. Machine name is the NETBIOS name of the computer on the network. 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: *nnn.nnn.nnn.nnn*. IP address (IPV6) is the IP address of a network adaptor on the computer in IPV6 format: <header>:<header>:*nnn.nnn.nnn.nnn*.
Port	80 443 <custom>	Port 80 is the standard port for HTTP requests to and from a server. Port 443 is the standard report for SSL connections. You can use any port that is not already reserved by another application.

PROPERTY	VALID VALUES	DESCRIPTION
Virtualdirectory	ReportServer*[_InstanceName] *Reports*[_InstanceName]* <custom>	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. 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 `UrlString`. 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. *InstanceName* 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 URL Reservations for Multi-Instance Report Server Deployments (SSRS Configuration Manager). The value for virtual directory is not case-sensitive. You can use any string as long as it does not include URL separator characters or URL encoding.

See Also

Configure Report Server URLs (SSRS Configuration Manager)
Configure a URL (SSRS Configuration Manager)

URLs in Configuration Files (SSRS Configuration Manager)

3/27/2020 • 5 minutes to read • Edit Online

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.

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 in SQL Server Books Online.
- For more information about the syntax of just the `UrlString` element, see URL Reservation Syntax (SSRS Configuration Manager).
- For instructions on how to configure URLs for application access, see Configure a URL (SSRS Configuration Manager).

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)

> **NOTE**
> Previous versions of Reporting Services included the `ReportServerVirtualDirectory` element. This value is obsolete in SQL Server 2008 and later versions. 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	DESCRIPTION
`ReportServerUrl`	Optional. This element is not included in the RSReportServer.config file unless you add it yourself. Set this element only if you are configuring one of the following scenarios: Report Manager provides Web front-end access to a Report Server Web service that runs on a different computer or a different instance on the same computer. When you have multiple URLs to a report server and you want Report Manager to use a specific URL. You have a specific report server URL through which you want all Report Manager connections to use. For example, you might enable Report Manager access for all computers on network, yet require that Report Manager connect to the report server through a local connection. In this case, you might configure `ReportServerUrl` to "http://localhost/reportserver". For instructions on how to implement these scenarios, see Configure Report Manager (Native Mode) in SQL Server Books Online.	This value specifies a URL to the Report Server Web service. This value is read by the Report Manager application at startup. If this value is set, Report Manager will connect to the report server that is specified in the URL. By default, Report Manager provides Web front-end access to the Report Server Web service that runs within the same report server instance as Report Manager. However, if you want to use Report Manager with a Report Server Web service that is part of another instance or runs in an instance on a different computer, you can set this URL to direct Report Manager to connect to the external Report Server Web service. If a Secure Sockets Layer (SSL) certificate is installed on the report server to which you are connecting, the `ReportServerUrl` value must be the name of the server that is registered for that certificate. If you get the error, "The underlying connection was closed: Could not establish trust relationship for the SSL/TLS security channel", set `ReportServerUrl` to the fully qualified domain name of the server for which the SSL certificate was issued. For example, if the certificate is registered to **https://adventure-works.com.onlinesales**, the report server URL would be **https://adventure-works.com.onlinesales/reportserver**.

SETTING	USAGE	DESCRIPTION
`ReportServerExternalUrl`	Optional. This element is not included in the RSReportServer.config file unless you add it yourself. Set this element only if you are using the SharePoint 2.0 Web Parts and you want users to be able to retrieve a report and open it in a new browser window. Add < `ReportServerExternalUrl` > underneath the < `ReportServerUrl` > element, and then set it to a fully qualified report server name that resolves to a report server instance when accessed in a separate browser window. Do not delete < `ReportServerUrl` >. The following example illustrates the syntax: `<ReportServerExternalUrl>http://myserver/reportserver</ReportServerExternalUrl>`	This value is used by the SharePoint 2.0 Web Parts. In previous releases, it was recommended that you set this value to deploy Report Builder on an Internet-facing report server. This is an untested deployment scenario. If you used this setting in the past to support Internet access to Report Builder, you should consider an alternative strategy.

See Also

Configure Report Server URLs (SSRS Configuration Manager)
Configure a URL (SSRS Configuration Manager)

URL Reservations for Multi-Instance Report Server Deployments (SSRS Configuration Manager)

3/27/2020 • 2 minutes to read • Edit Online

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.

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	UNIQUENESS
http://+:80/reportserver	http://+:8888/reportserver	Each instance listens on a different port.
`http://www.contoso.com/reportserver`	`http://SRVR-46/reportserver`	Each instance responds to different server names (fully qualified domain name, and machine name).

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

Configure Report Server URLs (SSRS Configuration Manager)
Configure a URL (SSRS Configuration Manager)

Create a Report Server Database (SSRS Configuration Manager)

10/3/2019 • 3 minutes to read • Edit Online

Reporting Services **Native mode** 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".

Reporting Services **SharePoint mode** includes a third database that is used for data alerting metadata. The three databases are created for each Reporting Services service application and the database names by default include a guid that represents the service application. The following are example names of the three SharePoint mode databases:

- ReportingService_90a9f37075544f22953c4a62e4a9f370
- ReportingService_90a9f37075544f22953c4a62e4a9f370TempDB
- ReportingService_90a9f37075544f22953c4a62e4a9f370_Alerting

> **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.
>
> The exception to this are the execution log views. For more information, see Report Server Execution Log and the ExecutionLog3 View

Ways to Create the Report Server Database

Native Mode: You can create the Native mode report server database in the following ways:

- Automatically: Use SQL Server Setup Wizard, if you choose the default configuration installation option. In the SQL Server Installation Wizard, this is the **Install and configure** in the Report Server Installation Options page. If you chose the **Install only** option, you must use the Reporting Services Configuration Manager to create the database.
- Manually: Use the Reporting Services Configuration Manager. 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 Create a Native Mode Report Server Database (SSRS Configuration Manager).

SharePoint Mode: The Report Server Installation Options page only has one option for SharePoint mode of **Install Only**. This option installs all the Reporting Services files and the Reporting Services shared service. The next step is to create at least one Reporting Services service application in one of the following ways:

- Use SharePoint Central Administration to create a Reporting Services service application. For more information see the "Service Application" section of Step 3: Create a Reporting Services Service Application.
- Use Reporting Services PowerShell cmdlets to create a service application and the report server databases. For more information see the sample for creating service applications in the topic PowerShell cmdlets for

Reporting Services SharePoint Mode.

Database Server Version Requirements

SQL Server is used to host the report server databases. The SQL Server Database Engine instance can be a local or remote instance. The following are the supported versions of SQL Server Database Engine that can be used to host the report server databases:

- SQL Server 2014
- SQL Server 2012
- SQL Server 2008 R2
- SQL Server 2008
- SQL Server 2005

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 Configure a Report Server Database Connection (SSRS Configuration Manager).

> **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.

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. For more information, see the "Report Server Database Server Edition Requirements" section of Features Supported by the Editions of SQL Server 2014.

See Also

Reporting Services Configuration Manager (del)

Create a Native Mode Report Server Database (SSRS Configuration Manager)

5/22/2019 • 5 minutes to read • Edit Online

Native Mode Reporting Services uses a SQL Server database for internal storage. The database is required and 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 Manager.

Applies to: Reporting Services Native Mode

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, the report server database was created and configured automatically when the report server instance was installed. You can use the Reporting Services Configuration Manager to view or modify the settings that Setup configured for you.

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
Review the supported versions of the SQL Server Database Engine and review the supported editions in the topic, Create a Report Server Database (SSRS Configuration Manager).

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 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 Manager will create a login and set the necessary permissions, it will not create an account for you. For more information, see Configure a Report Server Database Connection (SSRS Configuration Manager).

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.

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.

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 Configure a Windows Firewall for Database Engine Access in SQL Server Books Online.

To create a local report server database

1. Start the Reporting Services Configuration Manager and connect to the report server instance for which you are creating the database. For more information, see Reporting Services Configuration Manager (Native Mode).
2. On the Database page, click **Change Database**.
3. Click **Create a new report server 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:
    a. 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>.
    b. 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 Before You Start in this topic.
    c. Click **Test Connection** to validate the connection to the server.
    d. Click **Next**.
5. Specify properties used to create the database. For more information about how these properties are used, see Before You Start in this topic:
    a. 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.

b. Select the language in which you want role definitions and My Reports to appear.

c. The Report Server Mode is always set to **Native**.

d. Click **Next**.

6. Specify the credentials used by the report server to connect to the report server database.

   a. 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.

   b. 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 Verify a Reporting Services Installation in SQL Server Books Online.

See Also

Configure a Report Server Database Connection (SSRS Configuration Manager)
Database (SSRS Native Mode)
Manage a Reporting Services Native Mode Report Server
Reporting Services Configuration Manager (Native Mode)

Configure a Report Server Database Connection (SSRS Configuration Manager)

10/3/2019 • 9 minutes to read • Edit Online

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 Create a Native Mode Report Server Database (SSRS Configuration Manager).

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 Configure the Report Server Service Account (SSRS Configuration Manager).
- Configuring a report server scale-out deployment. Configuring a scale-out deployment requires that you create multiple connections to a report server database. For more information about how to perform this multi-step operation, see Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager).

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 Configure and Manage Encryption Keys (SSRS Configuration Manager).

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 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 2014, 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 2014**, click **Configuration Tools**, click **SQL Server Configuration Manager**, click **SQL Server Network Configuration**, and then click **Protocols for MSSQLSERVER**. For more information, see Enable or Disable a Server Network Protocol in SQL Server Books Online.

Defining a Report Server Database Connection

To configure the connection, you must use the Reporting Services Configuration Manager 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..
- 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 Create a Report Server Database (SSRS Configuration Manager).
- Credential type. You can use the service accounts, a Windows domain account, or a SQL Server database login.
- User name and password (required only if you are using Windows domain account or a SQL Server login).

The credentials that you provide must be granted access to the report server database. If you use the Reporting Services Configuration tool, this step is performed automatically. For more information about the permissions required to access the database, see the "Database Permissions" section in this topic.

Storing Database Connection Information

Reporting Services stores and encrypts the connection information in the following RSreportserver.config settings. You must use the Reporting Services Configuration tool or rsconfig utility to create encrypted values for these settings.

Not all of the values are set for every type of connection. If you configure the connection using the default values (that is, using the service accounts to make the connection), < `LogonUser` >, < `LogonDomain` >, and < `LogonCred` > will be empty, as follows:

```
<Dsn></Dsn>
<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.

> **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.

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 Configure the Report Server Service Account (SSRS Configuration Manager).

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 Upgrade and Migrate Reporting Services in SQL Server Books Online.

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 Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager).

See Also

Create a Report Server Database (SSRS Configuration Manager)
Manage a Reporting Services Native Mode Report Server
Configure the Report Server Service Account (SSRS Configuration Manager)

Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager)

10/3/2019 • 8 minutes to read • Edit Online

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 Manager 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.

Applies to: Reporting Services Native mode

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 Configuration CoClass 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.

> **NOTE**
> The default e-mail delivery extension does not provide support for digitally signing or encrypting outgoing mail messages.

Configuring a Report Server for Local or Remote SMTP Service

You can use a local SMTP service or a remote SMTP server or forwarder to support e-mail delivery. If you have access to an existing remote SMTP server, you should consider using it. If there is no SMTP server available or if you subsequently encounter report delivery errors that can be attributed to computer connection failures, you should switch to using a local SMTP service. Details about how to configure a report server for local or remote service are provided further on in this topic.

Setting Configuration Options for E-Mail Delivery

Before you can use Report Server e-mail delivery, you must set configuration values that provide information about which SMTP server to use.

To configure a report server for e-mail delivery, do the following:

- Use the Reporting Services Configuration Manager 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 E-mail Settings - Configuration Manager (SSRS Native Mode) and E-Mail Delivery in Reporting Services.
- (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 Modify a Reporting Services Configuration File (RSreportserver.config) in SQL Server Books Online.

> **NOTE**
> Report server e-mail settings are based on CDO. If you want more detail about specific settings, you can refer to the CDO production documentation.

Example Report Server E-Mail Configuration

The following example illustrates the settings in the RSreportserver.config file for a remote SMTP server. To read about the setting descriptions and valid values, see RSReportServer Configuration File in SQL Server Books Online or the CDO product documentation.

```
<RSEmailDPConfiguration>
     <SMTPServer>mySMTPServer.Adventure-Works.com</SMTPServer>
     <SMTPServerPort></SMTPServerPort>
     <SMTPAccountName></SMTPAccountName>
     <SMTPConnectionTimeout></SMTPConnectionTimeout>
     <SMTPServerPickupDirectory></SMTPServerPickupDirectory>
     <SMTPUseSSL></SMTPUseSSL>
     <SendUsing>2</SendUsing>
     <SMTPAuthenticate></SMTPAuthenticate>
     <From>my-rs-email-account@Adventure-Works.com</From>
     <EmbeddedRenderFormats>
          <RenderingExtension>MHTML</RenderingExtension>
     </EmbeddedRenderFormats>
     <PrivilegedUserRenderFormats></PrivilegedUserRenderFormats>
     <ExcludedRenderFormats>
          <RenderingExtension>HTMLOWC</RenderingExtension>
          <RenderingExtension>NULL</RenderingExtension>
     </ExcludedRenderFormats>
     <SendEmailToUserAlias>True</SendEmailToUserAlias>
     <DefaultHostName></DefaultHostName>
     <PermittedHosts>
          <HostName>Adventure-Works.com</HostName>
          <HostName>hotmail.com</HostName>
     </PermittedHosts>
</RSEmailDPConfiguration>
```

Configuration Options for Setting the To: Field in a Message

User-defined subscriptions that are created according to the permissions granted by the **Manage individual subscriptions** task contain a pre-set user name that is based on the domain user account. When the user creates the subscription, the recipient name in the **To:** field is self-addressed using the domain user account of the person creating the subscription.

If you are using an SMTP server or forwarder that uses e-mail accounts that are different from the domain user account, the report delivery will fail when the SMTP server tries to deliver the report to that user.

To 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.

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**.

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 Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager) and E-mail Settings - Configuration Manager (SSRS Native Mode).

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.

To configure report server e-mail using the Reporting Services Configuration Manager

1. Verify that the Report Server Windows service has `Send As` permissions on the SMTP server.
2. Start the Reporting Services Configuration Manager 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**.

To configure a remote SMTP Service for the report server

1. Verify that the Report Server Windows service has `Send As` permissions on the SMTP server.
2. Open the RSReportServer.config file in a text editor.
3. Verify that < `UrlRoot` > is set to the report server URL address. This value is set when you configure the report server and it should be filled in already. If it is not set, type the report server URL address.
4. In the Delivery section, find < `ReportServerEmail` >.
5. In < `SMTPServer` >, type the name of the SMTP server. This value can be an IP address, a UNC name of a computer on your corporate intranet, or a fully qualified domain name.
6. Verify that < `SendUsing` > is set to 2. If it is set another value, the report server is not configured to use a remote SMTP service.
7. In < `From` >, type the name an account that has permission to send e-mail from the SMTP server.
8. Save the file.

   The report server will use the new settings automatically; you do not need to restart the service. You can specify additional SMTP settings to further configure how the SMTP server is used for report server e-mail delivery. For more information, see Configuring a Report Server for E-Mail Delivery and RSReportServer Configuration File in SQL Server Books Online.

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

Reporting Services Configuration Manager (Native Mode)

Configure the Unattended Execution Account (SSRS Configuration Manager)

5/22/2019 • 5 minutes to read • Edit Online

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 Specify 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.

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.

> **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.

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 Reporting Services Configuration Manager (Native Mode).

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 Reporting Services Configuration Manager (Native Mode).
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 (SSRS) 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:

```
<UnattendedExecutionAccount>
     <UserName></UserName>
     <Password></Password>
     <Domain></Domain>
</UnattendedExecutionAccount>
```

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 on a SharePoint site, select the **Credentials are not required** 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.

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

Reporting Services Configuration Manager (del)

Configure and Manage Encryption Keys (SSRS Configuration Manager)

5/22/2019 • 2 minutes to read • Edit Online

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.

> **IMPORTANT**
> Periodically changing the Reporting Services encryption key is a security best practice. A recommended time to change the key is immediately following a major version upgrade of Reporting Services. Changing the key after an upgrade minimizes additional service interruption caused by changing the Reporting Services encryption key outside of the upgrade cycle.

To manage symmetric keys, you can use the Reporting Services Configuration tool or the **rskeymgmt** utility. The tools included in Reporting Services are used to manage the symmetric key only (the public and private keys are managed by the operating system). Both the Reporting Services Configuration tool and the **rskeymgmt** utility support the following tasks:

- Back up a copy of the symmetric key so that you can use it to recover a report server installation or as part of a planned migration.
- Restore a previously saved symmetric key to a report server database, allowing a new report server instance to access existing data that it did not originally encrypt.
- Delete the encrypted data in a report server database in the unlikely event that you can no longer access encrypted data.
- Re-create symmetric keys and re-encrypt data in the unlikely event that the symmetric key is compromised. As a security best practice, you should recreate the symmetric key periodically (for example, every few months) to protect the report server database from cyber attacks that attempt to decipher the key.
- Add or remove a report server instance from a report server scale-out deployment where multiple report servers share both a single report server database and the symmetric key that provides reversible encryption for that database.

In This Section

Initialize a Report Server (SSRS Configuration Manager)
Explains how encryption keys are created.

Back Up and Restore Reporting Services Encryption Keys

Explains how to back up encryption keys and restore them to recover or migrate a report server installation.

Store Encrypted Report Server Data (SSRS Configuration Manager)
Describes encryption on a report server.

Delete and Re-create Encryption Keys (SSRS Configuration Manager)
Explains how you can replace a symmetric key with a new version, and how to start over if symmetric keys cannot be validated.

Add and Remove Encryption Keys for Scale-Out Deployment (SSRS Configuration Manager)
Explains how to add and remove encryption keys to control which report servers are part of a scale-out deployment.

See Also

Store Encrypted Report Server Data (SSRS Configuration Manager)

Initialize a Report Server (SSRS Configuration Manager)

5/22/2019 • 4 minutes to read • Edit Online

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 Configure and Manage Encryption Keys (SSRS Configuration Manager) and Store Encrypted Report Server Data (SSRS Configuration Manager).

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 SSRS: Troubleshoot Issues and Errors with Reporting Services (https://social.technet.microsoft.com/wiki/contents/articles/1633.aspx) in Technet Wiki.

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.

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 Configure a Report Server Database Connection (SSRS Configuration Manager).
- 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 Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager).

> **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 Configure and Manage Encryption Keys (SSRS Configuration Manager).

> **NOTE**
> You can also use the Reporting Services Windows Management Instrumentation (WMI) provider to initialize a report server programmatically. For more information, see Access the Reporting Services WMI Provider in SQL Server Books Online.

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

Store Encrypted Report Server Data (SSRS Configuration Manager)

Store Encrypted Report Server Data (SSRS Configuration Manager)

5/22/2019 • 4 minutes to read • Edit Online

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.

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 machine-level 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 Configure a Report Server Database Connection (SSRS Configuration Manager).

- 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 Specify Credential and Connection Information for Report Data Sources in SQL Server Books Online.

- 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 Configure the Unattended Execution Account (SSRS Configuration Manager).

- 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.

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 Initialize a Report Server (SSRS Configuration Manager).

> **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 Configure the Report Server Service Account (SSRS Configuration Manager).

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.

> **NOTE**
> Reporting Services does not support caching or report history for reports that use parameters based on the security identify of the user.

See Also

Configure and Manage Encryption Keys (SSRS Configuration Manager)

Back Up and Restore Reporting Services Encryption Keys

10/23/2019 • 4 minutes to read • Edit Online

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.

Applies to: Reporting Services Native Mode | Reporting Services SharePoint mode

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 Manager, backing up the key is part of a service account name change operation.

  > **NOTE**
  > Resetting the password is not the same as changing the password. A password reset requires permission to overwrite account information on the domain controller. Password resets are performed by a system administrator when you forget or do not know a particular password. Only password resets require symmetric key restoration. Periodically changing an account password does not require you to reset the symmetric key.

- Renaming the computer or instance that hosts the report server (a report server instance is based on a SQL Server instance name).
- Migrating a report server installation or configuring a report server to use a different report server database.
- Recovering a report server installation due to hardware failure.

You only need to back up one copy of the symmetric key. There is a one-to-one correspondence between a report server database and a symmetric key. Although you only need to back up one copy, you might need to restore the key multiple times if you are running multiple report servers in a scale-out deployment model. Each report server instance will need its copy of the symmetric key to lock and unlock data in the report server database.

Backing Up the Encryption Keys

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 encrypt 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 the following tools:

Native mode: Either the Reporting Services Configuration Manager or the **rskeymgmt** utility.

SharePoint mode: SharePoint Central Administration pages or PowerShell.

Backup SharePoint Mode Report Servers

For SharePoint mode report servers you can either use PowerShell commands or use the management pages for the Reporting Services service application. For more information, see the "Key Management" section of Manage a Reporting Services SharePoint Service Application

Back up encryption keys -Reporting Services Configuration Manager (Native Mode)

1. Start the Reporting Services Configuration Manager, 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 disk separate from the report server.
5. Click **OK**.

Back up encryption keys -rskeymgmt (Native Mode)

1. 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:

   ```
   rskeymgmt -e -f d:\rsdbkey.snk -p<password>
   ```

Restore 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 Delete and Re-create Encryption Keys (SSRS Configuration Manager). For more information about creating symmetric keys, see Initialize a Report Server (SSRS Configuration Manager).

Restore encryption keys -Reporting Services Configuration Manager (Native Mode)

1. Start the Reporting Services Configuration Manager, 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**.

Restore encryption keys - rskeymgmt (Native Mode)

1. 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 d:\rsdbkey.snk -p<password>
```

See Also

Configure and Manage Encryption Keys (SSRS Configuration Manager)

Delete and Re-create Encryption Keys (SSRS Configuration Manager)

5/22/2019 • 5 minutes to read • Edit Online

Deleting and re-creating encryption keys are activities that fall outside of routine encryption key maintenance. You perform these tasks in response to a specific threat to your report server, or as a last resort when you can no longer access a report server database.

- Re-create the symmetric key when you believe the existing symmetric key is compromised. You can also re-create the key on a regular basis as a security best practice.
- Delete existing encryption keys and unusable encrypted content when you cannot restore the symmetric key.

Re-creating Encryption Keys

If you have evidence that the symmetric key is known to unauthorized users, or if your report server has been under attack and you want to reset the symmetric key as a precaution, you can re-create the symmetric key. When you re-create the symmetric key, all encrypted values will be re-encrypted using the new value. If you are running multiple report servers in a scale-out deployment, all copies of the symmetric key will be updated to the new value. The report server uses the public keys available to it to update the symmetric key for each server in the deployment.

You can only re-create the symmetric key when the report server is in a working state. Re-creating the encryption keys and re-encrypting content disrupts server operations. You must take the server offline while re-encryption is underway. There should be no requests made to the report server during re-encryption.

You can use the Reporting Services Configuration tool or the **rskeymgmt** utility to reset the symmetric key and encrypted data. For more information about how the symmetric key is created, see Initialize a Report Server (SSRS Configuration Manager).

How to re-create encryption keys (Reporting Services Configuration Tool)

1. 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.

   a. 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_50.MSSQLSERVER\Reporting Services.

   b. Open the rsreportserver.config file.

   c. 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:

   ```
   rskeymgmt -s
   ```

3. Restart the Windows service and enable Web service operations.

Deleting Unusable Encrypted Content

If for some reason you cannot restore the encryption key, the report server will never be able to decrypt and use any data that is encrypted with that key. To return the report server to a working state, you must delete the encrypted values that are currently stored in the report server database and then manually re-specify the values you need.

Deleting the encryption keys removes all symmetric key information from the report server database and deletes any encrypted content. All unencrypted data is left intact; only encrypted content is removed. When you delete the encryption keys, the report server re-initializes itself automatically by adding a new symmetric key. The following occurs when you delete encrypted content:

- Connection strings in shared data sources are deleted. Users who run reports get the error "The ConnectionString property has not been initialized."
- Stored credentials are deleted. Reports and shared data sources are reconfigured to use prompted credentials.
- Reports that are based on models (and require shared data sources configured with stored or no credentials) will not run.
- Subscriptions are deactivated.

Once you delete encrypted content, you cannot recover it. You must re-specify connection strings and stored credentials, and you must activate subscriptions.

You can use the Reporting Services Configuration tool or the **rskeymgmt** utility to remove the values.

How to delete encryption keys (Reporting Services Configuration Tool)

1. Start the Reporting Services Configuration tool, and then connect to the report server instance you want to configure.
2. Click **Encryption Keys**, and then click **Delete**. Click **OK**.
3. Restart the Report Server Windows service. For a scale-out deployment, do this on all report server instances.

How to delete encryption keys (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
```

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 Specify Credential and Connection Information for Report Data Sources in SQL Server Books Online.
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

Configure and Manage Encryption Keys (SSRS Configuration Manager)
Store Encrypted Report Server Data (SSRS Configuration Manager)

Add and Remove Encryption Keys for Scale-Out Deployment (SSRS Configuration Manager)

5/22/2019 • 3 minutes to read • Edit Online

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 Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager).

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 (SSRS).

   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> -u <administratoraccount> -v
   <administratorpassword>
   ```

How to remove a report server from a scale-out deployment (rskeymgmt)

1. Open the rsreportserver.config file of the report server you want to remove and find the installation ID. By default, this file is located at Program Files\Microsoft SQL Server\MSSQL.*n*\Reporting Services\ReportServer).

   If you installed a single instance, there will only be one rsreportserver.config file on the computer. If multiple instances of Reporting Services are installed, use the Server Status page in the Reporting Services

Configuration tool to find the instance identifier (for example, MSSQL.2) for the report server that you want to remove. The name of the folder that stores the program files for the report server instance will be based on the instance identifier (for example, Program Files\Microsoft SQL Server\MSSQL.2).

2. Run **rskeymgmt.exe**. You can run it on any report server that is part of the report server scale-out deployment.

3. Use the `-r` argument to release the report server instance from the scale-out deployment. The following example illustrates the arguments you must specify:

```
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 Uninstall an Existing Instance of SQL Server (Setup) in SQL Server Books Online.

See Also

Configure and Manage Encryption Keys (SSRS Configuration Manager)
Initialize a Report Server (SSRS Configuration Manager)

Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager)

2/28/2020 • 6 minutes to read • Edit Online

Reporting Services native mode supports a scale-out deployment model that allows you to run multiple report server instances that share a single report server database. 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

SharePoint mode report servers utilize the SharePoint products infrastructure for scale-out. SharePoint mode scale-out is performed by adding more SharePoint mode report servers to the SharePoint farm. For information on scale-out in SharePoint mode, see Add an Additional Report Server to a Farm (SSRS Scale-out).

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 Install SQL Server 2014 from the Installation Wizard (Setup) in SQL Server Books Online for instructions on how to install report server instances.
- 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 Configure a Report Server on a Network Load Balancing Cluster.
- Review the procedures in this topic for instructions on how to share a report server database and join report servers to a scale-out.

  The procedures 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.

  - Use Setup to install each report server instance that will be joined to the scale-out deployment.

    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 2014 report server instance, all other instances in the same deployment must also be SQL Server 2014.

  - Use the Reporting Services Configuration manager 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 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.

To install the first report server instance

1. 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 Configure a Report Server (Reporting Services Native Mode) in SQL Server Books Online.

4. Verify that the report server is operational. For more information, see Verify a Reporting Services Installation in SQL Server Books Online.

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:

    a. Click **Database** to open the Database page.

    b. Click **Change Database**.

    c. Click **Choose an existing report server database**.

    d. Type the server name of the SQL Server Database Engine instance that hosts the report server database you want to use. This must be the same server that you connected to in the previous set of the instructions.

    e. Click **Test Connection**, and then click **Next**.

    f. 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.

    g. 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**.

    h. 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.

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**.

> **NOTE**
>
> **Issue:** When you attempt to join a Reporting Services report server instance to the scale-out deployment, you may experience error messages similar to 'Access Denied'.
>
> **Workaround:** Back up the Reporting Services encryption key from the first Reporting Services instance and restore the key to the second Reporting Services report server. Then try to join the second server to the Reporting Services scale-out deployment.

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 Configure a Report Server on a Network Load Balancing Cluster.

See Also

Configure a Service Account (SSRS Configuration Manager) Configure a URL (SSRS Configuration Manager) Create a Native Mode Report Server Database (SSRS Configuration Manager) Configure Report Server URLs (SSRS Configuration Manager) Configure a Report Server Database Connection (SSRS Configuration Manager) Add and Remove Encryption Keys for Scale-Out Deployment (SSRS Configuration Manager) Manage a Reporting Services Native Mode Report Server

Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)

10/3/2019 • 2 minutes to read • Edit Online

This section provides F1 help for the Reporting Services Configuration Manager.

Applies to: Reporting Services Native mode.

In This Section

- Connect to a Report Server
- Report Server Status
- Service Account
- Web Service URL
- Advanced Multiple Web Site Configuration
- Database
- Change Database Wizard
- Change Credentials Wizard
- Report Manager URL
- E-mail Settings
- Execution Account
- Encryption Keys
- Backup Encryption Key
- Restore Encryption Key
- Scale-out Deployment

See Also

Reporting Services Configuration Manager (del)

Connect to a Native Mode Report Server

10/3/2019 • 2 minutes to read • Edit Online

Use this dialog box to connect to a local or remote SQL Server 2008 or later 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.

Applies to: Reporting Services Native mode.

> **NOTE**
> The Reporting Services Configuration Manager is not used to configure and administer Reporting Services SharePoint mode. You Use SharePoint Central Administration and PowerShell scripts to configure a report server in SharePoint mode. For more information, see Install Reporting Services SharePoint Mode for SharePoint 2010

> **TIP**
> TheReporting Services Configuration Manager (RSConfigTool.exe) is installed with a privilege level of "highestAvailable". This behavior is by design. The Reporting Services Configuration Manager requires communication with Reporting Services WMI APIs. Some of the Reporting Services WMI communication requires a higher level or administrative of privileges.

- To connect to a local report server instance, use the default values and click **Connect**. The Reporting Services Configuration Manager 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 Manager. This dialog box appears immediately when you start the tool. For more information, see Reporting Services Configuration Manager (Native Mode).

Options

Server Name

Enter the network name of the computer on which SQL Server 2008 or later 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

Report Server Status (SSRS Native Mode)

10/3/2019 • 3 minutes to read • Edit Online

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.

Applies to: Reporting Services Native mode.

To open this page, start the Reporting Services Configuration Manager and connect to the report server instance. For more information, see Reporting Services Configuration Manager (del).

> **TIP**
> TheReporting Services Configuration Manager (RSConfigTool.exe) is installed with a privilege level of "highestAvailable". This behavior is by design. The Reporting Services Configuration Manager requires communication with Reporting Services WMI APIs. Some of the Reporting Services WMI communication requires a higher level or administrative of privileges.

If you connect to the report server and all of the page links are grayed out, verify that the Report Server service is started. The **Report Service Status:** Should be "Started". You can also 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 Work with Multiple Versions and Instances of SQL Server in SQL Server Books Online.

> **NOTE**
> In SQL Server Express 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 the following:

- MSSQL12.MSSQLSERVER
- MSAS12.MSSQLSERVER
- MSRS12.MSSQLSERVER

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 MSSQL12.Contoso.

Edition

Displays edition information. For a list of features that are supported by the editions of SQL Server, see Features Supported by the Editions of SQL Server.

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

This should always show a value of **Native**. The Reporting Services configuration manager only supports Native mode report servers. If you see a value of **SharePoint integrated mode**, it may indicate your Native mode deployment is not configured correctly, and you need to connect to a Native mode report catalog. Use the **Database** page of Configuration Manager to change the database and either create a new database or connect to an existing native mode report server database catalog.

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 Start and Stop the Report Server Service in SQL Server Books Online.

See Also

Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Reporting Services Configuration Manager (del)
Initialize a Report Server (SSRS Configuration Manager)

Service Account (SSRS Native Mode)

10/3/2019 • 7 minutes to read • Edit Online

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.

Applies to: Reporting Services Native mode.

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 Manager to set the account. If you use the service account to connect to the report server database, the Configuration Manager 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 Manager and select the link in the navigation pane. For more information, see Reporting Services Configuration Manager (Native Mode).

> **IMPORTANT**
> Whenever you need to update the account or password, it is strongly recommended that you use the Reporting Services Configuration Manager. Using the Configuration Manager 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 Register a Service Principal Name (SPN) for a Report Server.

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.

NOTE

The Reporting Services Configuration manager 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 Encryption Keys (SSRS Native Mode).

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 Manager, 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 a SharePoint product or technology. For more information about how to grant database access in SharePoint Central Administration, see Configuration and Administration of a Report Server (Reporting Services SharePoint Mode) and Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013).

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. A domain user account is recommended because it isolates the Report 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 SharePoint integrated mode with SharePoint 2010 Products which require domain user accounts rather than built-in machine accounts. 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 Register a Service Principal Name (SPN) for a Report Server. 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 Considerations for Using Local Accounts in this topic.

ACCOUNT	EXPLANATION
Network Service	**Network Service** is a built-in least-privilege account that has network 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 **Network Service**, 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	**Local Service** is a built-in account that is like an authenticated local Windows user account. Services that run as the **Local Service** 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	**Local System** 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 **Network Service** instead.

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.
- Built-in Windows service accounts (Local Service or Network Service) are not supported as report server service accounts on a computer that is a domain controller.

See Also

Configure the Report Server Service Account (SSRS Configuration Manager)
Configure a Service Account (SSRS Configuration Manager)
Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)

Web Service URL (SSRS Native Mode)

10/3/2019 • 4 minutes to read • Edit Online

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 Configure a URL (SSRS Configuration Manager) and Configure Report Server URLs (SSRS Configuration Manager).

Applies to: Reporting Services Native mode.

To open this page, start the Reporting Services Configuration Manager and click **Web Service URL** in the navigation pane. For more information, see Reporting Services Configuration Manager (Native Mode).

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 Create a Native Mode Report Server Database (SSRS Configuration Manager).

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.

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.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:nnnn:nnnn: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.

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 a 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 Configure SSL Connections on a Native Mode Report Server 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.

> **NOTE**
>
> If you have existing SSL Bindings and URL Reservations and you want to change the SSL Binding, for example use a different certificate or hostheader, then it is recommended you complete the following steps in order:
>
> 1. First remove all URL Reservations.
> 2. Then remove all SSL Bindings.
> 3. Then recreate the URLs and the SSL bindings.
>
> The previous steps can be completed using Reporting Services Configuration Manager.
>
> Microsoft Windows supports one binding for each IP address to Port combination. If you configure a report server to use a specific hostheader value and the certificate on the Port to IP address combination is also issued to a different hostheader value, you will see in your browser, a warning indicating the certificate does not match the URL that is being used.
>
> To correct the issue, delete all bindings and then create new bindings with unique settings or configure the Reporting Services URL registrations with wildcards.

See Also

Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Configure Report Server URLs (SSRS Configuration Manager)

Advanced Multiple Web Site Configuration (SSRS Native Mode)

10/3/2019 • 3 minutes to read • Edit Online

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.

Applies to: Reporting Services Native mode.

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 Manager. 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.

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.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:nnnn:nnnn: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 Configure SSL Connections on a Native Mode Report Server 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.

See Also

Reporting Services Configuration Manager (Native Mode)
Configure a URL (SSRS Configuration Manager)
Configure Report Server URLs (SSRS Configuration Manager)

Database (SSRS Native Mode)

10/3/2019 • 2 minutes to read • Edit Online

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 Manager to create the database.

Applies to: Reporting Services Native mode.

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 Create a Native Mode Report Server Database (SSRS Configuration Manager).

To open this page, start the Reporting Services Configuration Manager and click **Database** in the navigation pane. For more information, see Reporting Services Configuration Manager (Native Mode).

Options

SQL Server Name

In Current Report Server Database, **SQL Server Name** specifies the name of the SQL Server 2014 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 Reporting Services Report Server.

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 Configure a Report Server Database Connection (SSRS Configuration Manager).

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.

See Also

Create a Native Mode Report Server Database (SSRS Configuration Manager)
Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Report Server Database (SSRS Native Mode)
Configure a Report Server Database Connection (SSRS Configuration Manager)

Change Database Wizard (SSRS Native Mode)

10/3/2019 • 5 minutes to read • Edit Online

The Reporting Services Configuration Manager 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 Manager. For instructions on how to start the Reporting Services Configuration Manager, see Reporting Services Configuration Manager (Native Mode).

Applies to: Reporting Services Native mode.

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 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 Manager 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 Manager.

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 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 Manager.

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 Manager 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 (SSRS Native Mode) in SQL Server Books Online.

You can also choose an existing report server database. The Reporting Services Configuration Manager 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.

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.

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 Configure a Report Server Database Connection (SSRS Configuration Manager).

Summary

Verify the settings before Setup configures the connection.

Progress and Finish

Monitor the progress of each task.

See Also

Database (SSRS Native Mode)
Change Credentials Wizard (SSRS Native Mode)
Create a Native Mode Report Server Database (SSRS Configuration Manager)
Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Configure a Report Server Database Connection (SSRS Configuration Manager)

Change Credentials Wizard (SSRS Native Mode)

10/3/2019 • 2 minutes to read • Edit Online

The Reporting Services Configuration Manager 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 Configuration Manager 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 Manager. For instructions on how to start the Reporting Services Configuration Manager, see Reporting Services Configuration Manager (Native Mode).

Applies to: Reporting Services Native mode.

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 Manager 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 Manager.

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 Configure a Report Server Database Connection (SSRS Configuration Manager).

Summary

Verify the settings before the wizard runs.

Progress and Finish

Monitor the progress of each task.

See Also

Database (SSRS Native Mode)

Change Database Wizard (SSRS Native Mode)
Create a Native Mode Report Server Database (SSRS Configuration Manager)
Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Configure a Report Server Database Connection (SSRS Configuration Manager)

Report Manager URL (SSRS Native Mode)

10/3/2019 • 2 minutes to read • Edit Online

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 connection to a remote report server, see Reporting Services Configuration Manager (Native Mode).

Applies to: Reporting Services Native mode.

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 Manager and click **Report Manager URL** in the navigation pane. For more information about how to start the Configuration Manager, see Reporting Services Configuration Manager (Native Mode).

> **NOTE**
> If Report Manager is not enabled, you cannot set options on this page. For more information about enabling Report Manager, see Reporting Services Configuration Manager (Native Mode).

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.

See Also

Configure a URL (SSRS Configuration Manager)
URLs in Configuration Files (SSRS Configuration Manager)
Configure Report Server URLs (SSRS Configuration Manager)

E-mail Settings - Configuration Manager (SSRS Native Mode)

5/22/2019 • 2 minutes to read • Edit Online

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 Manager. To configure the additional settings, you must manually edit the RSReportServer.config file. For more information, see Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager) and Reporting Services Configuration Files in SQL Server Books Online.

To open this page, start the Reporting Services Configuration Manager and click **E-mail Settings** in the navigation pane. For more information, see Reporting Services Configuration Manager (Native Mode).

Applies to: Reporting Services Native mode.

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 Manager.

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 Manager 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 Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager).

SMTP Server

Specify the SMTP server or gateway to use. You can use a local server or an SMTP server on your network.

See Also

Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager)
Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)

Execution Account (SSRS Native Mode)

10/3/2019 • 2 minutes to read • Edit Online

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.

Applies to: Reporting Services Native mode.

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.

> **NOTE**
>
> If you are using SQL Server 2012 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. For a list of features that are supported by the editions of SQL Server, see Features Supported by the Editions of SQL Server 2012.

To open this page, start the Reporting Services Configuration Manager and select **Execution Account** in the navigation pane. For more information, see Reporting Services Configuration Manager (Native Mode).

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.

See Also

Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Store Encrypted Report Server Data (SSRS Configuration Manager)
Configure the Unattended Execution Account (SSRS Configuration Manager)

Encryption Keys (SSRS Native Mode)

10/3/2019 • 4 minutes to read • Edit Online

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. The symmetric key is created and applied automatically when you create the report server database. 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.

Applies to: Reporting Services Native mode.

> **IMPORTANT**
> Periodically changing the Reporting Services encryption key is a security best practice. A recommended time to change the key is immediately following a major version upgrade of Reporting Services. Changing the key after an upgrade minimizes additional service interruption caused by changing the Reporting Services encryption key outside of the upgrade cycle.

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 Manager 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 to store 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, re-encrypt the symmetric key using the private key of the new Report Server service identity. Restoring the symmetric key is the process by which the re-encryption occurs.

Only restore a symmetric key 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 can no longer access sensitive data. In this case, delete and re-create the key.

> **IMPORTANT**
> The action of deleting and recreating the symmetric key cannot be reversed or undone. Deleting or recreating the key can have important ramifications on your current installation. If you delete the key, any existing data encrypted by the symmetric key will also deleted. 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 Manager and select the link in the navigation pane. For more information, see Reporting Services Configuration Manager (Native Mode).

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 Initialize a Report Server (SSRS Configuration Manager).

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

Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Back Up and Restore Reporting Services Encryption Keys
Delete and Re-create Encryption Keys (SSRS Configuration Manager)
Initialize a Report Server (SSRS Configuration Manager)
Store Encrypted Report Server Data (SSRS Configuration Manager)

Backup Encryption Key (SSRS Native Mode)

10/3/2019 • 2 minutes to read • Edit Online

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.

Applies to: Reporting Services Native mode.

To open the Backup Encryption Key dialog box, click **Encryption Keys** in the navigation pane of the Reporting Services Configuration Manager, 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 Manager. For more information on the Reporting Services Configuration Manager, see Reporting Services Configuration Manager (Native Mode).

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

Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Back Up and Restore Reporting Services Encryption Keys
Delete and Re-create Encryption Keys (SSRS Configuration Manager)
Initialize a Report Server (SSRS Configuration Manager)
Store Encrypted Report Server Data (SSRS Configuration Manager)
Encryption Keys (SSRS Native Mode)

Restore Encryption Key (SSRS Native Mode)

10/3/2019 • 2 minutes to read • Edit Online

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. This topic is an overview of how to use the Reporting Services Configuration Manager to restore keys.

Applies to: Reporting Services Native mode.

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 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.

Restore Encryption Key Dialog

For information on where to find the Reporting Services Configuration Manager, see Reporting Services Configuration Manager (Native Mode).

To open the Restore Encryption Key dialog box, click **Encryption Keys** in the navigation pane of the Reporting Services Configuration Manager, 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 Manager. For more information on the

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

Reporting Services Configuration Manager F1 Help Topics (SSRS Native Mode)
Back Up and Restore Reporting Services Encryption Keys
Delete and Re-create Encryption Keys (SSRS Configuration Manager)
Initialize a Report Server (SSRS Configuration Manager)

Store Encrypted Report Server Data (SSRS Configuration Manager)
Encryption Keys (SSRS Native Mode)

Scale-out Deployment (Native Mode Report Server)

10/3/2019 • 4 minutes to read • Edit Online

Use the **Scale-out Deployment** page in Reporting Services Configuration Manager to view the initialization status for a scale-out deployment or to join a report server to a scale-out deployment. A *scale-out deployment* refers to two or more report server instances that share a single report server database.

Applies to: Reporting Services Native mode.

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* 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 Manager and select **Scale-out Deployment** in the navigation pane. For more information, see Reporting Services Configuration Manager (Native Mode).

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 scale-out 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 Encryption Keys (SSRS Native Mode).

See Also

Initialize a Report Server (SSRS Configuration Manager)
Configure and Manage Encryption Keys (SSRS Configuration Manager)
Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager)

Install SQL Server BI Features with SharePoint (PowerPivot and Reporting Services)

4/14/2020 • 7 minutes to read • Edit Online

Analysis Services and Reporting Services can be integrated with a Microsoft SharePoint farm to enable Business Intelligence (BI) features in SharePoint. The features include PowerPivot for SharePoint, Power View, and Reporting Services. PowerPivot for SharePoint is used for PowerPivot data access in a SharePoint farm. PowerPivot for SharePoint is the data engine for workbooks created in PowerPivot for Excel and accessed from a SharePoint library. Once you save a PowerPivot workbook to SharePoint, you can use it as a data source for Power View reports.

Some of the installation and configuration steps required for SharePoint 2010 are different than the steps required for SharePoint 2013. Some of the topics in this section apply to both versions of SharePoint.

Applies to: SharePoint 2013 | SharePoint 2010

Tip For the current release notes, see SQL server 2014 Release Notes.

In this topic

- SQL Server BI Scenarios and SharePoint 2013
- Overview of Installation

In This Section

In addition to the information in this topic, the following related topics are in this section of content.

Deployment Topologies for SQL Server BI Features in SharePoint

Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm

Checklists for Installing BI Features with SharePoint

Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013)

PowerPivot for SharePoint 2013 Installation

PowerPivot for SharePoint 2010 Installation

SQL Server BI Scenarios and SharePoint 2013

This section summarizes the different levels of BI features you can choose to install and configure.

Excel Services in SharePoint 2013 includes data model functionality to enable interaction with a PowerPivot workbook in the browser. For basic data model functionality you do not need to deploy the PowerPivot for SharePoint 2013 add-in into the farm. You only need to install an Analysis Services server in SharePoint mode and register the server within the Excel Services **Data Model** settings.

Deploying the PowerPivot for SharePoint 2013 add-in enables additional functionality and features in your SharePoint farm. The additional features include PowerPivot Gallery, Schedule Data Refresh, and the PowerPivot

Management Dashboard. See the table for additional information.

	LEVEL	FEATURES	INSTALL OR CONFIGURE
1	SharePoint Only	Native Excel Services Features	Excel Services and other services included with SharePoint Server 2013.
2	SharePoint with Analysis Services in SharePoint Mode	Interactive PowerPivot workbooks in the browser	Install Analysis Services in SharePoint mode. Register Analysis Services Server in Excel Services.
3	SharePoint with Reporting Services in SharePoint Mode	Power View	Install Reporting Services in SharePoint mode. Install Reporting Services add-in **(rsSharePoint.msi)** for SharePoint. For more information, see Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013)
4	All PowerPivot Features	Access to workbooks as a data source from outside the farm. Schedule Data refresh. PowerPivot Gallery. Management Dashboard. BISM link file content type.	Deploy PowerPivot for SharePoint 2013 add-in **(spPowerPivot.msi)**. For more information, see the following: Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013) For information see on how to download **spPowerPivot.msi**, see Download SQL Server 2014 PowerPivot for SharePoint.

For additional information on enabling Analysis Services features, see The SQL Server BI Light-Up Story for SharePoint 2013 (https://blogs.msdn.com/b/analysisservices/archive/2012/07/27/introducing-the-bi-light-up-story-for-sharepoint-2013.aspx).

Overview of Installation

If you want to use both PowerPivot for SharePoint and Reporting Services, run SQL Server Installation Wizard twice. Reporting Services and PowerPivot are separate choices on the **Setup Role** page of the SQL Server setup wizard.

PowerPivot for SharePoint supports both SharePoint 2010 and SharePoint 2013; however a different architecture and installation process is used depending on the version of SharePoint.

The following is a summary of the installation steps to deploy SQL Server 2014 BI Features on a single server:

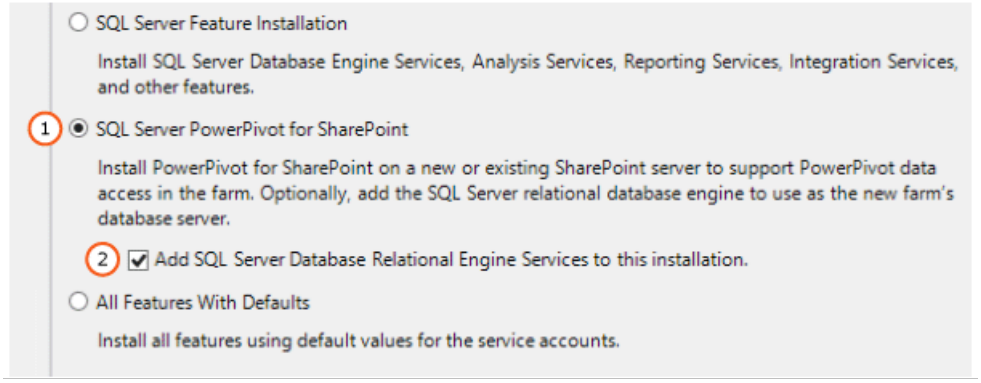
PowerPivot for SharePoint 2013

For **SharePoint 2013**, the PowerPivot for SharePoint installation can be run on a server that does not have a

SharePoint product installed. The PowerPivot architecture used for SharePoint 2013, runs **outside** the SharePoint farm and can either be installed on a server that also contains a SharePoint installation or it can be installed a server that does NOT contains a SharePoint installation.

1. Install SharePoint Server 2013 and enable Excel Services.
2. Install Analysis Services in SharePoint mode, and grant the SharePoint farm and services accounts server administrator rights in Analysis Services.

   For both versions of SharePoint, the PowerPivot installation process starts by selecting the setup role of **SQL Server PowerPivot for SharePoint** in the SQL Server Installation wizard or use a SQL Server command prompt installation.

3. For SharePoint 2013, you can extend the PowerPivot features and experience. Download and run **spPowerPivot.msi** to add server-side data refresh processing, collaboration, and management support for PowerPivot workbook. For more information, see Microsoft SQL Server 2014 PowerPivot for Microsoft® SharePoint.

   Run the PowerPivot for SharePoint 2013 installation package **spPowerPivot.msi** on each server in the SharePoint farm to ensure the correct version of the data providers are installed.

4. To configure PowerPivot for SharePoint 2013, use **PowerPivot for SharePoint 2013 Configuration** tool.

   The SQL Server installation wizard installs two PowerPivot Configurations tools. One of the configuration tools supports SharePoint 2013 and the other tool supports SharePoint 2010.

5. Configure Excel Services in SharePoint Server 2013 to use the Analysis Services instance. For more information, see the section "Configure Basic Analysis Services SharePoint Integration" in PowerPivot for SharePoint 2013 Installation.and Manage Excel Services data model settings (SharePoint Server 2013) (https://technet.microsoft.com/library/jj219780.aspx).
6. For more information, see PowerPivot for SharePoint 2013 Installation.

PowerPivot for SharePoint 2010

For SharePoint 2010, it is required that the PowerPivot for SharePoint Installation is run on a server that already has SharePoint 2010 installed or will be installed. The PowerPivot architecture for SharePoint 2010 runs **inside** the farm and requires SharePoint on the server that PowerPivot for SharePoint is installed.

1. Install Analysis Services in SharePoint mode, and grant the SharePoint farm and services accounts server administrator rights in Analysis Services.

A SharePoint 2010 deployment does not support spPowerPivot.msi, and the .msi is **not** required with SharePoint 2010.

For both versions of SharePoint, the PowerPivot installation process starts by selecting the setup role of **SQL Server PowerPivot for SharePoint** in the SQL Server Installation wizard or use a SQL Server command prompt installation.

2. The SQL Server installation wizard installs two PowerPivot Configurations tools. One of the configuration tools supports SharePoint 2013 and the other tool supports SharePoint 2010.

   To configure PowerPivot for SharePoint 2010, use the **PowerPivot Configuration Tool** .

3. For more information, see PowerPivot for SharePoint 2010 Installation.

Reporting Services for SharePoint 2010 and 2013

1. The installation for Reporting Services in SharePoint mode is unchanged from the previous release.

   The Reporting Services installation steps for SharePoint 2010 and SharePoint 2013 are very similar. Important notes regarding SharePoint versions are:

   - See the supported combinations of the following:
     - The version of Reporting Services.
     - The Reporting Services add-in for SharePoint Products.
     - The version of the SharePoint product.

     Supported Combinations of SharePoint and Reporting Services Server and Add-in (SQL Server 2014)

   - Reporting Services on SharePoint 2010 requires the SharePoint 2010 Service Pack 2 (SP2).

   a. Install Reporting Services in SharePoint mode. Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013) and Install Reporting Services SharePoint Mode for SharePoint 2010.

   b. Install the Reporting Services add-in for SharePoint products (rsSharePoint.msi). See Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013). For the current version of the Reporting Services add-in for SharePoint, see Where to find the Reporting Services add-in for SharePoint Products.

   c. Configure the Reporting Services SharePoint service and at least one Reporting Services service application. For more information, see the section "Create a Reporting Services Service Application" in Install Reporting Services SharePoint Mode for SharePoint 2013.

Overview of Database-attach Upgrade and SharePoint 2013

SharePoint 2013 does not support in-place upgrade. However **database-attach upgrade is supported**.

If you have an existing PowerPivot installation integrated with SharePoint 2010, you cannot in-place upgrade the SharePoint server. However, you can complete the following steps as part of a SharePoint database-attach upgrade:

1. Install a new SharePoint Server 2013 farm.
2. Complete a SharePoint database-attach upgrade, and migrate your PowerPivot related content databases to the SharePoint 2013 farm.
3. Install an instance of SQL Server Analysis Services in SharePoint mode and grant the SharePoint farm and services accounts, server administrator rights in Analysis Services.

4. Install the PowerPivot for SharePoint 2013 installation package **spPowerPivot.msi** on each server in the SharePoint farm.

5. In SharePoint 2013 Central Administration, configure Excel Services to use the Analysis Services server running in SharePoint mode created in step 3.

   To migrate refresh schedules, configure the PowerPivot service application.

> **NOTE**
> For more information on PowerPivot and SharePoint database-attach upgrade, see the following:

- Migrate PowerPivot to SharePoint 2013
- Overview of the upgrade process to SharePoint 2013.
- Clean up preparations before an upgrade to SharePoint 2013.
- Upgrade databases from SharePoint 2010 to SharePoint 2013.

See Also

Where to find the Reporting Services add-in for SharePoint Products Supported Combinations of SharePoint and Reporting Services Server and Add-in (SQL Server 2014) Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013)

Deployment Topologies for SQL Server BI Features in SharePoint

4/14/2020 • 8 minutes to read • Edit Online

This topic describes common topologies for installing the SQL Server Business Intelligence features Reporting Services and PowerPivot for SharePoint 2013 in SharePoint 2010 and SharePoint 2013 environments. For example single server and three tier installations.

Applies to: SharePoint 2013 | SharePoint 2010

In this topic:

- SharePoint 2013 Example Deployment Topologies
  - PowerPivot for SharePoint 2013 and Reporting Services Three Server Deployment
  - PowerPivot for SharePoint 2013 Single Server Deployment
  - PowerPivot for SharePoint 2013 Two Server Deployment
  - PowerPivot for SharePoint 2013 Three Server Deployment
  - PowerPivot for SharePoint 2013 and Reporting Services Single Server Deployment
  - PowerPivot for SharePoint 2013 and Reporting Services Two Server Deployment
- SharePoint 2010 Example Deployment Topologies
  - Single Server Deployments
  - Two-Tier Deployment
  - Three-Tier Deployment
  - Three-Tier Scale-Out Deployment

SharePoint 2013 Example Deployment Topologies

The SQL Server setup option **PowerPivot for SharePoint** has no dependencies on SharePoint. It does not use the SharePoint object model or interfaces to support integration. Therefore, Analysis Services can be installed on any computer running Windows Server 2008 R2 or later version. It can be but does not have to be an application server in a SharePoint farm. One of the configuration steps is to point Excel Services to the server running Analysis Services. For load balancing and fault tolerance, it is recommended to install and register multiple Analysis Services servers running in SharePoint mode.

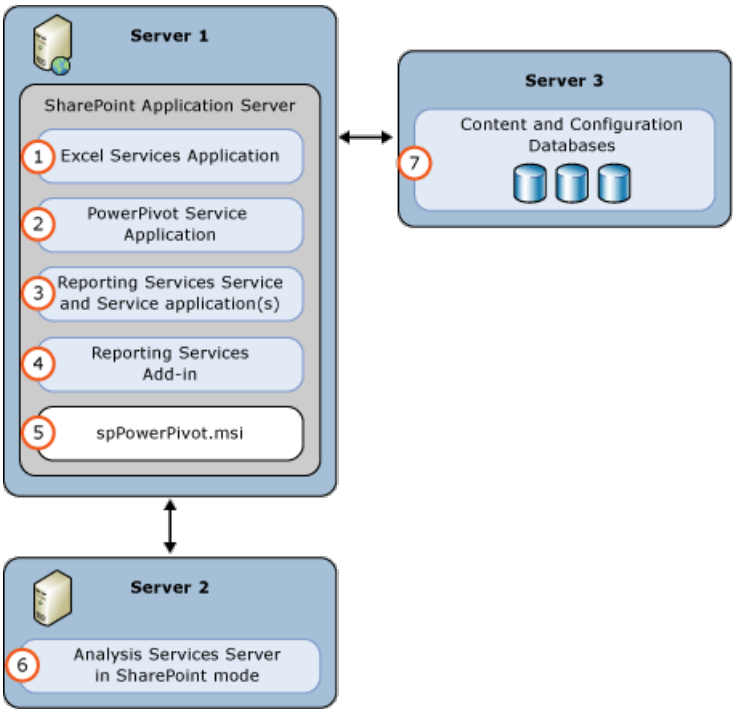
Reporting Services SharePoint mode requires SharePoint server 2013 and utilizes the SharePoint Service Application architecture.

The following sections illustrate typical deployment topologies:

PowerPivot for SharePoint 2013 and Reporting Services Three Server Deployment

In the following three-server deployment, the SQL Server Database Engine, Analysis Services server running in SharePoint mode, and SharePoint, each run on a separate server. The PowerPivot for SharePoint 2013 2013

installer package (**spPowerPivot.msi**) must be run on the SharePoint server.



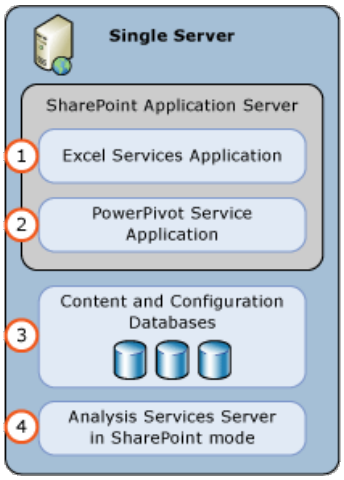
(1)	Excel Service Application. The service application is created as part of the SharePoint installation.
(2)	PowerPivot Service Application. Default name is **Default PowerPivot Service Application**.
(3)	Reporting Services service application.
(4)	Install the reporting services add-in for SharePoint from either the SQL Server 2014 installation media or the SQL Server 2014 feature pack.
(5)	Run the **spPowerPivot.msi** to install data providers, the PowerPivot configuration tool, PowerPivot Gallery, and schedule data refresh.
(6)	Analysis Services Server in SharePoint Mode. Configure the Excel Services Application **Data Model Settings** to use this server.
(7)	The SharePoint content, configuration, and service application databases.

Submit feedback and contact information through Microsoft SQL Server Connect (https://connect.microsoft.com/SQLServer/Feedback).

PowerPivot for SharePoint 2013 Single Server Deployment

A single server deployment is useful for testing purposes but it is not recommended for production deployments.

The following diagram illustrates the components that are part of a single server Analysis Services deployment.



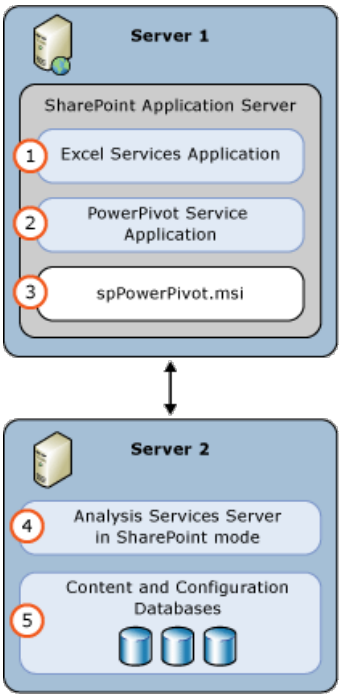
(1)	Excel Service Application. The service application is created as part of the SharePoint installation.
(2)	PowerPivot Service Application. Default name is **Default PowerPivot Service Application**.
(3)	The SharePoint content, configuration, and service application databases.
(4)	An Analysis Services Server in SharePoint Mode. Configure the Excel Services Application **Data Model Settings** to use this server.

PowerPivot for SharePoint 2013 Two Server Deployment

In the following two-server deployment, the SQL Server Database Engine and Analysis Services in SharePoint mode run on a separate server than SharePoint. For SharePoint 2013, the Microsoft SQL Server 2014 PowerPivot for SharePoint 2013 installer package (**spPowerPivot.msi**) is installed on the SharePoint server.

PowerPivot for SharePoint 2013 extends SharePoint Server 2013 to add server-side data refresh processing, data providers, PowerPivot Gallery, and management support for PowerPivot workbooks and Excel workbooks with advanced data models.

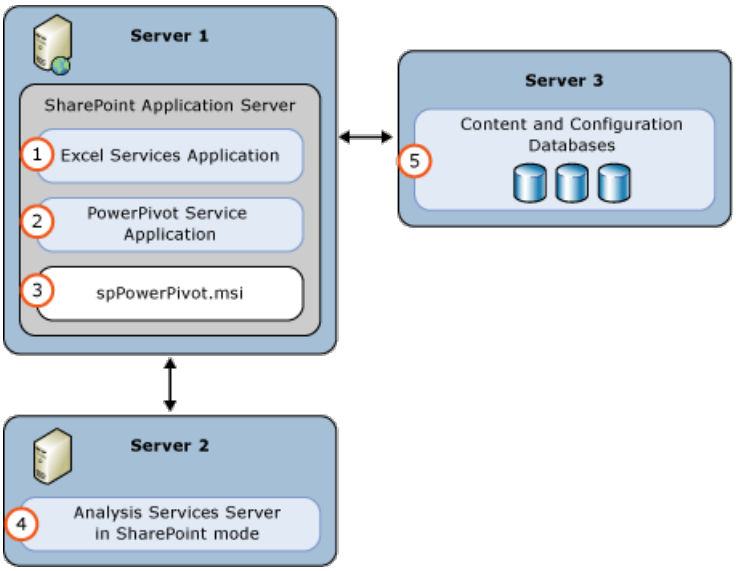
The installer package is available as part of the SQL Server 2014 feature pack. The feature pack can be downloaded from the Microsoft download center at Microsoft® SQL Server® 2014 PowerPivot® for Microsoft® SharePoint® (HYPERLINK "https://go.microsoft.com/fwlink/?LinkID=296473" \t "_blank" https://go.microsoft.com/fwlink/?LinkID=296473).



(1)	Excel Service Application. The service application is created as part of the SharePoint installation.
(2)	PowerPivot Service Application. Default name is **Default PowerPivot Service Application**.
(3)	RUN the **spPowerPivot.msi** to install data providers, the PowerPivot configuration tool, PowerPivot Gallery, and schedule data refresh.
(4)	An Analysis Services Server in SharePoint Mode. Configure the Excel Services Application **Data Model Settings** to use this server.
(5)	The SharePoint content, configuration, and service application databases.

PowerPivot for SharePoint 2013 Three Server Deployment

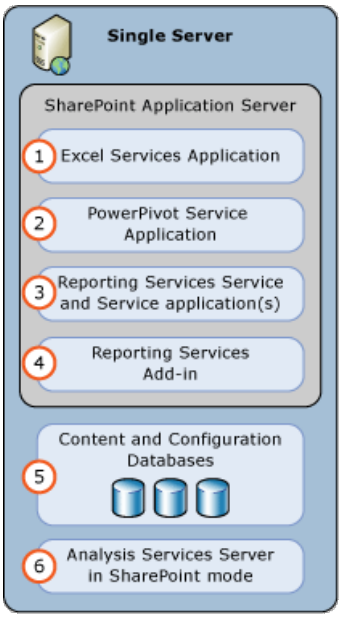
In the following three-server deployment, the SQL Server Database Engine, Analysis Services server running in SharePoint mode, and SharePoint, each run on a separate server. The PowerPivot for SharePoint 2013 installer package (spPowerPivot.msi) must be installed on the SharePoint server.



(1)	Excel Service Application. The service application is created as part of the SharePoint installation.
(2)	PowerPivot Service Application. Default name is **Default PowerPivot Service Application**.
(3)	RUN the spPowerPivot.msi to install data providers, the PowerPivot configuration tool, PowerPivot Gallery, and schedule data refresh.
(4)	An Analysis Services Server in SharePoint Mode. Configure the Excel Services Application **Data Model Settings** to use this server.
(5)	The SharePoint content, configuration, and service application databases.

PowerPivot for SharePoint 2013 and Reporting Services Single Server Deployment

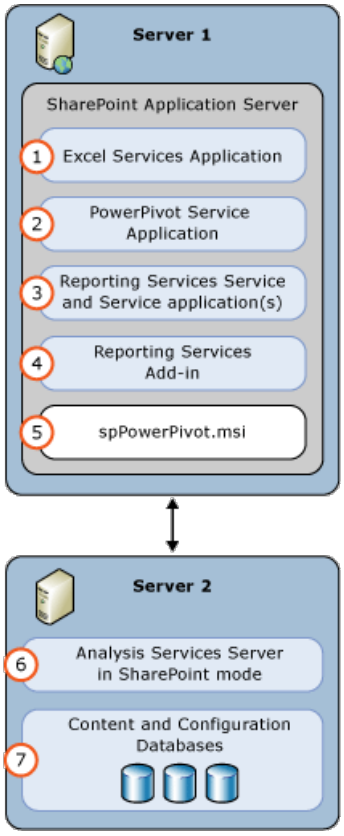
A single server deployment is useful for testing purposes but it is not recommended for production deployments.



(1)	Excel Service Application. The service application is created as part of the SharePoint installation.
(2)	PowerPivot Service Application. Default name is **Default PowerPivot Service Application**.
(3)	Reporting Services service application.
(4)	Install the reporting services add-in for SharePoint from either the SQL Server 2014 installation media or the SQL Server 2014 feature pack.
(5)	The SharePoint content, configuration, and service application databases.
(6)	Analysis Services Server in SharePoint Mode. Configure the Excel Services Application **Data Model Settings** to use this server.

PowerPivot for SharePoint 2013 and Reporting Services Two Server Deployment

In the following two-server deployment, the SQL Server Database Engine and Analysis Services server running in SharePoint mode run on a separate server from SharePoint. The PowerPivot for SharePoint 2013 installer package **(spPowerPivot.msi)** must be run on the SharePoint server.

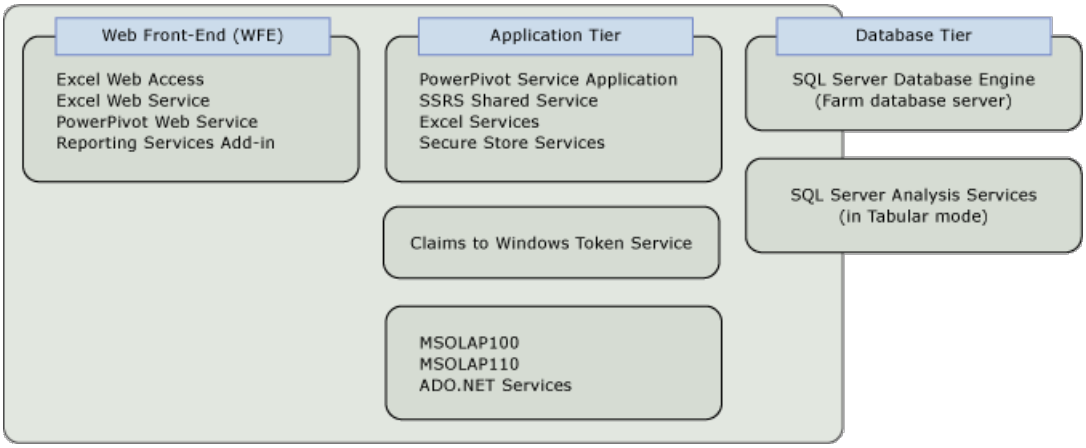


(1)	Excel Service Application. The service application is created as part of the SharePoint installation.
(2)	PowerPivot Service Application. Default name is **Default PowerPivot Service Application**.

(3)	Reporting Services service application.
(4)	Install the reporting services add-in for SharePoint from either the SQL Server 2014 installation media or the SQL Server 2014 feature pack.
(5)	RUN the **spPowerPivot.msi** to install data providers, the PowerPivot configuration tool, PowerPivot Gallery, and schedule data refresh.
(6)	Analysis Services Server in SharePoint Mode. Configure the Excel Services Application **Data Model Settings** to use this server.
(7)	The SharePoint content, configuration, and service application databases.

SharePoint 2010 Example Deployment Topologies

The following diagram shows which services and providers run on each tier. Notice that the diagram includes several built-in services; these services are required for some SQL Server BI scenarios. Excel Services, Secure Store Services, and the Claims to Windows Token Service are either required by or recommended for a PowerPivot for SharePoint or Reporting Services deployment in SharePoint. Additionally, the MSOLAP OLE DB providers and ADO.NET Services are required for some PowerPivot data access scenarios. Optionally, you can install Analysis Services on the data tier, if you want to build Power View reports based on tabular model databases that are hosted outside of SharePoint.



Single Server Deployments

You can install all server components, including the data tier, on a single computer. This deployment configuration is useful if you are evaluating the software or developing custom applications that include Reporting Services in SharePoint mode. This deployment is the simplest to configure. Because all the components are installed on the same computer, it also uses the least amount of licenses. Reporting Services, PowerPivot for SharePoint, and the Database Engine are installed as a single licensed copy of SQL Server.

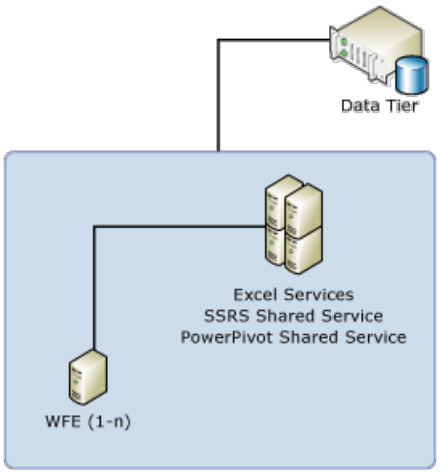
To install all features on a single server, you install Reporting Services and PowerPivot for SharePoint sequentially, on the same physical server. For instructions on a standalone server configuration, see Deployment Checklist: Reporting Services, Power View, and PowerPivot for SharePoint.

Two-Tier Deployment

A two-tier deployment is typically SharePoint Server 2010 on one computer and the SQL Server Database Engine on the second computer. Moving the data tier to a dedicated server is the most common configuration for a 2-computer farm. In a two-tier farm, you install both Reporting Services and PowerPivot for SharePoint on the SharePoint server. All web services on the front-end and shared services in the application tier run on the same physical server. Installation steps for a 2-tier deployment are very similar to a standalone deployment, in that you install Reporting Services and PowerPivot for SharePoint sequentially, on the same physical server.

Three-Tier Deployment

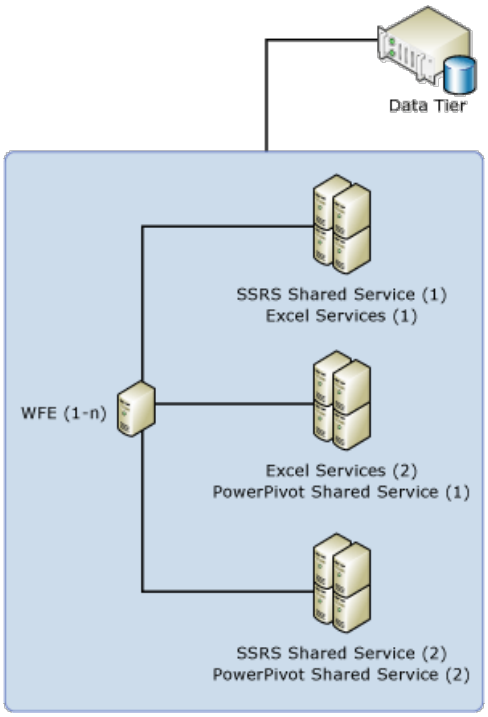
A three-tier deployment typically separates web front-end services from processing or memory-intensive applications. On this topology, you install Reporting Services and PowerPivot for SharePoint on just the application server. Web services that run on the web front-end are installed via solutions that are deployed to applications in the farm, during server configuration, as a post-installation task. The following diagram illustrates a 3-tier deployment.



Three-Tier Scale-Out Deployment

This topology illustrates a scale-out deployment that runs the same shared service on multiple servers, servicing a larger volume of requests and providing greater processing power for PowerPivot data or Reporting Services reports. In the diagram below, there are three application server clusters, each running a different combination of shared services. In a SharePoint environment, service discovery and availability is built into the farm. Load balancing across multiple physical servers running the same shared service application is part of the shared service architecture.

When deploying a multi-server farm, be sure to follow the instructions in this SharePoint article: Multiple Servers for a three-tier farm (SharePoint Server 2010).



See Also

Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013)
PowerPivot for SharePoint 2013 Installation
PowerPivot for SharePoint 2010 Installation

Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm

2/28/2020 • 5 minutes to read • Edit Online

This topic summarizes feature availability based on the versions and editions of the software you are using. It also explains SharePoint 2010 installation requirements for using specific SQL Server features. For information related to SharePoint 2013, see Deployment Topologies for SQL Server BI Features in SharePoint.

In this topic:

- General SharePoint 2010 Requirements
- SharePoint 2010 Service Pack 1 (SP1)
- SharePoint Editions and BI Feature support
- SharePoint 2010 Products Preparation Tool
- Requirements and suggestions for running the SharePoint installation

General SharePoint 2010 Requirements

- SharePoint 2010 products are 64-bit only. If you have a 32-bit installation of a previous version of SharePoint and Reporting Services installed in SharePoint integrated mode, you will not be able to upgrade to SharePoint 2010. For more information, review the SharePoint documentation.
- Reporting Services includes an add-in for SharePoint products. Supported configurations for the add-in and report server are available at a more granular level than what is indicated here. For more information, see Supported Combinations of SharePoint and Reporting Services Server and Add-in (SQL Server 2014).
- SharePoint developer tools only support a SharePoint standalone configuration. For more information, see the SharePoint documentation: Requirements for Developing SharePoint Solutions.

SharePoint Editions and BI Feature support

Some SQL Server 2014 Business Intelligence features are only supported on specific editions of SharePoint products.

SUPPORTED FEATURES	SHAREPOINT PRODUCT
Power View, a feature of SQL Server 2014Reporting Services Add-in for MicrosoftSharePoint Server 2010 Enterprise Edition. Reporting Services Data Alerts. PowerPivot.	SharePoint Server 2010 Enterprise Edition.
General Reporting Services report viewing and feature integration with SharePoint.	SharePoint Server 2010 Standard and Enterprise Editions. SharePoint Foundation 2010.

For more information, see Features Supported by the Editions of SQL Server 2012.

SharePoint 2010 Service Pack 1 (SP1)

It is recommended you update your SharePoint 2010 installation to SharePoint 2010 Service Pack 1 (SP1). SharePoint SP1 is required in the following cases:

- You want use SQL Server 2012 version of the database engine for the SharePoint content databases or Reporting Services catalog databases.
- You want to use PowerPivot and the PowerPivot configuration tool.

One of the primary reasons SP1 is required for SharePoint installations running with SQL Server 2012 is the database engine feature **sp_dboption**, that was deprecated in a previous release, is discontinued in the SQL Server 2012 release. For more information, see Discontinued Database Engine Functionality in SQL Server 2014

SharePoint 2010 SP1 Installation Guidance

Download SharePoint Server 2010 SP1 and apply it on all servers in the farm.

> **NOTE**
> On an existing farm, you will need to use one of the following **additional** steps to complete the SharePoint SP1 upgrade. For more information, see Known issues when you install Office 2010 SP1 and SharePoint 2010 SP1 and Description of SharePoint Server 2010 SP1:

- **SharePoint Products Configuration Wizard:** Run the wizard to complete the SP1 upgrade and configuration.
- **Complete the upgrade with psconfig:** Run the command `psconfig -upgrade` to complete the SP1 upgrade

For more information, see the "upgrade" section of (SharePoint Server 2010) and Resource center: Updates for SharePoint 2010 Products

SharePoint Installation with SQL Server BI Features

SharePoint 2010 Products Preparation Tool

The SharePoint Products Preparation Tool enables server roles in the operating system and installs other prerequisite software required by a SharePoint installation. The following table identifies additional steps to configure your server for SQL Server 2014.

COMPONENT	ACTION
Reporting Services Add-in	The SharePoint 2010 Products Preparation tool installs the SQL Server 2008 R2 version of the Reporting Services add-in. SQL Server 2014 includes a newer version of the add-in that is required for SQL Server 2014 features. The add-in can be installed using the SQL Server installation wizard as well as downloaded from MSDN. For more information on where to get the current version of the add-in and how to install it, see Where to find the Reporting Services add-in for SharePoint Products and Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013).

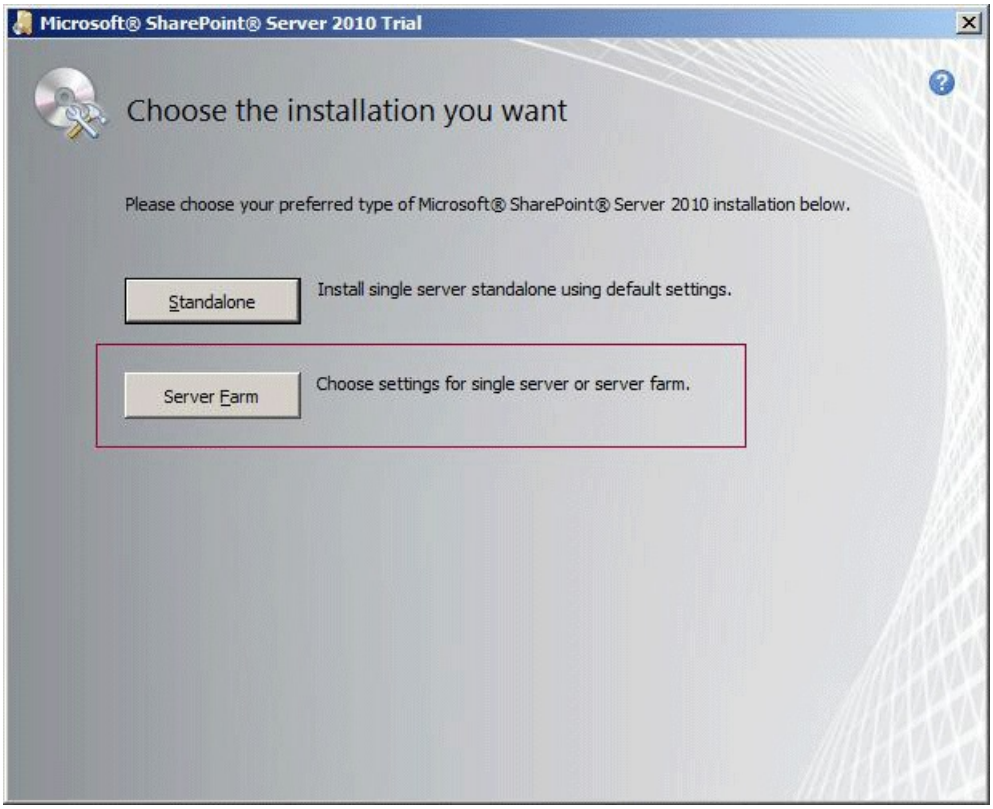
COMPONENT	ACTION
Analysis Services OLE DB Provider (MSOLAP)	SharePoint 2010 installs the SQL Server 2008 version of the OLE DB provider as part of an Excel Services deployment. This version does not support PowerPivot data access. You should install later versions of the provider on SharePoint servers that support PowerPivot data connections. For more information, see Install the Analysis Services OLE DB Provider on SharePoint Servers
ADO.NET Services	SharePoint 2010 lists ADO.NET services in the list of prerequisites, but the Prerequisite installer does not install it. To add ADO.NET services, you must install it manually. Installing ADO.NET services is required if you want to use SharePoint lists as data feeds to PowerPivot workbooks or Reporting Services reports. For instructions, see Install ADO.NET Data Services to support data feed exports of SharePoint lists.

Requirements and suggestions for running the SharePoint installation

Which SQL Server 2014 features you install and the installation order of those features will determine what levels of integration with SharePoint are possible. For example, while some level of feature integration is available through Reporting Services on a SharePoint server that uses a built-in database, most feature integration scenarios require a farm installation of SharePoint, because only a farm installation provides the infrastructure required for some BI features.

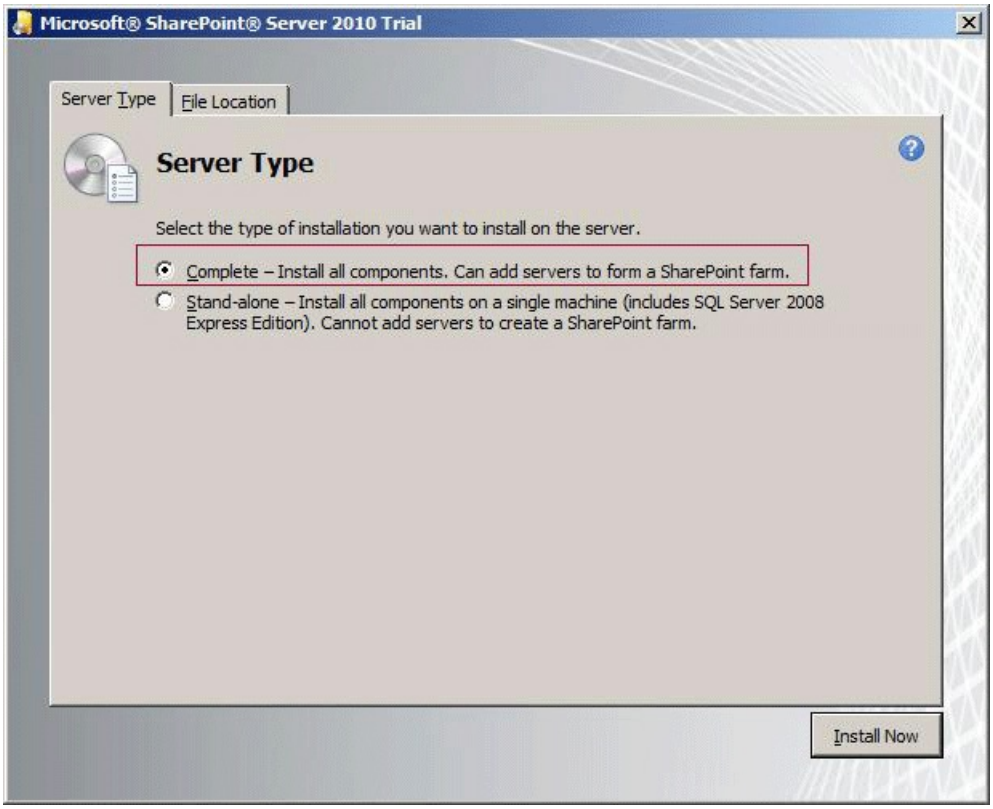
A SharePoint farm can be a single server or multiple servers. What sets a farm apart from a standalone installation is the presence of additional infrastructure, such as the Claims to Windows Token Service.

A farm is enabled when you choose the **Server Farm** option in SharePoint setup. You must have a farm installation if you want to use PowerPivot for SharePoint or Power View. The Standalone SharePoint installation is supported and common for development environments of Reporting Services. However a SharePoint farm installation is recommended for a production environment.



A complete server installation is also required for PowerPivot for SharePoint or the Reporting Services shared

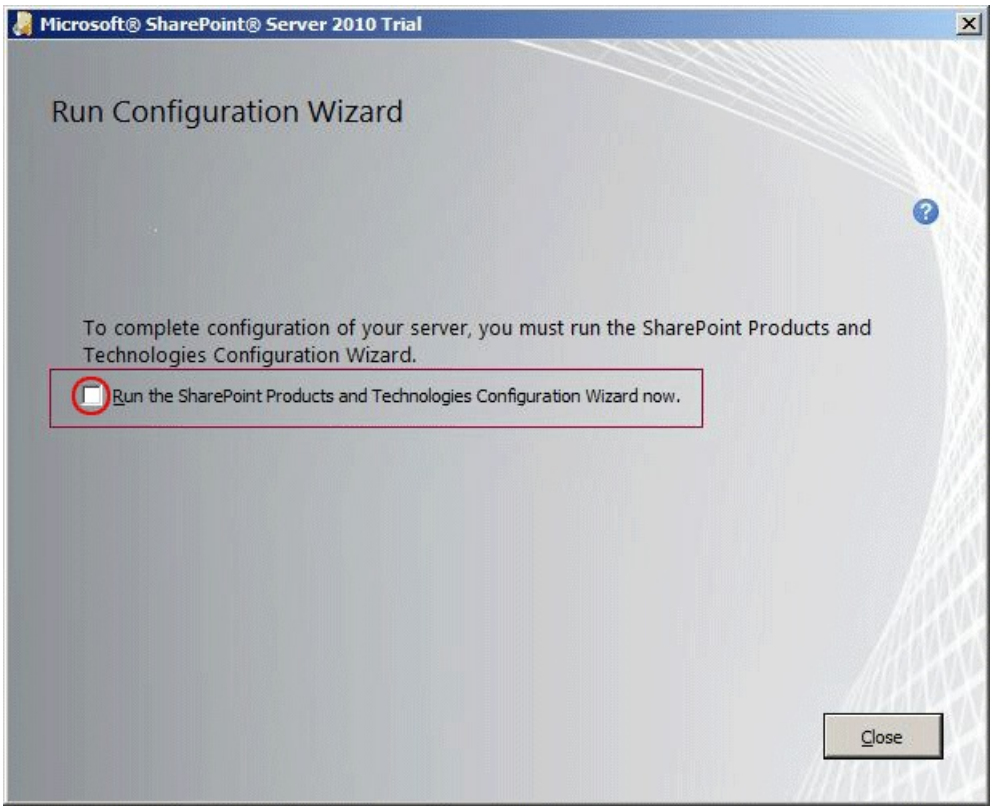
service.



The last page of the setup wizard prompts you to configure the farm and optionally configure a default web application and root site collection. Most setup users follow this installation path. If for some reason you do not configure the farm, you can use the farm configuration option in the PowerPivot Configuration Tool to configure the farm.

In previous releases, leaving the farm un-configured was recommended for installing PowerPivot because it reduced the steps needed when if you used SQL Server installation options to install PowerPivot for SharePoint in a ready-to-use state. In this release, it is no longer important. You can use the PowerPivot Configuration Tool to tune an existing farm so that it uses settings recommended for a PowerPivot for SharePoint installation.

Reporting Services can be installed into an existing SharePoint farm. You can install Reporting Services or SharePoint in any order, but the configuration of Reporting Services will vary depending on the order. For more information, see Add an Additional Report Server to a Farm (SSRS Scale-out) and Install Reporting Services SharePoint Mode for SharePoint 2010



See Also

Installation and Deployment for SharePoint Server 2010 Multiple Servers for a three-tier farm (SharePoint Server 2010)

Checklists for Installing BI Features with SharePoint

10/3/2019 • 2 minutes to read • Edit Online

This section provides checklists for installing SQL Server BI features in different combinations or in scale-out topologies. If you already know how to run a Setup program to install a BI feature, a checklist is the next step towards understanding the recommended installation order, and whether additional configuration changes are required to enable feature integration scenarios.

Checklists in this section

Deployment Checklist: Reporting Services, Power View, and PowerPivot for SharePoint

Deployment Checklist: Scale-out by adding PowerPivot Servers to a SharePoint 2010 farm

Deployment Checklist: Multi-Server Installation of PowerPivot for SharePoint 2010

Deployment Checklist: Install Reporting Services into an Existing SharePoint Farm

Deployment Checklist: Reporting Services, Power View, and PowerPivot for SharePoint

10/3/2019 • 7 minutes to read • Edit Online

Use the following checklist to install these BI features in the same SharePoint farm: PowerPivot for SharePoint, Report Builder, and Power View. Although this checklist recommends a specific installation order, in practice you can install these features in almost any order. This checklist assumes installation of the following products or features:

1. SharePoint Server 2010 with Service Pack 1 (SP1)
2. SQL Server 2012 Database Engine
3. SQL Server 2012 Reporting Services and Reporting Services Add-in
4. SQL Server 2012 PowerPivot for SharePoint

After you install these features, you will be able to do the following.

- Access the PowerPivot workbooks that you created in PowerPivot for Excel from SharePoint sites.
- Build interactive Power View reports based on PowerPivot workbooks in SharePoint.
- Create Report Builder reports when you launch Report Builder in SharePoint.

> **NOTE**
>
> PowerPivot for SharePoint requires that you install SharePoint 2010 Service Pack 1 (SP1) on the SharePoint farm. If you are installing the software for evaluation purposes, consider starting with a clean server so that you can avoid the overhead of a farm upgrade step. Upgrading a farm is impactful to SharePoint operations and typically requires careful planning. If your goal is to install PowerPivot for SharePoint as quickly as possible, follow this approach: install SharePoint 2010, install SharePoint 2010 SP1, then configure the farm in a later step. Upgrade is avoided because the farm is not yet configured when you install SharePoint 2010 SP1.
>
> In this checklist, the farm configuration step is assumed during PowerPivot for SharePoint configuration, using the PowerPivot Configuration Tool. Alternatively, you can use the SharePoint Product Configuration wizard if you prefer that approach. Both approaches result in an operational farm that supports PowerPivot for SharePoint.

Prerequisites

You must be a local administrator to run SQL Server Setup.

SharePoint Server 2010 enterprise edition is required for PowerPivot for SharePoint. You can also use the evaluation enterprise edition.

SharePoint Server 2010 SP1 must be installed. Without it, you cannot configure the farm to use SQL Server 2012 features.

The computer must be joined to a domain.

You must have one or more domain user accounts to provision the services. You will need domain user accounts for the following services: SharePoint web services and administrative services, Reporting Services, Analysis Services, Excel Services, Secure Store Services, and PowerPivot System Service. Domain accounts are required by the managed accounts feature in SharePoint. The Database Engine can be provisioned using a virtual account, but

all other services should run as a domain user.

The PowerPivot instance name must be available. You cannot have an existing PowerPivot named instance on the computer on which you are installing PowerPivot for SharePoint.

If you are installing PowerPivot for SharePoint on an existing farm, you must have one or more SharePoint web applications that are configured for classic mode authentication. PowerPivot data access will only work if the web application supports classic mode authentication. For more information about classic mode requirements, see PowerPivot Authentication and Authorization.

Review the following additional topics to understand system and version requirements:

- Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm

Steps

The following steps assume that an administrator is installing and configuring the server. The Setup user in SharePoint is also a farm administrator and often the primary site administrator for the default site collection. If you are dividing the following steps among several people, additional permissions might be required in order for the following steps to work.

STEP	LINK
Run the SharePoint 2010 Products Preparation Tool	You must have the SharePoint 2010 installation media. The preparation tool is PrerequisiteInstaller.exe on the installation media.
Install SharePoint Server 2010 enterprise or enterprise evaluation edition.	When installing SharePoint, you can choose to configure the farm later by not running the SharePoint 2010 Product Configuration Wizard after Setup is finished. Waiting to configure the farm will allow you to use a SQL Server 2012 Database Engine instance, which is installed in a later step, as the farm's database server. To configure the farm, you will use the PowerPivot Configuration Tool. It includes actions for provisioning the farm if the farm is not yet configured.
Install SharePoint Server 2010 SP1.	Download SP1 from https://support.microsoft.com/kb/2460045.

STEP	LINK
Run SQL Server 2012 Setup to install the Database Engine and PowerPivot for SharePoint.	Install PowerPivot for SharePoint 2010 Step 1 explains how to install PowerPivot for SharePoint. In this step, be sure to click the checkbox on the Setup Role page that adds the Database Engine to the role. Doing so adds the Database Engine to your installation so that you can use it as the farm's database server when you configure the farm in the next step. However, if the farm is already configured, you can skip this step. Step 2 asks you to configure the server. For this step, choose the PowerPivot Configuration tool. Although several approaches are available, using the configuration tool is the most efficient approach for a standalone installation. If SharePoint 2010 is installed but not configured, the tool pre-selects actions that will create the farm, a default web application, and a root site collection. Be sure to leave these options selected so that the farm will be created. If you already configured the farm, the tool will omit these actions, and offer just the actions that are necessary for configuring PowerPivot for SharePoint. Step 3 instructs you to install the SQL Server 2008 R2 version of the Analysis Services OLE DB Provider. This step is important for supporting versions of a workbook that were created in the 2008 R2 version of PowerPivot for Excel.
Verify the farm is operational.	First, start Central Administration and confirm that it is available. Next, open the team site by entering http://localhost. You should see a SharePoint team site.
Verify that PowerPivot for SharePoint is operational.	Verify a PowerPivot for SharePoint Installation This task confirms PowerPivot data access using a sample workbook that you upload.
Run SQL Server 2012 Setup to install and configure Reporting Services and the Reporting Services Add-in.	Install Reporting Services SharePoint Mode for SharePoint 2010 Optionally, while installing Reporting Services, you can add an additional Analysis Services instance to the Setup feature tree if you want a second resource for hosting tabular data. The additional Analysis Services instance would be used to host tabular model databases that you create in SQL Server Data Tools (SSDT). Tabular databases are a valid data source for Power View reports. Install Analysis Services in Tabular Mode
Verify that Reporting Services is operational.	Verify a Reporting Services Installation

STEP	LINK
(Site Administrators) Configure SharePoint permissions.	Contribute permissions are required to add, edit, or delete items in SharePoint libraries. View permission level is sufficient for read-only access to reports and PowerPivot workbooks that present embedded data. PowerPivot workbooks that are accessed as external data sources (where the workbook URL is a connection string in another workbook or report) require Read permissions, which is higher than View permissions. BI semantic model connections also required Read permissions. You might need to create new permission levels or SharePoint groups to get the correct permissions in place.
(Site Administrators) Extend document libraries	Extend document libraries to use BI content types: BI semantic model connections, Reporting Services Shared Data Sources, Report Builder reports: 1) **Enable content type management**. In Shared Documents or another document library, in the Library tab, click **Library Settings**. Under General Settings click **Advanced settings**. In Content Types, select **Yes** to allow management of content types, and then click **OK**. 2) **Select the BI content types**. In the Library tab, click **Library Settings**. Under Content Types, click **Add from existing site content types**. From the Business Intelligence content type group, add **BI Semantic Model Connection File** and **Report Data Source**. Optionally, you can also add other Reporting Services content types, such as Report Model, to enable additional report building scenarios. For more information, see Add a BI Semantic Model Connection Content Type to a Library (PowerPivot for SharePoint) and Add Report Server Content Types to a Library (Reporting Services in SharePoint Integrated Mode).
(Site Administrators) Create data connection files that are used to launch Power View.	You must create a BI semantic model connection (.bism) or a Reporting Services shared data source (.rsds) as a data source for Power View. After you create a data connection file, you can launch Power View using the data connection as its data source. Create a BI Semantic Model Connection to a PowerPivot Workbook Create a BI Semantic Model Connection to a Tabular Model Database Note: Power View is available because you installed the SQL Server 2012 version of Reporting Services and configured the server as a shared service. If you installed Reporting Services and configured it for the SQL Server 2008 level of integration, Power View is not available.

See Also

Features Supported by the Editions of SQL Server 2012

Deployment Checklist: Scale-out by adding PowerPivot Servers to a SharePoint 2010 farm

10/3/2019 • 5 minutes to read • Edit Online

If you anticipate a high volume of requests for PowerPivot query processing in a SharePoint farm, you can add an extra PowerPivot for SharePoint instance to seamlessly add new query and data processing support.

After you install a new instance, you will have additional capacity for querying PowerPivot data or processing PowerPivot data refresh jobs. Optionally, you will have the choice of configuring each server to handle one type of request: query or data refresh.

Prerequisites

SharePoint Server 2010 is installed and configured.

SharePoint Server 2010 SP1 is applied and the farm is upgraded.

The existing PowerPivot for SharePoint instance in the farm is SQL Server 2012 (either a new installation or upgraded from SQL Server 2008 R2).

The computer on which you are installing the new SQL Server 2012 PowerPivot for SharePoint server is joined to the farm. The computer and the other servers in the farm must be in the same domain.

Review the following additional topics to understand system and version requirements:

- Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm

> **NOTE**
> In a multi-server farm, all PowerPivot for SharePoint instances must be at the same version. If you applied service packs or updates to other PowerPivot servers that are already in the farm, the new instance you are adding must be updated to the same version as the existing instance that is already in the farm. The new instance will be unavailable until the updates have been applied.

Steps

Use this checklist to add additional PowerPivot servers to a SharePoint farm. These instructions assume that you already have a PowerPivot for SharePoint server in the farm, and that you are adding a second server to handle additional processing load. Except for differences in installation requirements, post-install configuration, and verification, the steps for deploying a scale-out solution are identical to adding a single PowerPivot server to an existing farm.

STEP	LINK

STEP	LINK
Determine the service account of the Analysis Services instance that is already in the farm	Each additional instance you install must run under the same account as the first instance. Use either approach to determine the service account: In Central Administration, in the Security section, click **Configure Service Accounts**. Select **Windows Service - SQL Server Analysis Services**. After you select the service, the service account name will appear in the page. On a server that already has a PowerPivot service installation, open the **Services** console application in Administrative Tools. Double-click **SQL Server Analysis Services**. Click the **Log On** tab to view the service account. ** **Important** ** Only use Central Administration to change service accounts. If you use another tool or approach, permissions will not be updated correctly in the farm.
Run Setup to install a second instance of PowerPivot for SharePoint	Install PowerPivot for SharePoint 2010 Choose an application server that is joined to the farm, but does not have an existing PowerPivot instance on the server. During Setup, when prompted to specify a service account, enter the account from the previous step. All instances of the Analysis Services service must run under the same domain account. This requirement enables the use of the managed accounts feature in SharePoint that lets you update the password in one place for all service instances of the same type.
Configure the second instance	You can use either approach to configure the instance: PowerPivot Configuration Tools or PowerPivot Configuration using Windows PowerShell When configuring a second instance, you only need to provision the local services. All other configuration tasks (such as creating service applications or configuring data refresh) are performed during the initial configuration, and used by subsequent instances that you install.
Post-installation tasks	No further steps are specifically required. You do not need to create service applications, activate features, deploy solutions, or change service application identity. Existing Web applications and service applications will discover and use the new server software automatically. Optionally, if you installed a second server for the purpose of using one server for queries and one for data refresh, you can configure server instance properties now to specify the type of requests handled by each server. For more information, see Configure Dedicated Data Refresh or Query-Only Processing (PowerPivot for SharePoint).

STEP	LINK
Verify installation of the second instance	You can use the following steps to verify PowerPivot query processing on the server you just installed. 1) In Central Administration, open the Manage services on server page to confirm that the server and its services appear. -In Server, click the down arrow, click Change Server, and then select the server that has the new PowerPivot for SharePoint installation. -Verify that SQL Server Analysis Services and SQL Server PowerPivot System Service are started. 2) In Central Administration, stop other PowerPivot for SharePoint servers so that the server you just installed is the only one available. For more information, see Start or Stop a PowerPivot for SharePoint Server. 3) Click a PowerPivot workbook to open it from the library. 4) Click a slicer or pivot the data to initiate a query. The server will load PowerPivot data in the background. In the next step, you will connect to the server to verify the data is loaded and cached. 5) Start SQL Server Management Studio from the Microsoft SQL Server program group in the Start menu. If this tool is not installed on your server, you can skip to the last step to confirm the presence of cached files. 6) In Server Type, select **Analysis Services**. 7) In Server Name, enter **<server-name>\powerpivot**, where **<server-name>** is the name of the computer that has the new PowerPivot for SharePoint installation. 8) Click **Connect**. 9) In Object Explorer, click **Databases** to view the list of PowerPivot data files that are loaded. 10) On the computer file system, check the following folder to determine whether files are cached to disk. The presence of cached files is further verification that your deployment is operational. To view the file cache, go to the \Program Files\Microsoft SQL Server\MSAS11.POWERPIVOT\OLAP\Backup folder. 11) Restart the services that you stopped earlier.

See Also

Initial Configuration (PowerPivot for SharePoint)
PowerPivot for SharePoint 2010 Installation
PowerPivot Server Administration and Configuration in Central Administration

Deployment Checklist: Install Reporting Services into an Existing SharePoint Farm

10/3/2019 • 3 minutes to read • Edit Online

Reporting Services SharePoint report servers can be installed into a new SharePoint Farm or into an existing SharePoint farm. This topic describes the possible scenarios and best practices for installing Reporting Services into and existing SharePoint farm.

Prerequisites

Before you run Setup, review the following information:

STEP	LINK
Create or identify the accounts used in a report server deployment. You must have a service account for the Report Server service, and credentials for connecting to the report server database	
Decide on an instance of SQL Server to host the report server database. You can use a local or remote instance of SQL Server. You should choose an instance that is on a computer that has the storage capacity to accommodate your reports.	
(Optional) Find the name of the SMTP server or gateway that provides e-mail service to your organization if you want to use report server e-mail in subscriptions	Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager)
Note: If you are upgrading a computer from a previous CTP release SQL Server 2012 and you had made custom changes to the configuration files, you will need to make the same changes to the configuration files, following the upgrade to SQL Server 2014. The affected files are **web.config** and **client.config**.	

Installation Scenarios

The following table describes the possible scenarios when you are installing Reporting Services into and existing SharePoint Farm. Local mode allows reports to be rendered locally from the SharePoint document library, without integration with a Reporting Services report server. The Reporting Services add-in for SharePoint products is required but a Reporting Services report server is not. For more information on local mode, see Local Mode vs. Connected Mode Reports in the Report Viewer (Reporting Services in SharePoint Mode) and Where to find the Reporting Services add-in for SharePoint Products.

STARTING CONFIGURATION	WORKFLOW	ENDING CONFIGURATION	COMMENTS
SQL Server 2008 R2 in Local mode	Installation	Connected mode SQL Server 2014.	

STARTING CONFIGURATION	WORKFLOW	ENDING CONFIGURATION	COMMENTS
Connected Mode SQL Server 2008 or SQL Server 2008 R2	In place upgrade	Connected mode SQL Server 2014.	
Connected Mode SQL Server 2008 or SQL Server 2008 R2	Migration	Connected mode SQL Server 2014.	

Installation and in place Upgrade Checklist

The following table summarizes the steps, tools, and information you should review and use for the installation:

STEP	LINK
Installation and initial configuration	
Install the SharePoint add-in on all Web front-end (WFE) computers.	Add an Additional Reporting Services Web Front-end to a Farm
Install SQL Server SQL Server 2012 Reporting Services and the Database engine.	Install Reporting Services SharePoint Mode for SharePoint 2010
Create at least one SSRS service application and configure service app association.	See the 'Service Application' section in Install Reporting Services SharePoint Mode for SharePoint 2010
Additional Configuration	
Add SSRS content types to your document library.	Add Report Server Content Types to a Library (Reporting Services in SharePoint Integrated Mode)
Provision SQL Server Agent	Provision Subscriptions and Alerts for SSRS Service Applications
Configure e-mail settings for your Service application	Configure E-mail for a Reporting Services Service Application (SharePoint 2010 and SharePoint 2013)
Configure Claims to Windows Token Service (c2WTS)	Claims to Windows Token Service (C2WTS) and Reporting Services

Migration Checklist

The following is a list of the basic steps for a migration from an existing installation to a new installation.

STEP	LINK
Install and configure your new server. This includes the following: SharePoint Products Preparation Tool SharePoint 2010 Product SharePoint 2010 SP1 Reporting Services in SharePoint Mode Reporting Services add-in for SharePoint 2010 products	Install Reporting Services SharePoint Mode for SharePoint 2010
Create at least one Reporting Services service application	
Backup Reporting Services Databases	
Backup Reporting Services Encryption keys	
Restore Reporting Services database and Encryption keys	
Map all web applications to new Reporting Servicesservice application(s)	The new installation **is now functional**
Remove the Integration URL on the old server.	From SharePoint Central Administration, on the **General Application Settings** Page, click **Reporting Services Integration**.
Uninstall Reporting Services from the old installation, if desired.	

Next Steps

See Also

Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013)
Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm
Supported Combinations of SharePoint and Reporting Services Server and Add-in (SQL Server 2014)

Deployment Checklist: Multi-Server Installation of PowerPivot for SharePoint 2010

10/3/2019 • 13 minutes to read • Edit Online

This checklist guides you through the steps for adding PowerPivot for SharePoint to a three-tier SharePoint 2010 farm that you build from the ground up. A three-tier farm includes database, application, and web tiers. Adding PowerPivot to this topology requires that you run SQL Server Setup to install PowerPivot for SharePoint on the application tier. PowerPivot program files are added to the web tier, but only as a post-installation task when you the deploy web application solution. Although there are deployment steps, there is no separate installation step on either the web tier or data tier that you need to perform. The only installation step that you need to perform is installing PowerPivot for SharePoint on the application servers.

Applies to: SharePoint 2010

Prerequisites

You must be a local administrator to install SQL Server and SharePoint 2010.

The person who installs SharePoint must also configure the farm. To configure the farm, you must have a SQL Server login on the database server. The login must be assigned to the following roles: `dbcreator`, `securityadmin` and `public`.

You must have the enterprise or enterprise evaluation edition of SharePoint Server 2010.

The computer must be joined to a domain.

You must know which accounts you will use to run the Database Engine, the services in your farm, and Analysis Services in SharePoint integrated mode. Although you can change these accounts later, you will specify them for the first time during installation.

Service accounts that you specify during installation must be domain user accounts.

Before you begin installation, check your browser settings to verify you have an internet connection. The prerequisite installer opens an internet connection to download required software. You should make the following changes to ensure you get all of the required software:

- In Server Manager, temporarily disable Internet Explorer Enhanced Security Configuration to allow downloads to the server. For the purposes of downloading required software, you can turn off IE ESC for Administrators only.
- In Internet Explorer, you might also need to configure your browser to bypass a proxy server to allow localhost access to internet URLs.

    1. In Internet Explorer, on the Tools menu, click Internet Options.
    2. On the Connections tab, in the Local Area Network (LAN) settings area, click LAN Settings.
    3. In the Automatic configuration area, clear the Automatically detect settings check box.
    4. In the Proxy Server area, select the Use a proxy server for your LAN check box.
    5. Type the address of the proxy server in the Address box.

6. Type the port number of the proxy server in the Port box.
7. Select the Bypass proxy server for local addresses check box.
8. Click OK to close the Local Area Network (LAN) Settings dialog box.
9. Click OK to close the Internet Options dialog box.

Install a database server

This topic assumes your farm topology is based on the one described in the article Multiple servers for a three-tier farm. If you already have a farm that is operational, skip ahead to Install PowerPivot for SharePoint.

If you are just getting started with your topology, begin by installing a SQL Server Database Engine. These instructions result in a database server that can be accessed by the SharePoint servers in your farm.

1. On the computer that you are using for the database server, run SQL Server Setup to install SQL Server Database Engine (see Install SQL Server 2014 from the Installation Wizard (Setup)).

   When selecting features to install, choose the following:

   - Database Engine Services
   - Client Tools Connectivity
   - Management Tools - Complete (Basic will be included automatically)

2. After the Database Engine is installed, enable TCP/IP for remote connections and restart the service:

   a. Start SQL Server Configuration Manager.
   b. Open SQL Server Network Configuration.
   c. Select **Protocols for MSSQLSERVER**.
   d. Right-click **TCP/IP** and select **Enable**.
   e. Click **SQL Server Services**.
   f. Right-click **SQL Server (MSSQLSERVER)**, and click **Restart**.

3. Enable inbound access to the database server through Windows Firewall. This allows the SharePoint servers in the farm to connect to the SharePoint databases. For more information, see Configure the Windows Firewall to Allow SQL Server Access.

   a. In Windows Control Panel, in Administrative Tools, click **Windows Firewall with Advanced Security**.
   b. Click **Inbound Rules**.
   c. Click **New Rule**.
   d. In Rule Type, click **Custom**.
   e. Click **Next**.
   f. In Program, in the Services section, click **Customize**.
   g. Click **Apply to this service**.
   h. Select **SQL Server (MSSQLSERVER)** if you installed SQL Server as the default instance, and then click **OK**.
   i. Click **Next**.

j. In Protocol and Ports, accept the default settings and click **Next**.

k. In Scope, accept the default settings and click **Next**.

l. In Action, accept the default settings and click **Next**.

m. In Profile, clear the checkboxes for **Private** and **Public**, and then click **Next**.

n. In Name, type a descriptive name for the inbound rule (for example, **SQL Server**).

o. Click **Finish**.

Install and configure a three-tier SharePoint 2010 farm

On the computers you are using as SharePoint servers, run the SharePoint Prerequisite Installer program on each, followed by SharePoint Server Setup.

Use the following instructions in SharePoint 2010 documentation to install and configure a SharePoint 2010 farm that includes two web servers and an application server:

Multiple servers for a three-tier farm (SharePoint Server 2010)

When asked to specify a database server, specify the database server you installed earlier.

In the procedures that follow, it is assumed that you configured the farm using the instructions provided for three-tier farm setup. The farm should include the following services and features:

- Excel Services, Search Service, and Secure Store Service
- A web application and site collection
- Usage and health data collection
- Diagnostic logging

Install PowerPivot for SharePoint on an application server

Run SQL Server Setup to add PowerPivot for SharePoint to a SharePoint farm. If the farm consists of multiple SharePoint servers, you must run SQL Server Setup on an application server that is already joined to the farm.

Always install PowerPivot for SharePoint on an application server. Although the web front-end servers will also run PowerPivot for SharePoint server components, components that run on the web front-end are installed during the PowerPivot for SharePoint configuration step, when you deploy solutions in the farm. For more information about setup, see Install PowerPivot for SharePoint 2010.

If your deployment topology calls for two PowerPivot for SharePoint instances, run SQL Server Setup on each application server. You can only have one instance of PowerPivot for SharePoint on a single computer. If you require multiple instances, you must use additional servers. For more information about adding multiple PowerPivot for SharePoint servers to the same farm, see Deployment Checklist: Scale-out by adding PowerPivot Servers to a SharePoint 2010 farm.

Install Analysis Services client libraries on SharePoint applications servers that do not have an installation of PowerPivot for SharePoint

A farm topology that includes a web-front end or application server running the following applications, without an installation of PowerPivot for SharePoint on the same computer, will require additional software to support PowerPivot data access and features:

- Excel Services or PerformancePoint Services

- Central Administration running as a standalone application on its own server

Both Excel Services and PerformancePoint Services require a newer version of the Analysis Services OLE DB provider to support PowerPivot data access. If you run either application on a computer that does not have the most recent OLE DB provider, you must install the provider manually. For more information, see Install the Analysis Services OLE DB Provider on SharePoint Servers

Similarly, a computer that has just Central Administration, without PowerPivot for SharePoint on the same computer, will require the ADOMD.NET client library. This library is used by the PowerPivot Management Dashboard to access internal data it uses to populate the dashboard. For more information, see Install ADOMD.NET on Web Front-End Servers Running Central Administration.

Configure the server

Use the PowerPivot Configuration Tool to configure PowerPivot for SharePoint. The tool will scan the farm's existing configuration and provide options for installing or activating the SharePoint features that are needed by PowerPivot for SharePoint. During this step, the Claims to Windows Token Service will be started. Additionally, if other required SharePoint features are not yet enabled, the configuration tool will add them to the list and include actions for enabling the feature.

For more information, see Configure or Repair PowerPivot for SharePoint 2010 (PowerPivot Configuration Tool).

Configure alternate access mapping for Web front-end servers

To ensure that requests for PowerPivot data access or data refresh are handled by each web front-end server, you must map the different URLs of each server to the same web application.

1. In Central Administration, in Application Management, click **Configure alternate access mappings**.
2. In Alternate Access Mapping Collection, select your web application.
3. Click **Add Internal URL**.
4. Specify the URL of the first web front-end server.
5. Repeat the previous steps to add the URL of the second web front-end server.

Activate PowerPivot Feature Integration for Site Collections

Feature activation at the site collection level makes application pages and templates available to your sites, including configuration pages for scheduled data refresh and application pages for PowerPivot Gallery and Data Feed libraries.

The PowerPivot Configuration Tool will activate feature integration for the site collection that you specify. You can run the tool multiple times to select additional site collections. Alternatively, site administrators can configure feature activation from within SharePoint. For more information, see Activate PowerPivot Feature Integration for Site Collections in Central Administration.

Verify integration and server availability

PowerPivot query processing in the farm occurs when a user or application opens an Excel workbook that contains PowerPivot data. At a minimum, you can check pages on SharePoint sites to verify that PowerPivot features are available. However, to fully verify an installation, you must have a PowerPivot workbook that you can publish to SharePoint and access from a library. For testing purposes, you can publish a sample workbook that already contains PowerPivot data and use it to confirm that SharePoint integration is correctly configured.

To verify PowerPivot integration with a SharePoint site, do the following:

1. In a browser, open the Web application you created. If you used default values, you can specify http://<your computer name> in the URL address.
2. Verify that PowerPivot data access and processing features are available in the application. You can do this by verifying the presence of PowerPivot-provided library templates:
   a. On Site Actions, click **More Options...**
   b. In Libraries, you should see **Data Feed Library** and **PowerPivot Gallery**. These library templates are provided by the PowerPivot feature and will be visible in the Libraries list if the feature is integrated correctly.

To verify PowerPivot data access on the server, do the following:

1. Download the Picnic data sample that accompanies a Reporting Services tutorial. You will use the sample workbook in this download to verify PowerPivot data access. Extract the files.
2. Upload a PowerPivot workbook to PowerPivot Gallery or any SharePoint library.
3. Click on the document to open it from the library.
4. Click on a slicer or filter the data to start a PowerPivot query. The server will load PowerPivot data in the background and return the results. In the next step, you will connect to the server to verify the data is loaded and cached.
5. Start SQL Server Management Studio from the Microsoft SQL Server 2008 R2 program group in the Start menu. If this tool is not installed on your server, you can skip to the last step to confirm the presence of cached files.
6. In Server Type, select **Analysis Services**.
7. In Server Name, enter **<server-name>\powerpivot**, where **<server-name>** is the name of the computer that has the PowerPivot for SharePoint installation.
8. Click **Connect**.
9. In Object Explorer, click **Databases** to view the list of PowerPivot data files that are loaded.
10. On the computer file system, check the following folder to determine whether files are cached to disk. The presence of cached files is further verification that your deployment is operational. To view the file cache, go to the \Program Files\Microsoft SQL Server\MSAS10_50.POWERPIVOT\OLAP\Backup folder.

Post-Installation Steps

After you verify the installation, finish service configuration by creating a PowerPivot Gallery or tuning individual configuration settings. To make full use of the server components you just installed, you can download PowerPivot for Excel to create and then publish your first PowerPivot workbook.

Set Upper Limits on Disk Space Usage

You can set a maximum limit on how much disk space is used for PowerPivot data files that are cached to disk. The default is to use all available disk space. For instructions on how to limit disk space usage, see Configure Disk Space Usage (PowerPivot for SharePoint).

Increase File Maximum Upload Size for SharePoint Web Applications

Because PowerPivot workbooks can be large, you might want to increase the maximum file upload size. There are two file size settings to configure: Maximum Upload Size for the web application, and Maximum Workbook Size in Excel Services. The maximum file size should be set to the same value in both applications. For instructions, see Configure Maximum File Upload Size (PowerPivot for SharePoint).

Grant SharePoint Permissions to Workbook Users

Users will need SharePoint permissions before they can publish or view workbooks. Be sure to grant **View** permissions to users who need to need to view published workbooks and **Contribute** permissions to users who publish or manage workbooks. You must be a site collection administrator to grant permissions.

1. In the site, click **Site Actions**.
2. Click **Site Permissions**.
3. Select the checkbox for the site collection **Members** group.
4. On the ribbon, click **Grant Permissions**.
5. Enter the Windows domain user or group accounts who should have permission to add or remove documents.
6. Click **OK**.
7. Select the checkbox for the site collection **Visitors** group.
8. On the ribbon, click **Grant Permissions**.
9. Enter the Windows domain user or group accounts who should have permission to view documents. As before, do not use e-mail addresses or distribution group if the application is configured for classic authentication.
10. Click **OK**.

Install ADO.NET Data Services 3.5 SP1

ADO.NET Data Services is required for a data feed export of SharePoint lists. SharePoint 2010 does not include this component in the PrerequisiteInstaller program, so you must install it manually. For more information on how to install ADO.NET Data Services, see Install ADO.NET Data Services to support data feed exports of SharePoint lists.

Install Data Providers Used in Data Refresh and Check User Permissions

Server-side data refresh allows users to re-import updated data to their workbooks in unattended mode. In order for data refresh to succeed, the server must have the same data provider that was used to originally import the data. In addition, the user account under which data refresh runs often requires read permissions on the external data sources. Be sure to check the requirements for enabling and configuring data refresh to ensure a successful outcome. For more information, see PowerPivot Data Refresh with SharePoint 2010.

Create a PowerPivot Gallery

PowerPivot Gallery is a library that includes preview and presentation options for viewing PowerPivot workbooks on a SharePoint site. Using PowerPivot Gallery to publish and view PowerPivot workbooks is recommended for its preview capability. In addition, if you also deployed Reporting Services to the same SharePoint server, a PowerPivot Gallery provides ease of use in creating reports. You can launch Report Builder from within PowerPivot Gallery to base a new report on a published PowerPivot workbook. For more information about creating and using the library, see Create and Customize PowerPivot Gallery and Use PowerPivot Gallery.

Tune Configuration Settings

A PowerPivot service application is created using default properties and values. You can modify configuration settings for individual service applications to change the methodology by which requests are allocated, set server timeouts, change the thresholds for query response report events, or specify how long usage data is retained. For more information about configuration in Central Administration or about using PowerPivot features in SharePoint Web applications, see PowerPivot Server Administration and Configuration in Central Administration.

See Also

Features Supported by the Editions of SQL Server 2012
Install PowerPivot for SharePoint 2010
Deployment Checklist: Scale-out by adding PowerPivot Servers to a SharePoint 2010 farm

CheckList: Use PowerShell to Verify PowerPivot for SharePoint

2/28/2020 • 17 minutes to read • Edit Online

No PowerPivot for SharePoint installation or recovery operation is complete without a solid verification test pass that confirms your services and data are operational. In this article, we show you how to perform these steps using Windows PowerShell. We put each step into its own section so that you can go straight to specific tasks. For example, run the script in the Databases section of this topic to verify the name of the service application and content databases if you want to schedule them for maintenance or backup.

	A full PowerShell script is included at the bottom of the topic. Use the full script as a starting point to build a custom script for auditing your full PowerPivot for SharePoint deployment.

Applies to: SharePoint 2013 | SharePoint 2010

In this topic: The lettered items in the following the TOC correspond to areas of the diagram. The diagram illustrates the

Prepare your PowerShell environment

Symptoms and Recommended Actions

(A) Analysis Services Windows Service

(B) PowerPivotSystemService and PowerPivotEngineService

(C) PowerPivot Service Application(s) and proxies

(D) Databases

SharePoint Features

Timer Jobs

Health Rules

(E) Windows and ULS Logs

MSOLAP Provider

ADOMD.Net client Library

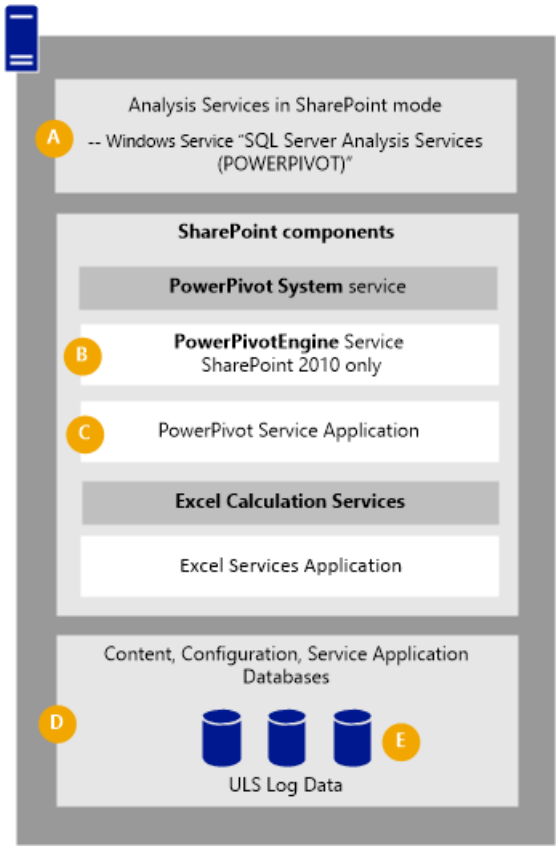
Health Data Collection Rules

Solutions

Manual Verification Steps

More Resources

Full PowerShell Script



Prepare your PowerShell environment

The steps in this section prepare your PowerShell environment. The steps may not be required, depending on how your scripting environment is currently configured.

PowerShell Permissions

Open a Powershell window or the PowerShell ISE (Integrated Scripting Environment) with **administrative privileges**. If you do not have administrative privileges when you run commands, you will see an error message similar to the following:

Get-SPLogEvent : You need to have Machine **administrator privileges** to run this cmdlet.

SharePoint and PowerPivot for SharePoint Module

If you see an error message similar to the following when you run SharePoint related cmdlets, run the Add-PSSnapin command:

The term 'Get-PowerPivotSystemService' **is not recognized as the name of a cmdlet**, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.

```
Add-PSSnapin Microsoft.Sharepoint.Powershell -EA 0
```

Windows PowerShell

For more information on the PowerShell ISE, see Introducing the Windows PowerShell ISE and Use Windows PowerShell to administer SharePoint 2013.

	You can optionally verify a majority of the components in Central Administration, using the PowerPivot management dashboard. To open the dashboard in Central Administration, click **General Application Settings**, and then click **Management Dashboard** in the **PowerPivot**. For more information on the dashboard, see PowerPivot Management Dashboard and Usage Data.

Symptoms and Recommended Actions

The following table is a list of symptoms or issues and the suggested section of this topic to consult to help you resolve the issue.

SYMPTOM	SEE SECTION
Data refresh is not running	See the section Timer Jobs and verify the **Online PowerPivot Data Refresh Timer Job** is online.
Management dashboard data is old	See the section Timer Jobs and verify the **Management Dashboard Processing Timer Job** is online.
Some portions of the Management Dashboard	If you install PowerPivot for SharePoint into a farm that has the topology of Central Administration, without Excel Services or PowerPivot for SharePoint, you must download and install the Microsoft ADOMD.NET client library if you want full access to the built-in reports in the PowerPivot management dashboard. Some reports in the dashboard use ADOMD.NET to access internal data that provides reporting data on PowerPivot query processing and server health in the farm. See the section ADOMD.Net client Library and the topic Install ADOMD.NET on Web Front-End Servers Running Central Administration.
<future content>	

Analysis Services Windows Service

The script in this section verifies the instance of Analysis Services in SharePoint mode. Verify the service is **running**.

```
Get-Service | Select name, displayname, status | Where {$_.Name -eq "msolap`$powerpivot"} | Format-Table -
Property * -AutoSize | Out-Default
```

```
Name             DisplayName                                Status
----             -----------                                ------
MSOLAP$POWERPIVOT SQL Server Analysis Services (POWERPIVOT) Running
```

PowerPivotSystemService and PowerPivotEngineService

The scripts in this section verify the PowerPivot for SharePoint system services. There is one system service for a

SharePoint 2013 deployment and two services for a SharePoint 2010 deployment.

PowerPivotSystemService

Verify the Status is **Online**.

```
Get-PowerPivotSystemService | Select typename, status, applications, farm | Format-Table -Property * -AutoSize
| Out-Default
```

```
TypeName                                  Status Applications                            Farm
--------                                  ------ ------------                            ----
SQL Server PowerPivot Service Application Online {Default PowerPivot Service Application} SPFarm
Name=SharePoint_Config_77d8ab0744a34e8aa27c806a2b8c760c
```

PowerPivotEngineService

> **NOTE**
>
> **Skip this script if** you are using SharePoint 2013. The PowerPivotEngineService is not part of a SharePoint 2013 deployment. If you run the Get-PowerPivot**Engine**Service cmdlet on SharePoint 2013, you will see an error message similar to the following. This error message is returned even if you have run the Add-PSSnapin command described in the prerequisites section of this topic.
>
> The term 'Get-PowerPivotEngineService' is not recognized as the name of a cmdlet

In a SharePoint 2010 deployment, verify the status is **Online**.

```
Get-PowerPivotEngineService | Select typename, status, name, instances, farm | Format-Table -Property * -
AutoSize | Out-Default
```

```
TypeName  : SQL Server Analysis Services
Status    : Online
Name      : MSOLAP$POWERPIVOT
Instances : {POWERPIVOT}
Farm      : SPFarm Name=SharePoint_Config
```

PowerPivot Service Application(s) and proxies

Verify the status is **Online**. The Excel Services Application does not use a service application database and therefore the cmdlet does not return a database name. Note the database used by the PowerPivot service application so you can verify the database is online in the database section later in this topic.

PowerPivot and Excel Service Application(s)

For a SharePoint 2010 deployment, verify the status is **Online**.

```
Get-PowerPivotServiceApplication | Select typename,name, status, unattendedaccount, applicationpool, farm,
database
Get-SPExcelServiceApplication | Select typename, DisplayName, status
```

```
TypeName          : PowerPivot Service Application
Name              : PowerPivotServiceApplication1
Status            : Online
UnattendedAccount : PowerPivotUnattendedAccount
ApplicationPool   : SPIisWebServiceApplicationPool Name=sqlbi_serviceapp
Farm              : SPFarm Name=SharePoint_Config
Database          : GeminiServiceDatabase Name=PowerPivotServiceApplication1_19648f3f2c944e27acdc6c20aab8487a

TypeName    : Excel Services Application Web Service Application
DisplayName : Excel Services Application
Status      : Online
```

Service Application Pool

> **NOTE**
>
> The following code sample first returns the applicationpool property of the default PowerPivot for SharePoint service application. The name is parsed from the string and used to get the status of the application pool object.
>
> Verify the Status is **Online**. If the status is not Online or you see "http error" when you browse the PowerPivot site, verify the identity credentials in the IIS application pools are still correct. The IIS pool name will is the value of the ID property returned by the Get-SPServiceApplicationPool command.

```
$poolname = [string](Get-PowerPivotServiceApplication | Select -Property applicationpool)
$position = $poolname.lastindexof("=")
$poolname = $poolname.substring($position+1)
$poolname = $poolname.substring(0,$poolname.length-1)

Get-SPServiceApplicationPool | Select name, status, processaccountname, id | Where {$_.Name -eq $poolname} |
Format-Table -Property * -AutoSize | Out-Default
```

```
Name                          Status ProcessAccountName Id
----                          ------ ------------------ -------
SharePoint Web Services System Online DOMAIN\account     89b50ec3-49e3-4de7-881a-2cec4b8b73ea
```

Tip
The application pool can also be verified on the Central Administration page **Manage Service Applications**. Click the name of the service application and then click **properties** in the ribbon.

PowerPivot and Excel Service Application proxies

Verify the Status is **Online**.

```
Get-SPServiceApplicationProxy | Select typename, status, unattendedaccount, displayname | Where {$_.TypeName -
Like "*powerpivot*" -Or $_.TypeName -Like "*excel services*"} | Format-Table -Property * -AutoSize | Out-
Default
```

```
TypeName                                          Status UnattendedAccount           DisplayName
--------                                          ------ -----------------           -----------
PowerPivot Service Application Proxy              Online PowerPivotUnattendedAccount
PowerPivotServiceApplication1
Excel Services Application Web Service Application Proxy Online                       Excel Services
Application
```

Databases

The following script returns the status of the service application databases and all content databases. Verify the status is **Online**.

```
Get-SPDatabase | Select name, status, server, typename | Where {$_.TypeName -eq "content database" -Or
$_.TypeName -Like "*Gemini*"} | Format-Table -Property * -AutoSize | Out-Default
```

```
Name                                                                    Status Server
TypeName
----                                                                    ------ ------                  -----
---
DefaultPowerPivotServiceApplicationDB-38422181-2b68-4ab2-b2bb-9c00c39e5a5e Online SPServer Name=TESTSERVER
Microsoft.AnalysisServices.SPAddin.GeminiServiceDatabase
DefaultWebApplicationDB-f0db1a8e-4c22-408c-b9b9-153bd74b0312            Online TESTSERVER\POWERPIVOT
Content Database
SharePoint_Admin_3cadf0b098bf49e0bb15abd487f5c684                       Online TESTSERVER\POWERPIVOT
Content Database
```

SharePoint Features

Verify the site, web, and farm features are online.

```
Get-SPFeature | Select displayname, status, scope, farm | Where {$_.displayName -Like "*powerpivot*"} | Format-
Table -Property * -AutoSize | Out-Default
```

```
DisplayName     Status Scope Farm
-----------     ------ ----- ----
PowerPivotSite  Online  Site SPFarm Name=SharePoint_Config
PowerPivotAdmin Online   Web SPFarm Name=SharePoint_Config
PowerPivot      Online  Farm SPFarm Name=SharePoint_Config
```

Timer Jobs

Verify the Time Jobs are **Online**. The PowerPivot EngineService is not installed on SharePoint 2013, therefore the script will not list EngineService timer jobs in a SharePoint 2013 deployment.

```
Get-SPTimerJob | Where {$_.service -Like "*power*" -Or $_.service -Like "*mid*"} | Select status, displayname,
LastRunTime, service | Format-Table -Property * -AutoSize | Out-Default
```

```
      Status DisplayName                                                                  LastRunTime
Service
------ -----------                                                                  -----------
-------
Online Health Analysis Job (Daily, SQL Server Analysis Services, All Servers)              4/9/2014 12:00:01
AM EngineService Name=MSOLAP$POWERPIVOT
Online Health Analysis Job (Hourly, SQL Server Analysis Services, All Servers)             4/9/2014 1:00:01 PM
EngineService Name=MSOLAP$POWERPIVOT
Online Health Analysis Job (Weekly, SQL Server Analysis Services, All Servers)             4/6/2014 12:00:10
AM EngineService Name=MSOLAP$POWERPIVOT
Online PowerPivot Management Dashboard Processing Timer Job                                4/8/2014 3:45:38 AM
MidTierService
Online PowerPivot Health Statistics Collector Timer Job                                    4/9/2014 1:00:12 PM
MidTierService
Online PowerPivot Data Refresh Timer Job                                                   4/9/2014 1:09:36 PM
MidTierService
Online Health Analysis Job (Daily, SQL Server PowerPivot Service Application, All Servers)  4/9/2014 12:00:00
AM MidTierService
Online Health Analysis Job (Daily, SQL Server PowerPivot Service Application, Any Server)   4/9/2014 12:00:00
AM MidTierService
Online Health Analysis Job (Weekly, SQL Server PowerPivot Service Application, All Servers) 4/6/2014 12:00:03
AM MidTierService
Online Health Analysis Job (Weekly, SQL Server PowerPivot Service Application, Any Server)  4/6/2014 12:00:03
AM MidTierService
Online PowerPivot Setup Extension Timer Job                                                4/1/2014 1:40:31 AM
MidTierService
```

Health Rules

There are fewer rules in a SharePoint 2013 deployment. For a full list of rules for each SharePoint environment and an explanation of how to use the rules, see PowerPivot Health Rules - Configure.

```
Get-SPHealthAnalysisRule | Select name, enabled, summary | Where {$_.summary -Like "*power*"}  | Format-Table -
Property * -AutoSize | Out-Default
```

```
Name                          Enabled Summary
----                          ------- -------
SecondaryLogonHealthRule         True PowerPivot:  Secondary Logon service (seclogon) is disabled
DataRefreshTimerJobHealthRule    True PowerPivot: The PowerPivot Data Refresh timer job is disabled.
ASUsageLoadHealthRule            True PowerPivot: The ratio of load events to connections is too high.
ASMiniDumpHealthRule             True PowerPivot: One or more minidump files were found in the Logs directory,
indicating a program crash
ASUsageCubeRule                  True PowerPivot: Usage data is not getting updated at the expected frequency.
ASADOMDNETHealthRule             True PowerPivot: ADOMD.NET is not installed on a standalone WFE that is
configured for central admin
MidTierAcctReadPermissionRule    True PowerPivot: MidTier process account should have 'Full Read' permission on
all associated SPWebApplications.
```

Windows and ULS Logs

Windows event log

The following command will search the windows event log for events related to the instance of Analysis Services in SharePoint mode. For information on disabling events or changing the event level, see Configure and View SharePoint Log Files and Diagnostic Logging (PowerPivot for SharePoint).

Service Name: MSOLAP$POWERPIVOT

Display name in Windows Services: SQL Server Analysis Services (POWERPIVOT)

```
Get-EventLog "application" | Where-Object {$_.source -Like "msolap`$powerpivot*"}  | Select timegenerated,
entrytype , source, message | Format-Table -property * -AutoSize | Out-Default
```

```
TimeGenerated           EntryType Source            Message
-------------           --------- ------            -------
4/16/2014 1:45:19 PM  Information MSOLAP$POWERPIVOT Software usage metrics are disabled.
4/16/2014 1:45:19 PM  Information MSOLAP$POWERPIVOT Service started. Microsoft SQL Server Analysis Services 64
Bit Evaluation (x64) RTM 12.0.1997.5.
4/16/2014 1:45:18 PM  Information MSOLAP$POWERPIVOT The flight recorder was started.
4/14/2014 6:45:37 PM  Information MSOLAP$POWERPIVOT Software usage metrics are disabled.
```

SharePoint ULS Log, last 48 hours

The following command will return PowerPivot messages from the ULS log that were created in the last 48 hours. Adjust the addhours parameter for your need.

```
Get-SPLogEvent -StartTime(Get-Date).AddHours(-48) | Where-Object {$_.Area -eq "powerpivot service" -and
$_.level -eq "high"} | Select timestamp, area, category, eventid,level, message | Format-Table -Property * -
AutoSize | Out-Default
```

The following variation of the command only returns log events for the **data refresh** category.

```
Get-SPLogEvent -starttime(Get-Date).addhours(-48) | Where-Object {$_.category -eq "data refresh" -and $_.level
-eq "high"} | Select timestamp, area, category, eventid, level, correlation, message
```

```
Timestamp   : 4/14/2014 7:15:01 PM
Area        : PowerPivot Service
Category    : Data Refresh
EventID     : 43
Level       : High
Correlation : 5755879c-7cab-e097-8f80-f27895d44a77
Message     : The following error occurred when working with the service application, Default PowerPivot
Service Application. Skipping the service application..

Timestamp   : 4/14/2014 7:15:02 PM
Area        : PowerPivot Service
Category    : Data Refresh
EventID     : 99
Level       : High
Correlation : 5755879c-7cab-e097-8f80-f27895d44a77
Message     : EXCEPTION: System.TimeoutException: The request channel timed out while waiting for a reply after
00:00:47.0625313. Increase the timeout value passed to
              the call to Request or increase the SendTimeout value on the Binding. The time allotted to this
operation may have been a portion of a longer timeout.
              ---> System.TimeoutException: The HTTP request to
'http://localhost:32843/SecurityTokenServiceApplication/securitytoken.svc/actas' has exceeded the
              allotted timeout of 00:00:54.5930000. The time allotted to this operation may have been a portion
of a longer timeout. ---> System.Net.WebException: The
              operation has timed out     at System.Net.HttpWebRequest.GetResponse()     at

System.ServiceModel.Channels.HttpChannelFactory`1.HttpRequestChannel.HttpChannelRequest.WaitForReply(TimeSpan
timeout...
```

MSOLAP Provider

Verify the provider MSOLAP provider. SQL Server 2012 and SQL Server 2014 PowerPivot require MSOLAP.5.

```
$excelApp = Get-SPExcelServiceApplication
Get-SPExcelDataProvider -ExcelServiceApplication $excelApp | Select providerid, providertype, description |
Where {$_.providerid -like "msolap*" } | Format-Table -Property * -AutoSize | Out-Default
```

```
ProviderId ProviderType Description
---------- ------------ -----------
MSOLAP     Oledb        Microsoft OLE DB Provider for OLAP Services
MSOLAP.3   Oledb        Microsoft OLE DB Provider for OLAP Services 9.0
MSOLAP.4   Oledb        Microsoft OLE DB Provider for OLAP Services 10.0
MSOLAP.5   Oledb        Microsoft OLE DB Provider for OLAP Services 11.0
```

For more information, see Install the Analysis Services OLE DB Provider on SharePoint Servers and Add MSOLAP.5 as a Trusted Data Provider in Excel Services.

ADOMD.Net client Library

```
Get-WMIObject -Class win32_product | Where-Object {$_.name -Like "*ado*"} | Select name, version, vendor |
Format-Table -Property * -AutoSize | out-default
```

```
name                                                version      vendor
----                                                -------      ------
Microsoft SQL Server 2008 Analysis Services ADOMD.NET 10.1.2531.0  Microsoft Corporation
Microsoft SQL Server 2005 Analysis Services ADOMD.NET 9.00.1399.06 Microsoft Corporation
```

For more information, see Install ADOMD.NET on Web Front-End Servers Running Central Administration.

Health Data Collection Rules

Verify the **Status** is Online and **Enabled** is True.

```
Get-SPUsageDefinition | Select name, status, enabled, tablename, DaysToKeepDetailedData | Where {$_.name -Like
"powerpivot*"} | Format-Table -Property * -AutoSize | Out-Default
```

```
Name                        Status Enabled TableName                  DaysToKeepDetailedData
----                        ------ ------- ---------                  ----------------------
PowerPivot Connections      OnlineTrue AnalysisServicesConnections  14
PowerPivot Load Data Usage  Online    True AnalysisServicesLoads                           14
PowerPivot Query Usage      Online    True AnalysisServicesRequests                        14
PowerPivot Unload Data Usage Online   True AnalysisServicesUnloads                         14
```

For more information, see PowerPivot Usage Data Collection.

Solutions

If the other components are online then you can skip verifying the solutions. If however the Health rules are missing, verify the two solutions exist and showed Verify the two PowerPivot solutions are **Online** and **Deployed**.

```
Get-SPSolution | Select name, status, deployed, DeploymentState, DeployedServers | Where {$_.Name -Like
"*powerpivot*"} | Format-Table -Property * -AutoSize | Out-Default
```

SharePoint 2013:

```
Name                                Status Deployed        DeploymentState DeployedServers
----                                ------ --------        --------------- ---------------
powerpivotfarm14solution.wsp        Online     True         GlobalDeployed {UETESTA00}
powerpivotfarmsolution.wsp          Online     True         GlobalDeployed {UETESTA00}
powerpivotwebapplicationsolution.wsp Online     True WebApplicationDeployed {UETESTA00}
```

SharePoint 2010:

```
Name                Status Deployed        DeploymentState DeployedServers
----                ------ --------        --------------- ---------------
powerpivotfarm.wsp  Online     True         GlobalDeployed {uesql11spoint2}
powerpivotwebapp.wsp Online     True WebApplicationDeployed {uesql11spoint2}
```

For more information on how to deploy SharePoint solutions, see Deploy solution packages (SharePoint Server 2010).

Manual Verification Steps

This section describes verification steps that cannot be completed with PowerShell cmdlets.

Scheduled Data Refresh: Configure the refresh schedule a workbook to **Also refresh as soon as possible**. For more information, see the "Verify Data Refresh" section of Schedule Data Refresh and Data Sources That Do Not Support Windows Authentication (PowerPivot for SharePoint).

More Resources

Web Server (IIS) Administration Cmdlets in Windows PowerShell.

PowerShell to check services, IIS sites and Application Pool status in SharePoint.

Windows PowerShell for SharePoint 2013 reference

Windows PowerShell for SharePoint Foundation 2010 reference

Manage Excel Services with Windows PowerShell (SharePoint Server 2010)

View and Read SQL Server Setup Log Files

Use the Get-EvenLog cmdlet

Full PowerShell Script

The Following script contains all of the commands from the previous sections. The script runs the commands in the same order as they are presented in this topic. The script contains some optional variations of the commands noted in this topic in case you need additional filtering. The variations are disabled with a comment character (#). The script also includes some statements for verifying Reporting Services SharePoint mode. The Reporting Services statements are disabled with a comment character (#).

```
# This script audits services related to PowerPivot for SharePoint
$starttime = Get-Date
write-host -foregroundcolor DarkGray StartTime $starttime

Write-Host  "Import the SharePoint PowerShell snappin"
Add-PSSnapin Microsoft.Sharepoint.Powershell -EA 0

Write-Host -ForegroundColor Green "Analysis Services Windows Service"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-Service | Select name, displayname, status | Where {$_.Name -eq "msolap`$powerpivot"} | Format-Table -
Property * -AutoSize | Out-Default
```

```
Property * -AutoSize | Out-Default

Write-Host -ForegroundColor Green "PowerPivotEngineService and PowerPivotSystemService"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

Get-PowerPivotSystemService | Select typename, status, applications, farm | Format-Table -property * -AutoSize
| Out-Default
# If needed, you can run the following to compare job definitions specific to the service against the results
of the timer job definition section
#Get-PowerPivotSystemService | select -ExpandProperty jobdefinitions | select displayname, schedule, service |
format-table -property * -autosize | out-default

Get-PowerPivotEngineService | Select typename, status, name, instances, farm | Format-Table -property * -
AutoSize | Out-Default
# If needed, you can run the following to compare job definitions specific to the service against the results
of the timer job definition section
#Get-PowerPivotEngineService | select -ExpandProperty jobdefinitions | select displayname, schedule, service |
format-table -property * -autosize | out-default

#Write-Host -ForegroundColor Green "Service Instances - optional if you want to associate services with the
server"
#Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
#Get-SPServiceInstance | select typename, status, server, service, instance | where {$_.TypeName -like
"*powerpivot*" -or $_.TypeName -like "*excel*" -or $_.TypeName -like "*Analysis Services*"} | format-table -
property * -autosize | out-default
#Get-PowerPivotEngineServiceInstance  | select typename, ASServername, status, server, service, instance
#Get-PowerPivotSystemServiceInstance  | select typename, ASSServerName, status, server, service, instance

Write-Host -ForegroundColor Green "PowerPivot And Excel Service Applications"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-PowerPivotServiceApplication | Select typename,name, status, unattendedaccount, applicationpool, farm,
database
Get-SPExcelServiceApplication | Select typename,  DisplayName, status

#Write-Host ""
Write-Host -ForegroundColor Green "PowerPivot Service Application pool"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
# the following assumes there is only 1 PowerPivot Service Application, and returns that application's pool
name.  if you have more than one, use the 2nd version
$poolname = [string](Get-PowerPivotServiceApplication | Select -Property applicationpool)
$position = $poolname.lastindexof("=")
$poolname = $poolname.substring($position+1)
$poolname = $poolname.substring(0,$poolname.length-1)
Get-SPServiceApplicationPool | Select name, status, processaccountname, id | Where {$_.Name -eq $poolname} |
Format-Table -Property * -AutoSize | Out-Default

#Write-Host ""
Write-Host -ForegroundColor Green "PowerPivot and Excel Service Application Proxy"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-SPServiceApplicationProxy |  Select typename, status, unattendedaccount, displayname | Where {$_.TypeName -
like "*powerpivot*" -or $_.TypeName -like "*excel services*"} | Format-Table -Property * -AutoSize | Out-
Default
#Get-SPServiceApplicationProxy |  select typename, status, unattendedaccount, displayname | where {$_.TypeName
-like "*powerpivot*" -or $_.TypeName -like "*Reporting Services*" -or $_.TypeName -like "*excel services*"} |
format-table -property * -autosize | out-default

Write-Host -ForegroundColor Green "DATABASES"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-SPDatabase | Select name, status, server, typename | Where {$_.TypeName -eq "content database" -or
$_.TypeName -like "*Gemini*"} | Format-Table -Property * -AutoSize | Out-Default
#Get-SPDatabase | select name, status, server, typename | where {$_.TypeName -eq "content database" -or
$_.TypeName -like "*Gemini*" -or $_.TypeName -like "*ReportingServices*"}

Write-Host -ForegroundColor Green "features"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-SPFeature | Select displayname, status, scope, farm | Where {$_.displayName -like "*powerpivot*"} | Format-
Table -Property * -AutoSize | Out-Default
#Get-SPFeature | select displayname, status, scope, farm | where {$_.displayName -like "*powerpivot*" -or
$_.displayName -like "*ReportServer*"}  | format-table -property * -autosize | out-default
```

```
Write-Host -ForegroundColor Green "Timer Jobs (Job Definitions) -- list is the same as seen in the 'Review
timer job definitions' section of the management dashboard"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-SPTimerJob | Where {$_.service -like "*power*" -or $_.service -like "*mid*"} | Select status, displayname,
LastRunTime, service | Format-Table -Property * -AutoSize | Out-Default

Write-Host -ForegroundColor Green "health rules"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-SPHealthAnalysisRule | Select name, enabled, summary | Where {$_.summary -Like "*power*"}  | Format-Table -
Property * -AutoSize | Out-Default

$time = Get-Date
Write-Host -ForegroundColor DarkGray StartTime $starttime
Write-Host -ForegroundColor DarkGray EndTime $time

Write-Host -ForegroundColor Green "Windows Event Log data MSSQL$POWERPIVOT and "
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-EventLog "application" | Where-Object {$_.source -like "msolap`$powerpivot*"}  | Select timegenerated,
entrytype , source, message | Format-Table -Property * -autosize | Out-Default
#The following is the same command but with the Inforamtion events filtered out.
#Get-EventLog "application" | Where-Object {$_.source -like "msolap`$powerpivot*" -and ($_.entrytype -match
"error" -or $_.entrytype -match "critical" -or $_.entrytype -match "warning")}  |select timegenerated,
entrytype , source, message | format-table -property * -autosize | out-default

#Write-Host ""
Write-Host -ForegroundColor Green "ULS Log data"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-SPLogEvent -StartTime(Get-Date).AddHours(-48) | Where-Object {$_.Area -eq "powerpivot service" -and
$_.level -eq "high"} | Select timestamp, area, category, eventid, level, correlation, message | Format-Table -
Property * -AutoSize | Out-Default
#the following example filters for the category 'data refresh'
#Get-SPLogEvent -starttime(get-date).addhours(-48) | Where-Object {$_.category -eq "data refresh" -and $_.level
-eq "high"} | select timestamp, area, category, eventid, level, correlation, message

$time = Get-Date
Write-Host -ForegroundColor DarkGray StartTime $starttime
Write-Host -ForegroundColor DarkGray EndTime $time

Write-Host -ForegroundColor Green "MSOLAP data provider for Excel Servivces, service application"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
$excelApp = Get-SPExcelServiceApplication
Get-SPExcelDataProvider -ExcelServiceApplication $excelApp | Select providerid, providertype, description |
Where {$_.providerid -like "msolap*" } | Format-Table -Property * -AutoSize | Out-Default

Write-Host -ForegroundColor Green "ADOMD.net client library"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-WMIObject -Class win32_product | Where-Object {$_.name -like "*ado*"} | Select name, version, vendor |
Format-Table -Property * -AutoSize | Out-Default

Write-Host -ForegroundColor Green "Usage Data Rules"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-SPUsageDefinition | Select name, status, enabled, tablename, DaysToKeepDetailedData | Where {$_.name -like
"powerpivot*"} | Format-Table -Property * -AutoSize | Out-Default

Write-Host -ForegroundColor Green "Solutions"
Write-Host -ForegroundColor Green ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"
Get-SPSolution | Select name, status, deployed, DeploymentState, DeployedServers | Where {$_.Name -like
"*powerpivot*"} | Format-Table -Property * -AutoSize | Out-Default

$time = Get-Date
Write-Host -ForegroundColor DarkGray StartTime $starttime
Write-Host -ForegroundColor DarkGray EndTime $time
```

Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013)

5/22/2019 • 2 minutes to read • Edit Online

Reporting Services in SharePoint mode is a collection of server components that provide report generation and delivery, based on Microsoft SQL Server and Microsoft SharePoint products.

Running Reporting Services in SharePoint mode provides the Power View and data alerting features. For more information regarding features in SharePoint mode, see the section "Feature Support and Behavior Differences by Server Mode" in Reporting Services Report Server

There are two fundamental installations needed for Reporting Services in SharePoint mode:

INSTALLATION	DESCRIPTION
The Microsoft SQL Server Reporting Services add-in for SharePoint products.	The add-in installs the Reporting Services user interface (UI) pages and features on a SharePoint web front-end server. The UI features include Power View, administration pages in SharePoint Central Administration, feature pages used within SharePoint document libraries, and Reporting Services Data Alerting pages.
The Microsoft SQL Server Reporting Services report server installed in SharePoint Mode	The report server handles the data and report processing and rendering as well subscription and Data Alert processing. The SharePoint mode report server is architected and installed as a SharePoint Shared Service.

To install Reporting Services, use the SQL Server 2014 installation media.

For instructions on advanced deployment scenarios, see Deployment Checklist: Reporting Services, Power View, and PowerPivot for SharePoint and Deployment Checklist: Install Reporting Services into an Existing SharePoint Farm.

In This Section

Supported Combinations of SharePoint and Reporting Services Server and Add-in (SQL Server 2014)

Install Reporting Services SharePoint Mode for SharePoint 2013

Install Reporting Services SharePoint Mode for SharePoint 2010

Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013)

Where to find the Reporting Services add-in for SharePoint Products

Add an Additional Report Server to a Farm (SSRS Scale-out)

Add an Additional Reporting Services Web Front-end to a Farm

Configure E-mail for a Reporting Services Service Application (SharePoint 2010 and SharePoint 2013)

Provision Subscriptions and Alerts for SSRS Service Applications

Claims to Windows Token Service (C2WTS) and Reporting Services

See Also

Data Alerts Architecture and Workflow
Data Alert Manager for Alerting Administrators

Hardware and Software Requirements for Reporting Services in SharePoint Mode

12/16/2019 • 2 minutes to read • Edit Online

This topic describes prerequisites, hardware requirements, and installation considerations for SQL Server 2014 Reporting Services running in SharePoint mode. Because Reporting Services SharePoint mode requires a SharePoint server, most of the requirements are based on the SharePoint environment. For native mode report servers, your hardware should meet minimum hardware and software requirements for running SQL Server 2014. For more information, see Hardware and Software Requirements for Installing SQL Server 2014.

- Prerequisites
- Report Server Database Requirements
- Power View Requirements
- More Information

Prerequisites

- For local installations, the account logged in during installation of SharePoint and Reporting Services needs to be a member of the administrators group in the local operating system. The setup account does not need to be a member of the SharePoint farm administrators group.

  For more information, see Account permissions and security settings in SharePoint 2013.

- Reporting Services running in SharePoint mode requires SharePoint Server. For more information about SharePoint requirements and configurations, see the following:

  - Hardware and software requirements (SharePoint 2013) (https://go.microsoft.com/fwlink/p/?LinkId=256365)
  - Capacity management and sizing for SharePoint Server 2013
  - Software requirements for business intelligence (SharePoint 2013)
  - Hardware and software requirements (SharePoint Server 2010)
  - Capacity management and sizing for SharePoint Server 2010

- If you want to upgrade or update existing Reporting Services SharePoint installation with SQL Server 2014, see Upgrade and Migrate Reporting Services.

- Verify the **SharePoint 2013 Administration** service is started in Windows Server Manager.

Report Server Database Requirements

- Both Reporting Services and SharePoint products and technologies use SQL Server relational databases to store application data.

- SQL Server 2014 Reporting Services (SSRS) requires an instance of the SQL Server Database Engine of a compatible SQL Server edition. For more information on hardware and software requirements, see Hardware and Software Requirements for Installing SQL Server 2014.

- SharePoint products can use an existing database instance. If an instance of Database Engine is not installed, the SharePoint Products Setup program installs SQL Server Express Edition for the SharePoint application

databases.

- The report server instance cannot use the SQL Server Express Edition for its database. However, the SQL Server Express Edition instance installed by the SharePoint product can exist side-by-side with other Database Engine editions.

Power View Requirements

Review the most up-to-date Power View documentation on Office.Microsoft.com. Power View is a feature of Microsoft Excel 2013, and is part of the SQL Server 2014 Reporting Services add-in for Microsoft SharePoint Server 2010 and 2013 Enterprise Editions.

More Information

For information on SharePoint changes, see Changes from SharePoint 2010 to SharePoint 2013 (https://technet.microsoft.com/library/ff607742(office.15).aspx).

SQL Server 2014 Release Notes.

Supported Combinations of SharePoint and Reporting Services Server and Add-in (SQL Server 2014)

5/22/2019 • 2 minutes to read • Edit Online

Reporting Services report servers can be installed in SharePoint mode and integrated with a SharePoint deployment. Not all features are supported in all combinations of report server, Reporting Services add-in for SharePoint, and SharePoint Products. This topic summarizes the supported combinations. In SQL Server 2014 the integration is a result of the combination of the following:

- A version of a Reporting Services report server configured for SharePoint mode.
- A SharePoint product.
- The Reporting Services add-in for SharePoint products, which you install on the SharePoint servers.

Applies to: SharePoint 2013 | SharePoint 2010 | SharePoint 2007

Supported Combinations of SharePoint and Reporting Services Components

The following table summarizes the supported combinations of report server, the Reporting Services add-in for SharePoint products, and SharePoint products. Combinations that are not list in the following table are not supported

Supported Combinations

	REPORT SERVER	ADD-IN	SHAREPOINT VERSION	SUPPORTED
1	SQL Server 2014	SQL Server 2014	SharePoint 2013	Yes
2	SQL Server 2014	SQL Server 2014	SharePoint 2010	Yes
3	SQL Server 2012 SP1	SQL Server 2014 and SQL Server 2012 SP1	SharePoint 2013	Yes
4	SQL Server 2012 and SQL Server 2012 SP1	SQL Server 2014	SharePoint 2010	Yes Exception: Power view integration is not supported.
5	SQL Server 2012	SQL Server 2012	SharePoint 2010	Yes
6	SQL Server 2008 R2	SQL Server 2014	SharePoint 2010	Yes

	REPORT SERVER	ADD-IN	SHAREPOINT VERSION	SUPPORTED
7	SQL Server 2008 R2	SQL Server 2012 and SQL Server 2012 SP1	SharePoint 2010	Yes
8	SQL Server 2008 R2	SQL Server 2008 R2	SharePoint 2010	Yes
9	SQL Server 2008 R2	SQL Server 2008 SP2	SharePoint 2007	Yes
10	SQL Server 2008 SP2	SQL Server 2008 R2	SharePoint 2010	Yes
11	SQL Server 2008 SP2	SQL Server 2008 and SQL Server 2008 SP2	SharePoint 2007	Yes

For more information on Reporting Services features and report server modes, see Reporting Services Report Server.

Additional Notes:

- SharePoint 2013 support, including Power view integration, requires the Reporting Services report server and the Reporting Services add-in version of SQL Server 2012 SP1 or later.
- Power View was introduced in SQL Server 2012. Therefore, Power View integration with SharePoint 2010 requires the SQL Server 2012 or later of the add-in.
- The SQL Server 2008 R2 Add-In is not supported by SQL Server 2012 (or later) report servers. The SharePoint 2010 prerequisite installer automatically installs the SQL Server 2008 R2 Add-In. It must be uninstalled before installing newer versions of the add-in. In place upgrade of the add-in is not supported.
- **Upgrade:** SharePoint 2010 with the Reporting Services Add-In installed, cannot be upgraded in-place to SharePoint 2013. SharePoint 2013 requires SQL Server 2012 SP1 or later of the Reporting Services add-in and report server. For more information on upgrade, see Upgrade and Migrate Reporting Services.

See Also

Where to find the Reporting Services add-in for SharePoint Products
Features Supported by the Editions of SQL Server 2014
Upgrade and Migrate Reporting Services

Install Reporting Services SharePoint Mode for SharePoint 2013

10/23/2019 • 17 minutes to read • Edit Online

The procedures in this topic guide you through a single server installation of Reporting Services in SharePoint mode. The steps include running the SQL Server installation wizard as well as configuration tasks that use SharePoint Central Administration. The topic can also be used for individual procedures for updating an existing installation, for example to create a Reporting Services service application.

Applies to: SharePoint 2013 | **Note:** Reporting Services SharePoint mode does **not** support SharePoint Server multi-tenancy.

For information on adding more Reporting Services servers to an existing farm, see the following:

- Add an Additional Report Server to a Farm (SSRS Scale-out)
- Add an Additional Reporting Services Web Front-end to a Farm

A single server installation is useful for development and testing scenarios but it is not recommended for production environments.

In this topic:

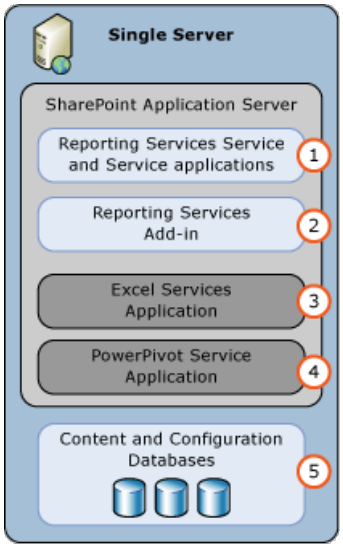
- Example Single-Server Deployment
- Setup accounts
- Step 1: Install Reporting Services Report Server in SharePoint mode
- Step 2: Register and Start the Reporting Services SharePoint Service
- Step 3: Create a Reporting Services Service Application
- Step 4: Activate the Power View Site Collection Feature.
- Windows PowerShell script for Steps 1-4
- Additional Configuration including provisioning for subscriptions and alerts, Reporting Services content types, and configuring Excel Services to Use an Analysis Services Server.
- Verify the installation

Example Single-Server Deployment

A single-server installation is useful for development and testing scenarios but a single-server is not recommended for a production environment. The single-server environment refers to a single computer that has SharePoint and Reporting Services components installed on the same computer. The topic does not cover scale-out with multiple Reporting Services servers.

The following diagram illustrates the components that are part of a single server Reporting Services deployment.

(1)	SharePoint service installed from SQL Server installation. You can create one or more Reporting Services service applications.
(2)	Reporting Services add-in for SharePoint products provides the user interface components on the SharePoint Servers.
(3)	The Excel Service Application used by Power View and PowerPivot.
(4)	PowerPivot service application.



TIP

For more complex deployment examples, see Deployment Topologies for SQL Server BI Features in SharePoint.

Setup accounts

This section describes the accounts and permissions used for the primary deployment steps of Reporting Services in SharePoint mode.

Installation and registering the Reporting Services Service:

- The current account during the installation (referred to as the 'setup' account) of Reporting Services in SharePoint mode needs to have administrative rights in the local computer. If you are installing Reporting Services after SharePoint is installed and the 'setup' account is also a member of the SharePoint farm administrators group, the Reporting Services installation will register the Reporting Services service for you. If you install Reporting Services before SharePoint is installed or the 'setup' account is not a member of the farm administrators group, you register the service manually. See the section Step 2: Register and Start the Reporting Services SharePoint Service.

Creating Reporting Services Service Applications

- Following installation and registering the Reporting Services service, create one or more Reporting Services service applications. The "SharePoint farm service account " needs to temporarily be a member of the local administrators group so the Reporting Services service application can be created. For more information on SharePoint 2013 account permissions, see Account permissions and security settings in SharePoint 2013 (https://technet.microsoft.com/library/cc678863.aspx).

It is security best practice that SharePoint farm administrator accounts are not also local operating system administrator accounts. If you add a farm admin account to the local administrators group as part of your installation process, it is recommended you remove the account from the local administrators group after installation is complete.

Step 1: Install Reporting Services Report Server in SharePoint mode

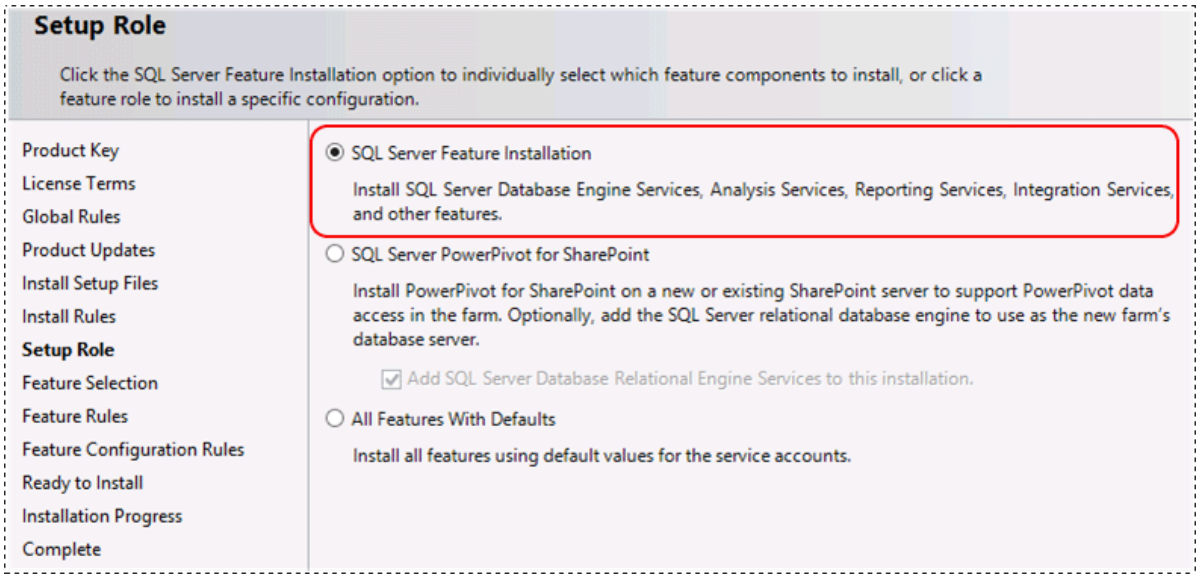
This step installs a Reporting Services report server in SharePoint mode and the Reporting Services add-in for SharePoint products. Depending on what is already installed on your computer, you may not see some of the installation pages described in the following steps.

1. Run the SQL Server Installation Wizard (Setup.exe).
2. Click **Installation** in the left side of the wizard and then click **New SQL Server stand-alone installation or add features to an existing installation**.
3. Click **OK** on the **Setup Support Rules** page, assuming all rules passed.
4. Click **Install** on the **Setup Support Files** page. Depending on what is already installed on your computer, you might see the following message:
   - "One or more affected files have operations pending. You must restart your computer after the setup process is completed."
   - Click **Ok**.
5. Click **Next** after the support files have completed installing and the **Support Rules** pages show a status of **Passed**. Review any warnings or blocking issues.
6. On the **Installation Type** page, click **Add features to an existing instance of SQL Server 2014**. Select the correct instance in the drop-down list and click **Next**.
7. If you see the **Product Key** page, type your key or accept the default of the 'Enterprise Evaluation' edition.

   Click **Next**.
8. If you see the License terms page, review and accept the license terms. Microsoft appreciates you clicking to agree to send feature usage data to help improve product features and support.

   Click **Next**.
9. If you see the **Setup Role** page, select **SQL Server Feature Installation**

   Click **Next**



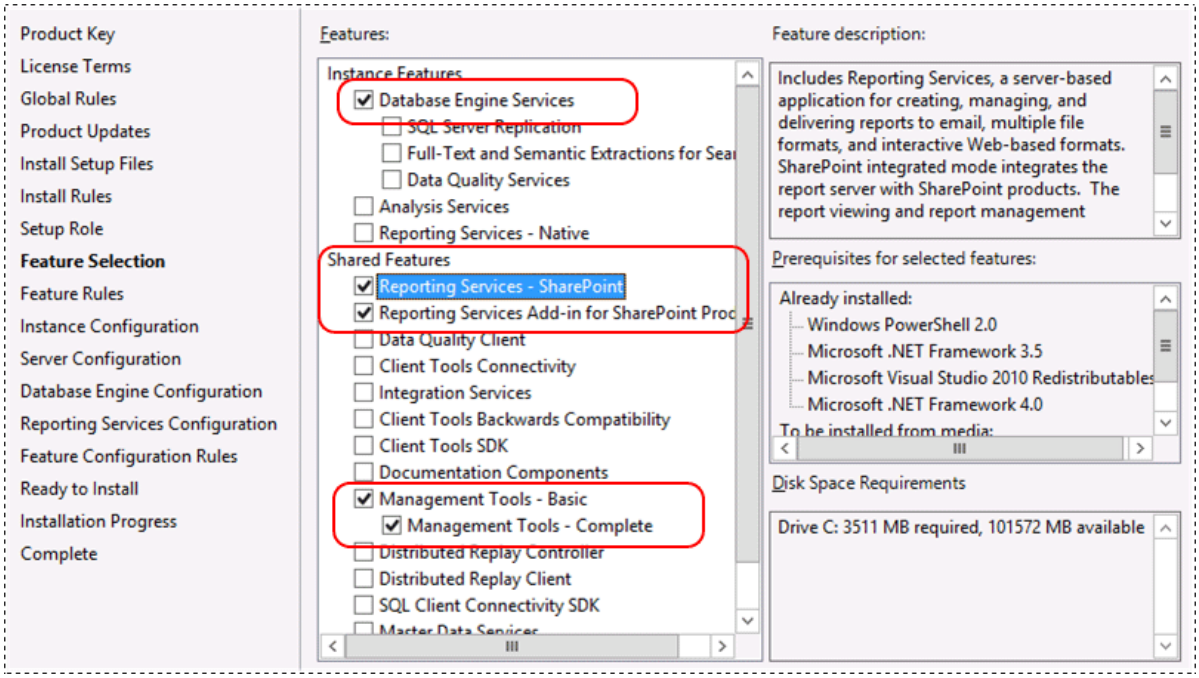
10. Select the following on the **Feature Selection** page:

    - **Reporting Services - SharePoint**
    - **Reporting Services add-in for SharePoint Products.**

      Tip
      The installation wizard option for installing the add-in is new with the SQL Server 2014 release.

    - If you do not already have an instance of SQL Server Database Engine, you could also select **Database Engine Services** and **Management Tools Complete** for a complete environment.

    Click **Next**.



11. If you see the **Installation Rules** page. Review any warnings or blocking issues. Then click **Next**

12. If you selected the Database Engine services, accept the default instance of **MSSQLSERVER** on the **Instance Configuration** page and click **Next**.

    Tip
    The Reporting Services SharePoint service architecture is not based on a SQL Server "instance" as was the previous Reporting Services architecture.

13. Review the **Disk Space Requirements** page and click **Next**.

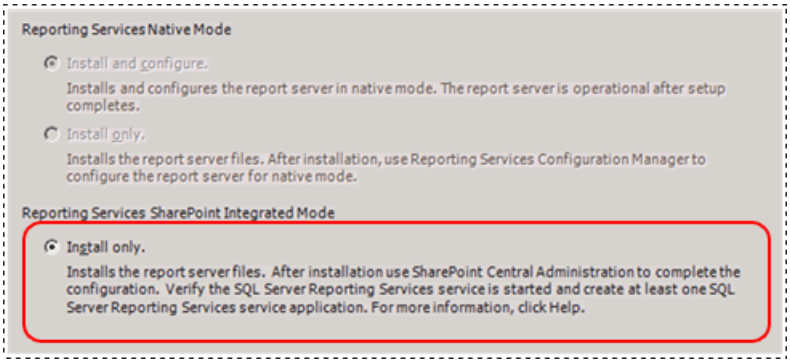
14. If you see the **Server Configuration** page type appropriate credentials. If you want to use the Reporting Services data alerting or subscription features, you need to change the **Startup Type** for SQL Server Agent to **Automatic**. You may not see the **Server Configuration** page, depending on what is already installed on the computer.

    Click **Next**.

15. If you selected the Database Engine services, you will see the **Database Engine Configuration** page, add appropriate accounts to the list of SQL Administrators and click **Next**.

16. On the **Reporting Services Configuration** page you should see the **Install only** option is selected. This option installs the report server files, and does not configure the SharePoint environment for Reporting Services.

    > **NOTE**
    > When the SQL Server installation is complete, follow the other sections of this topic to configure the SharePoint environment. This Includes installing the Reporting Services shared service and creating Reporting Services service applications.

17. Help Microsoft improve SQL Server features and services by clicking the check box to send error reports on the **Error Reporting** page.

    Click **Next**.

18. Review any warnings and then click **Next** on the **Installation Configuration Rules** page.

19. On the **Ready to Install** page, review the installation summary and then click **Next**. The summary will include a **Reporting Services SharePoint Mode** child node that will show a value of **SharePointFilesOnlyMode**. Click **Install**.

20. The installation will take several minutes. You will see the **Complete** page with the features listed and the status of each feature. You may see an information dialog indicating the computer needs to be restarted.

Step 2: Register and Start the Reporting Services SharePoint Service

> **NOTE**
> If you are installing into an existing SharePoint farm, you do not need to complete the steps in this section. The Reporting Services SharePoint service is installed and started when you ran the SQL Server installation wizard as part of the previous section of this document.

The following are the common reasons why you need to manually register the Reporting Services service.

- You installed Reporting Services SharePoint mode before SharePoint was installed.
- The account used to install Reporting Services SharePoint mode, was not a member of the SharePoint farm administrators group. For more information, see the section Setup accounts.

The necessary files were installed as part of the SQL Server installation wizard, but the services need to be registered into the SharePoint farm. The SQL Server 2014 release introduces PowerShell support for Reporting Services in SharePoint mode.

The following steps guide you through opening the SharePoint Management Shell and running cmdlets:

1. Click the **Start** button
2. Click the **Microsoft SharePoint 2013 Products** group.
3. Right-click **SharePoint 2013 Management Shell** click **Run as administrator**. NOTE: the SharePoint commands are not recognized in the standard Windows PowerShell window. Use the **SharePoint 2013 Management Shell**.
4. Run the following PowerShell command to install the SharePoint service. A successful completion of the command displays a new line in the management shell. **No message is returned** to the management shell when the command completes successfully:

```
Install-SPRSService
```

> **IMPORTANT**
>
> If you see an error message similar to the following:
>
> Install-SPRSService : The term 'Install-SPRSService' **is not recognized** as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.

5. Run the following PowerShell command to install the service proxy. A successful completion of the command displays a new line in the management shell. **No message is returned** to the management shell when the command completes successfully:

```
Install-SPRSServiceProxy
```

6. Run the following PowerShell command to start the service or see the following notes for instructions on how to start the service from SharePoint Central administration:

```
Get-SPServiceInstance -all |where {$_.TypeName -like "SQL Server Reporting*"} | Start-SPServiceInstance
```

Either you are in the Windows Powershell instead of the SharePoint Management Shell or Reporting Services SharePoint mode is not installed. For more information on Reporting Services and PowerShell, see PowerShell cmdlets for Reporting Services SharePoint Mode.

You can also start the service from SharePoint central Administration rather than running the third PowerShell command. The following steps are also useful to verify that the service is running.

1. In SharePoint Central Administration, click **Manage Services on Server** in the **System Settings** group.

2. Find **SQL Server Reporting Services Service** and click **Start** in the Action column.
3. The status of the Reporting Services service will change from **Stopped** to **Started**. If the Reporting Services service is not in the list, use PowerShell to install the service.

> **NOTE**
> If the Reporting Services service stays in the **Starting** status and does not change to **Started**, verify the 'SharePoint 2013 Administration' service is started in Windows Server Manager.

Step 3: Create a Reporting Services Service Application

This section provides the steps to create a service application and a description of the properties, if you are reviewing an existing service application.

1. In SharePoint Central Administration, in the **Application Management** group, click **Manage Service Applications**.
2. In the SharePoint ribbon, click the **New** button.
3. In the New menu, click **SQL Server Reporting Services Service Application.**.

> **IMPORTANT**
> If the Reporting Services option does not appear in the list, it is an **indication that the Reporting Services shared service is not installed**. Review the previous section on how to use PowerShell cmdlts to install the Reporting Services service.

4. In the **Create SQL Server Reporting Services Service Application** page, enter a name for the application. If you are creating multiple Reporting Services service applications, a descriptive name or naming convention will help you organize your administration and management operations.
5. In **Application Pool** section, create a new application pool for the application (recommended). If you use the same name for both the application pool and the services application, it can make ongoing administration easier. This can also be affected by how many service applications you will create and if you need to use several in a single application pool. See the SharePoint Server documentation on recommendations and best practices for application pool management.

   Select or create a security account for the application pool. Be sure to specify a domain user account. A domain user account enables the use of the SharePoint managed account feature, which lets you update passwords and account information in one place. Domain accounts are also required if you plan to scale out the deployment to include additional service instances that run under the same identity.
6. In the **Database Server**, you can use the current server or choose a different SQL Server.
7. In **Database Name** the default value is `ReportingService_<guid>`, which is a unique database name. If you type a new value, type a unique value. This is the new database to be created specifically for the services application.
8. In **Database Authentication**, the default is Windows Authentication. If you choose **SQL Authentication**, refer to SharePoint documentation for best practices on how to use this authentication type in a SharePoint deployment.
9. In the **Web Application Association** section, select the Web Application to be provisioned for access by the current Reporting Services Service Application. You can associate one Reporting Services service application to one web application. If all of the current web applications are already associated with a

Reporting Services service application, you see a warning message.

10. Click **OK**.
11. The process to create a service application could take several minutes to complete. When it is complete, you will see a confirmation message and a link to a **Provision Subscriptions and Alerts** page. Complete the provision step if you want to use the Reporting Services subscriptions feature or the data alerts feature. For more information, see Provision Subscriptions and Alerts for SSRS Service Applications.

For information on using PowerShell to create a Reporting Services service application, see:

- See the following section Windows PowerShell script for Steps 1-4.
- Topic To create a Reporting Services Service Application using PowerShell.

Step 4: Activate the Power View Site Collection Feature.

Power View, a feature of SQL Server 2014Reporting Services Add-in for Microsoft SharePoint Products, is a site collection feature. The feature is activated automatically for root site collections and site collections created after the Reporting Services add-in is installed. If you plan to use Power View, verify that the feature is activated.

If you install the Reporting Services add-in for SharePoint Products after the installation of the SharePoint Server, then the Report Server integration feature and the Power View integration feature will only be activated for root site collections. For other site collections, manually activate the features.

To Activate or Verify the Power View Site Collection Feature

1. The following steps assume your SharePoint site is configured for the 2013 **experience version**.

   Open your browser to the desired SharePoint site. For example http://<servername>/sites/bi
2. Click **Settings**.
3. Click **Site settings**.
4. In the **Site Collection Administration** group Click **Site collection features**.
5. Find **Power View Integration Feature** in the list.
6. Click **Activate**. The feature status will change to **Active**.

This procedure is completed per site collection. For more information, see Activate the Report Server and Power View Integration Features in SharePoint.

Windows PowerShell script for Steps 1-4

The PowerShells script in this section are the equivalent of completing steps 1 to 4 in the previous sections. The script completes the following:

- Installs Reporting Services service and service proxy, and starts the service.
- Creates a service proxy named "Reporting Services".
- Creates a Reporting Services service application named "Reporting Services Application".
- Enables the Power View feature for a site collection.

Parameters

- Update the **-Account** for the service proxy. The account needs to be a managed service account in the SharePoint farm. For more information, see the SharePoint topic Plan for administrative and service

accounts in SharePoint 2013.

- Update the **-DatabaseServer** parameter for the service application. This parameter is the database engine instance
- Update the **-url** parameter of the site that you want the Power View feature enabled.

To use the script:

1. Open Windows PowerShell with administrative privileges.
2. Copy the following code into the script window.
3. Update the three parameters described in the previous section, and then run the script.

```
#This script Configures SQL Server Reporting Services SharePoint mode

$starttime = Get-Date
Write-Host -foregroundcolor DarkGray StartTime>> $starttime

Write-Host -ForegroundColor Green "Import the SharePoint PowerShell snappin"
Add-PSSnapin Microsoft.Sharepoint.Powershell -EA 0

Write-Host -ForegroundColor Green "Install SSRS Service and Service Proxy, and start the service"
Write-Host -ForegroundColor Green
">>>>
```

```
Write-Host -ForegroundColor Green "Associate service application proxy to default web site and grant web
applications rights to SSRS application pool"
Write-Host -ForegroundColor Green
">>>>
```

Model, and **Report Data Source** content type to a library enables the **New** command so that you can create new documents of that type. For more information, see Add Report Server Content Types to a Library (Reporting Services in SharePoint Integrated Mode).

Activate the Report Server File sync Feature

If users will frequently upload published report items directly to SharePoint document libraries, the **Report Server File Sync** site level feature will be beneficial. The file sync feature will synchronize the report server catalog with items in document libraries on a more frequent basis. For more information, see Activate the Report Server File Sync Feature in SharePoint Central Administration.

Verify the installation

The following are suggested steps and procedures to verify the Reporting Services SharePoint mode deployment.

- See the SharePoint section in the verification topic Verify a Reporting Services Installation.
- In a SharePoint document library, create a basic Reporting Services report that only contains a text box, for example a title. The report does not contain any data sources or datasets. The goal is to verify you can open Report Builder, build a basic report, and preview the report.

  Save the report to the document library and the run the report from the library. For more information on creating reports with Report Builder, see Start Report Builder (Report Builder).

See Also

PowerShell cmdlets for Reporting Services SharePoint Mode
Upgrade and Migrate Reporting Services
Content Roadmap: Set up and configure SharePoint Server and SQL Server BI
Features Supported by the Editions of SQL Server 2012
Reporting Services SharePoint Service and Service Applications

Install Reporting Services SharePoint Mode for SharePoint 2010

10/23/2019 • 10 minutes to read • Edit Online

The procedures in this topic guide you through a single server installation of a Reporting Services report server in SharePoint mode. The steps include running the SQL Server installation wizard as well as additional configuration tasks that use SharePoint 2010 central administration. The topic can also be used for individual procedures for an existing installation, for example to create a Reporting Services service application. For information on adding additional Reporting Services servers to an existing farm, see Add an Additional Report Server to a Farm (SSRS Scale-out) and Add an Additional Reporting Services Web Front-end to a Farm.

Applies to: SharePoint 2010

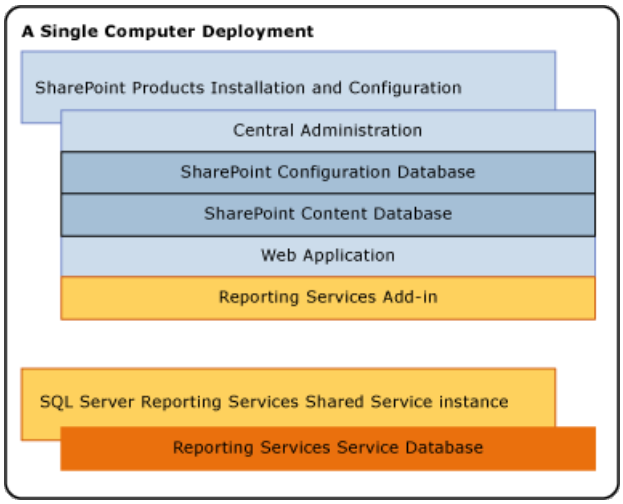
A single server installation is useful for development and testing scenarios but is not recommended for production environments.

> **NOTE**
> For information on upgrading and existing Reporting Services SharePoint mode installation to SQL Server 2014, see Upgrade and Migrate Reporting Services.

Prerequisites

- > **IMPORTANT**
  > The Reporting Services Configuration Manager is no longer required or supported to configure and administer Reporting Services SharePoint mode. Use SharePoint Central Administration to configure a report server in SharePoint mode. For more information, see Manage a Reporting Services SharePoint Service Application.

- Review the following topics for requirements, including SharePoint 2010 products:
  - Online Release Notes
  - Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm
- This topic does no cover the installation of SharePoint 2010 Products. For more information, see Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm.
- These procedures are intended for configuring a SQL Server 2014 report server and do not work for previous versions of report server. Previous versions of report server did not use the SharePoint Shared Service architecture. For example, SQL Server 2008 Reporting Services report servers and SQL Server 2008 R2 Reporting Services report servers.
- Verify the **SharePoint 2010 Administration** service is started in Windows Server Manager.



Database Considerations for a Single Server Configuration

- Both Reporting Services and SharePoint products and technologies use SQL Server relational databases to store application data.
- SQL Server 2014 Reporting Services (SSRS) requires a compatible SQL Server evaluation edition instance of the SQL Engine.
- SharePoint products can use an existing database instance. If an instance of Database Engine is not installed, the SharePoint Products Setup program installs SQL Server Express Edition for the SharePoint application databases.
- The report server instance cannot use the SQL Server Express Edition for its database. However, the SQL Server Express Edition instance that is installed by the SharePoint product can exist side-by-side with other Database Engine editions.

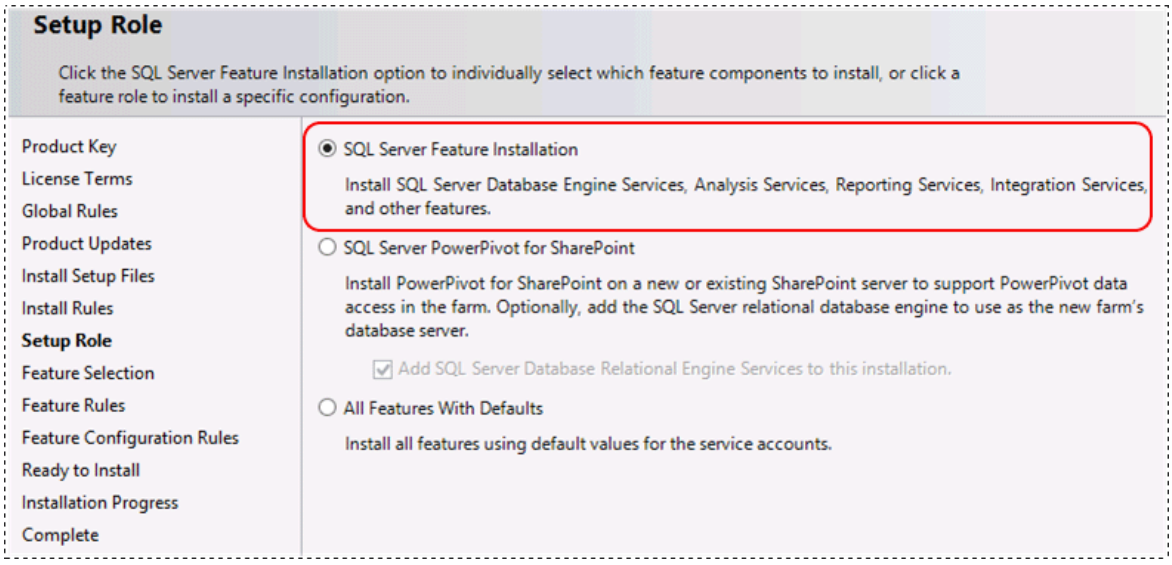
Install Reporting Services Report Server in SharePoint mode

1. Run the SQL Server Installation Wizard.
2. Click **Installation** in the left side of the wizard and then click **New SQL Server stand-alone installation or add features to an existing installation**.
3. Click **OK** on the **Setup Support Rules** page, assuming all rules passed.
4. Click **Install** on the **Setup Support Files** page.
5. Click **Next** after the support files have completed installing and the support rules show a status of **passed**. Review any warnings or blocking issues.
6. On the **Product Key** page, type your key or accept the default of the 'Enterprise Evaluation' edition.

   Click **Next**.
7. Review and accept the license terms. Microsoft appreciates you clicking to agree to send feature usage data to help improve product features and support.

   Click **Next**.
8. Select **SQL Server Feature Installation** on the **Setup Role** page.

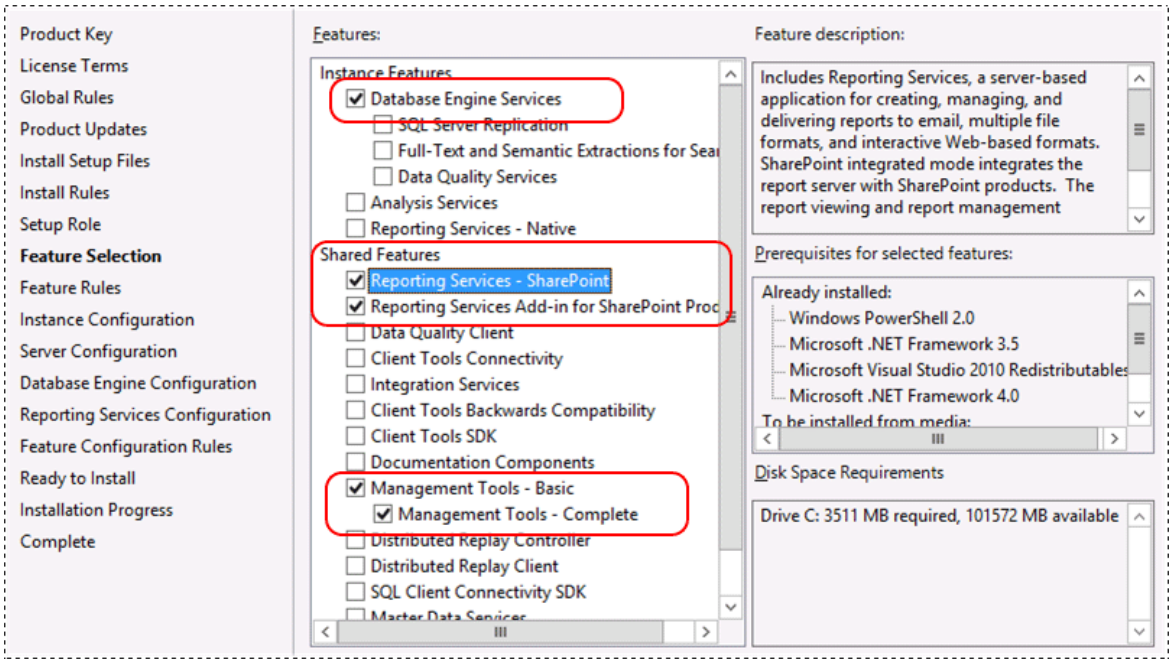
   Click **Next**



9. Select the following on the **Feature Selection** page:

   - **Reporting Services - SharePoint**
   - **Reporting Services add-in for SharePoint 2010 products.** Tip The installation wizard option for installing the add-in is new with the SQL Server 2014 release.
   - If you do not already have an instance of SQL Server Database Engine, you could also select **Database Engine Services** and **Management Tools Complete** for a complete environment.

   Click **Next**.



10. Click **Next** on the **Installation Rules** page. Review any warnings or blocking issues.

11. If you selected the Database Engine services, accept the default instance of **MSSQLSERVER** on the **Instance Configuration** page and click **Next**. The Reporting Services shared service architecture is not based on a SQL Server "instance" was the previous Reporting Services architecture.

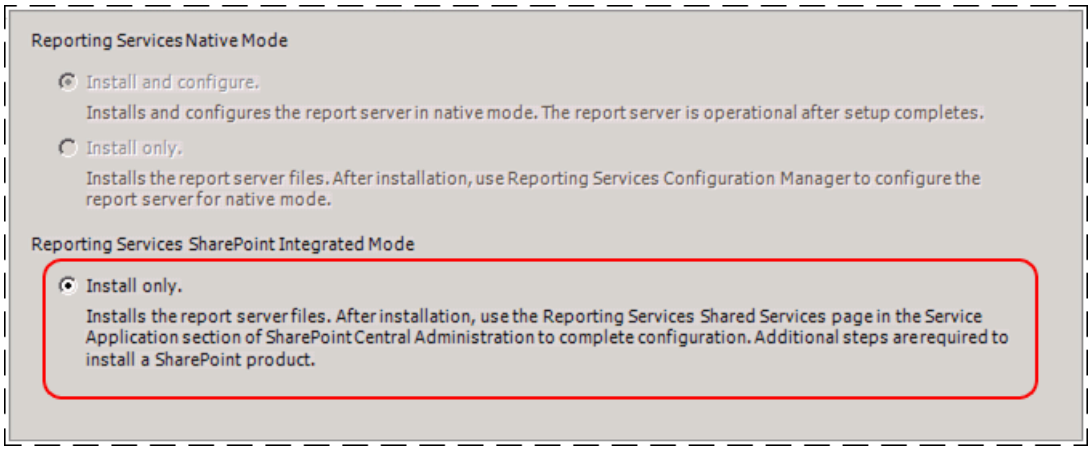
12. Review the **Disk Space Requirements** page and click **Next**.

13. On the **Server Configuration** page type appropriate credentials. If you want to use the Reporting Services data alerting or subscription features, you need to change the **Startup Type** for SQL Server Agent to **Automatic**.

Click **Next**.

14. If you selected the Database Engine services, you will see the **Database Engine Configuration** page, add appropriate accounts to the list of SQL Administrators and click **Next**.

15. On the **Reporting Services Configuration** page you should see the **Install only** option is selected. This option installs the report server files, and does not configure the SharePoint environment for Reporting Services.When the SQL Server installation is complete, follow the other sections of this topic to configure the SharePoint environment. This Includes installing the Reporting Services shared service and creating Reporting Services service applications.



16. Help Microsoft improve SQL Server features and services by clicking the check box to send error reports on the **Error Reporting** page.

    Click **Next**.

17. Review any warnings and then click **Next** on the **Installation Configuration Rules** page.

18. On the **Ready to Install** page, review the installation summary and then click **Next**. The summary will include a **Reporting Services** node that will include the installation mode value of **SharePointFilesOnlyMode** as well as the account information.

Install and Start the Reporting Services SharePoint Service

> **NOTE**
>
> If you are installing into an existing SharePoint farm, **you do not need to** complete the steps in this section. The Reporting Services SharePoint service was installed and started when you ran the SQL Server installation wizard in the previous section.

The necessary files were installed as part of the SQL Server installation wizard, but the services need to be registered into the SharePoint farm. The SQL Server 2014 release introduces PowerShell support for Reporting Services in SharePoint mode. The following steps guide you through opening the SharePoint Management Shell and running cmdlets:

1. Click the **Start** button
2. Click the **Microsoft SharePoint 2010 Products** group.
3. Right-click **SharePoint 2010 Management Shell** click **Run as administrator**.
4. Run the following PowerShell command to install the SharePoint service. A successful completion of the

command displays a new line in the management shell. No message is returned to the management shell when the command completes successfully:

```
Install-SPRSService
```

5. Run the following PowerShell command to install the service proxy:

```
Install-SPRSServiceProxy
```

6. Run the following PowerShell command to start the service or see the following notes for instructions to start the service from SharePoint Central administration:

```
Get-SPServiceInstance -All | Where {$_.TypeName -like "SQL Server Reporting*"} | Start-
SPServiceInstance
```

You can also start the service from SharePoint central Administration rather than running the third PowerShell command. The following steps are also useful to verify that the service is running.

1. In SharePoint Central Administration, click **Manage Services on Server** in the **System Settings** group.
2. Find **SQL Server Reporting Services Service** and click **Start** in the Action column.
3. The status of the Reporting Services service will change from **Stopped** to **Started**. If the Reporting Services service is not in the list, use PowerShell to install the service.

> **NOTE**
> If the Reporting Services service stays in the **Starting** status and does not change to **Started**, verify the 'SharePoint 2010 Administration' service is started in Windows Server Manager.

Create a Reporting Services Service Application

This section provides the steps to create a service application and a description of the properties, if you are reviewing an existing service application.

1. In SharePoint Central Administration, in the **Application Management** group, click **Manage Service Applications**.
2. In the SharePoint ribbon, click the **New** button.
3. In the New menu, click **SQL Server Reporting Services Service Application.**.

> **WARNING**
> If the Reporting Services option does not appear in the list, it is an **indication that the Reporting Services shared service is not installed**. Review the previous section on how to use PowerShell cmdlts to install the Reporting Services service.

4. In the **Create SQL Server Reporting Services Service Application** page, enter a name for the application. If you are creating multiple Reporting Services service applications, a descriptive name or naming convention helps you organize your administration and management operations.

5. In **Application Pool** section, create a new application pool for the application (recommended). Using the same name for the new application pool as the service application, makes ongoing administration easier.

   Select or create a managed account for the application pool. Be sure to specify a domain user account. A domain user account enables the use of the SharePoint managed account feature, which lets you update passwords and account information in one place. Domain accounts are also required if you plan to scale out the deployment to include additional service instances that run under the same identity.

6. In the **Database Server**, you can use the current server or choose a different SQL Server.

7. In **Database Name** the default value is `ReportingService_<guid>`, which is a unique database name. If you type a new value, type a unique value.

8. In **Database Authentication**, the default is Windows Authentication. If you choose **SQL Authentication**, refer to the SharePoint administrator guide for best practices on how to use this authentication type in a SharePoint deployment.

9. In the **Web Application Association** section, select the Web Application to be provisioned for access by the current Reporting Services Service Application. You can associate one Reporting Services service application to one web application. If all of the current web applications are already associated with a Reporting Services service application, you see a warning message.

10. Click **OK**.

11. The process to create a service application could take several minutes to complete. When it is complete, you will see a confirmation message and a link to a **Provision Subscriptions and Alerts** page. Complete the provision step if you want to use the Reporting Services subscriptions and alerts features. For more information, see Provision Subscriptions and Alerts for SSRS Service Applications.

For information on using PowerShell to create a Reporting Services service application, see To create a Reporting Services Service Application using PowerShell.

Activate the Power View Site Collection Feature.

Power View, a feature of SQL Server 2014Reporting Services Add-in for MicrosoftSharePoint Server 2010 Enterprise Edition, is a site collection feature. The feature is activated automatically for root site collections and site collections created after the Reporting Services add-in is installed. If you plan to use Power View, verify that the feature is activated.

If you install the Reporting Services add-in for SharePoint 2010 Products after the installation of the SharePoint 2010 product, then the Report Server integration feature and the Power View integration feature will only be activated for root site collections. For other site collections, manually activate the features.

To Activate the Power View Feature

1. Open your browser to the desired SharePoint site.

2. Click **Site Actions**.

3. Click **Site Settings**.

4. Click **Site Collection Features** in the Site Collection Administration Group.

5. Find **Power View Integration Feature** in the list.

6. Click **Activate**.

This procedure is completed per site collection. For more information, see Activate the Report Server and Power View Integration Features in SharePoint .

Additional Configuration

This section describes additional configuration steps that are important in most SharePoint deployments.

Provision Subscriptions and Alerts

The Reporting Services subscription and data alert features may require the configuration of SQL Server Agent permissions. If you see an error message that indicates SQL Server Agent is required and you have verified SQL Server Agent is running, update the permissions. You can click the link **Provision Subscriptions and Alerts** on the create service application success page to go to another page for provisioning SQL Server Agent. The provision step is needed if your deployment crosses machine boundaries, for example when the SQL Server database instance is on a different machine. For more information, see Provision Subscriptions and Alerts for SSRS Service Applications

Configure e-mail for a service application

The Reporting Services data alerts feature sends alerts in e-mail messages. To send e-mail you may need to configure your Reporting Services service application and you may need to modify the e-mail delivery extension for the service application. If you plan to use the e-mail delivery extension for the Reporting Services subscription feature, the e-mail settings are required. For more information, see Configure E-mail for a Reporting Services Service Application (SharePoint 2010 and SharePoint 2013)

Add Reporting Services Content Types

Reporting Services provides predefined content types that are used to manage shared data source (.rsds) files, report models (.smdl), and Report Builder report definition (.rdl) files. Adding a **Report Builder Report**, **Report Model**, and **Report Data Source** content type to a library enables the **New** command so that you can create new documents of that type. For more information, see Add Report Server Content Types to a Library (Reporting Services in SharePoint Integrated Mode).

Activate the File sync feature

If users will frequently upload published report items directly to SharePoint document libraries, the report server file sync feature will be beneficial. The file sync feature will synchronize the report server catalog with items in document libraries on a more frequent basis. For more information, see Activate the Report Server File Sync Feature in SharePoint Central Administration.

See Also

PowerShell cmdlets for Reporting Services SharePoint Mode
Features Supported by the Editions of SQL Server 2012
Reporting Services SharePoint Service and Service Applications

Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013)

2/28/2020 • 12 minutes to read • Edit Online

Run the installation package Microsoft SQL Server Reporting Services Add-in for SharePoint products (rsSharePoint.msi) on SharePoint servers to enable Reporting Services features within a SharePoint deployment. Features include Power View, a Report Viewer Web Part, a URL proxy endpoint, content types, and application pages so that you can create, view, and manage reports, report models, data sources and other report server content on a SharePoint site. The Reporting Services Add-in for SharePoint products is a required component for a report server that runs in SharePoint mode. The add-in can be installed from either the SQL Server 2014 setup wizard or by downloading the rsSharePoint.msi from the SQL Server 2014 feature pack. For a list of the versions of the add-in and download pages, see Where to find the Reporting Services add-in for SharePoint Products.

Applies to: SharePoint 2013 | SharePoint 2010

In this topic:

- Prerequisites
- What Does The Add-in Install?
- Overview of the Installation Methods
- Install the add-in using the installation file rsSharePoint.msi
  - Files-only installation
- How to Remove the Reporting Services Add-in
- How to Repair rssharepoint.msi from the Command Line
- Setup Log Files
- Upgrade
- RsCustomAction.exe

Prerequisites

Installing the Reporting Services Add-in is one of several steps that are necessary for integrating a report server with an instance of a SharePoint product. For more information about the complete set of requirements for using SharePoint mode, see Hardware and Software Requirements for Reporting Services in SharePoint Mode. For more information on installing and configuring Reporting Services, see Install Reporting Services SharePoint Mode for SharePoint 2013.

- If you are integrating Reporting Services with a SharePoint farm that has multiple Web front end applications, install the add-in 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.
- To install the Reporting Services Add-in, you must be an administrator on the computer. For example if you are going to run the rsSharePoint.msi at the command prompt, you should open the command prompt with administrator privileges by using the **Run as administrator** option.

- To install the Reporting Services Add-in, you must be a member of the SharePoint Farm Administrators group.
- You must be a Site Collection administrator to activate the Reporting Services integration feature.
- For diagrams of example deployments with the add-in, see Deployment Topologies for SQL Server BI Features in SharePoint.

What Does The Add-in Install?

The add-in setup process is composed of two phases, both are completed automatically when you complete a standard installation:

- The first phase is to install files to the proper folders. The folders are standard for SharePoint deployments. One of the files that is installed is rsCustomAction.exe.
- The second portion of the installation is to run a set of custom actions to register the Reporting Services files with SharePoint. The custom actions are run from rsCustomAction.exe. The exe is removed when the full two phase installation completes. You can run a **files only** installation and rsCustomAction.exe is not run at the end of installation and it is left on the drive.

The Reporting Services Installation order

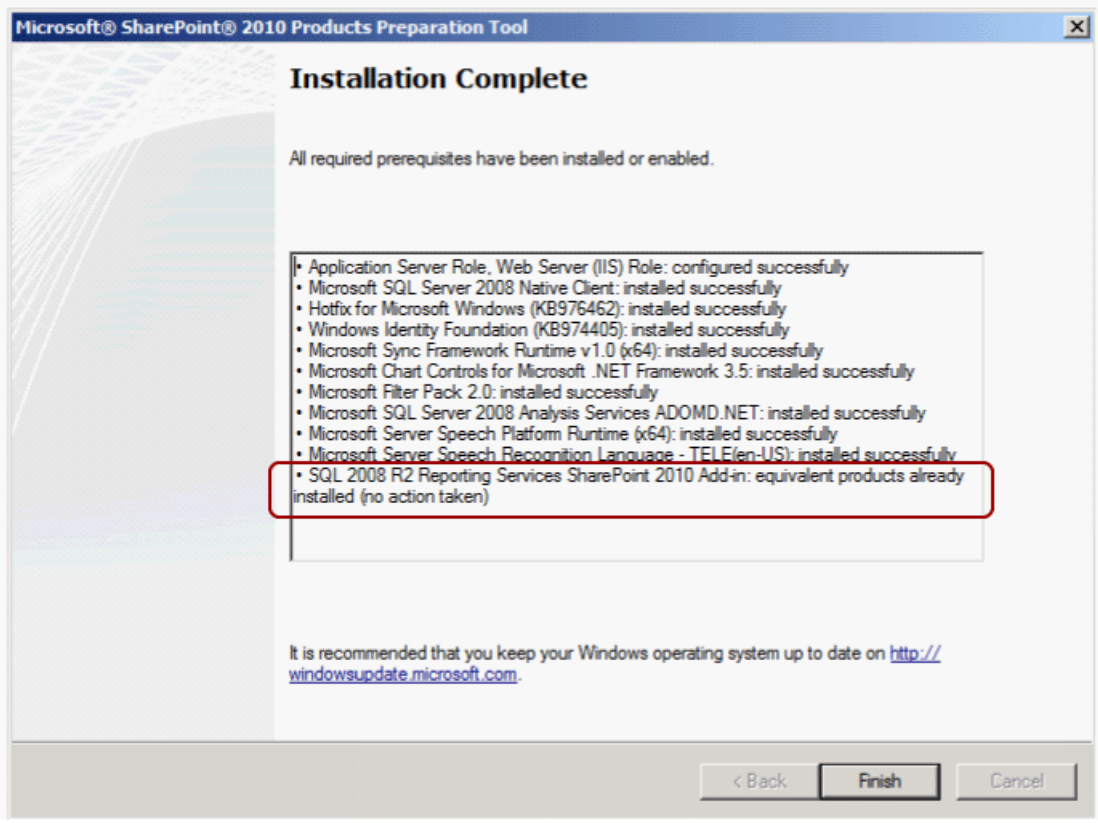
The add-in can be installed before installing SharePoint or after SharePoint installation. The add-in follows SharePoint pre-deployment standards and installs files in locations used by the SharePoint installation.

> **NOTE**
> The advantage of installing the add-in prior to the SharePoint product is that as new servers are added to the farm, the Reporting Services Add-in will be configured and activated by the SharePoint farm.

SharePoint 2010

- The SharePoint 2010 Products Preparation tool installs the SQL Server 2008 R2 version of the Reporting Services add-in. SQL Server 2014 includes a new version of the add-in that is required for SQL Server 2014 features.

  If you run the SharePoint Products Preparation Tool, you still need to install the SQL Server 2014 version of the Reporting Services Add-in.
- If you install SQL Server 2014 version of the Reporting Services add-in first, then when you run the SharePoint Products preparation tool you will see the following dialog indicating the preparation tool did not install the older version of the add-in as the newer version was detected. This is expected behavior



SharePoint 2013

The SharePoint 20103 Products Preparation tool does **Not** install the Reporting Services add-in for SharePoint products.

Overview of the Installation Methods

The SQL Server 2014 Reporting Services Add-in for SharePoint products can be installed using one of the following two methods:

- **The installation wizard:** Tip New with SQL Server 2014, the add-in can be installed by the SQL Server installation wizard. Choose **Reporting Services Add-in for SharePoint Products** on the **Feature Selection** page of the wizard.
- **rsSharepoint.msi:** The add-in can be installed directly from the installation media or downloaded and installed. The rsSharepoint.msi supports both a graphical user interface and a command line installation. You must run the .msi with administrator privileges by first opening a command prompt with elevated permissions, and then running the rsSharepoint.msi from the command line. For more information on downloading the add-in, see Where to find the Reporting Services add-in for SharePoint Products.

> **NOTE**
> If you use the **/q** switch for a silent command line installation, the end-user license agreement will not be displayed. Regardless of the installation method, the use of this software is governed by a license agreement and you are responsible for complying with the license agreement.

Install the add-in using the installation file rsSharePoint.msi

This section is related to installing the rssharepoint.msi directly, by either running the .msi installation wizard or a command line installation. If you installed the add-in using the SQL Server installation Wizard, you do not need to follow these steps.

You can see a full list of command line switches by running the following command:

```
Rssharepoint.msi /?
```

1. Download the Setup program (`rsSharepoint.msi`) for the Reporting Services Add-in. For more information on downloading the add-in, see Where to find the Reporting Services add-in for SharePoint Products.

2. As an administrator, run `rsSharepoint.msi` to run the Installation Wizard. The wizard displays a Welcome page, the Software license terms, and a registration information page. Setup creates folders under the following path and copies files to the folders:

   `%program files%\common files\Microsoft Shared\Web Server Extensions\14\`

   or

   `%program files%\common files\Microsoft Shared\Web Server Extensions\15\`

3. Configure the report server settings and feature activation in SharePoint Central Administration. . For more information on installing and configuring Reporting Services SharePoint mode, see Install Reporting Services SharePoint Mode for SharePoint 2010.

Files-only installation

To install the files but skip the custom action phase of installation, run the rssharepoint.msi from the command line with the SKIPCA option:

1. Open a command prompt **with administrator permissions**.

2. Run the following command:

   ```
   Msiexec.exe /i rsSharePoint.msi SKIPCA=1
   ```

The installation user interface will open and run as normal and the `rsCustomAction.exe` file is installed. However, the .exe will not run at the end of the installation and `rsCustomAction.exe` will remain on the computer when the installation is completed.

Use a two-step installation to troubleshoot installation issues or install the content types

If you get errors during installation or the Reporting Services content types do not appear in the document library settings, you can run Setup as a two-step process from the command line:

1. Open a command prompt **with administrator permissions** and run a files only installation as described in the previous section.

2. Run the custom actions executable:

   a. Navigate to the folder that contains the file `rsCustomAction.exe`. This file is copied to your computer by the files only installation of the add-in. `rsCustomAction.exe` is located in the **%Temp%** directory. To navigate to the file, type the following from the command prompt:

      CD %temp%.

      The file should be located in: **\Users\<your name>\AppData\Local\Temp**

   b. Type the following command. This configuration step will take several minutes to finish. The W3SVC service will be restarted during this process. Several Status messages will be displayed as the program copies files, registers components, and runs the SharePoint Product Configuration Wizard.

```
rsCustomAction.exe /i
```

c. The amount of time it takes for the changes to take effect may vary, depending on your server environment. You can also run **iisreset** to force a quicker update.

Quiet installation for scripting

You can use the **/q** or **/quiet** switches for a "quiet" installation that will not display any dialogs or warnings. The quiet installation is useful if you want to script the installation of the add-in.

> **NOTE**
> If you use the **/q** switch for a silent command line installation, the end-user license agreement will not be displayed. Regardless of the installation method, the use of this software is governed by a license agreement and you are responsible for complying with the license agreement.

To perform a quiet installation:

1. Open a command prompt **with administrator permissions**.
2. Run the following command:

```
Msiexec.exe /i rsSharePoint.msi /q
```

How to Remove the Reporting Services Add-in

You can uninstall the Reporting Services Add-in for SharePoint Products from Microsoft Windows control panel or the command line.

1. Using control panel will run a complete uninstall of the files on the current computer **AND** it will remove the Reporting Services object and features from the SharePoint farm. When the Reporting Services object and features are removed you can no longer review and update reports.
2. The command line method to uninstall the add-in allows you to use the LocalOnly parameter to only remove the add-in files from the local computer and the Reporting Services object and features in the farm will not be changed.

Uninstalling the add-in will remove server integration features that are used to process reports on a report server. It will also remove the Reporting Services pages from SharePoint Central Administration and other custom Reporting Services pages. You may also want to remove any reports and other report server items that you no longer use on the affected SharePoint sites. They will not run after the Reporting Services Add-in is removed.

To uninstall the Reporting Services Add-in, you must have a SharePoint Foundation 2010 or SharePoint Server 2010 installation still running. If you uninstall the SharePoint 2010 first, you must reinstall it to uninstall the Reporting Services Add-in.

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.

Uninstalling the add-in will not remove the following:

- Logins created for the Report Server service account that is 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), shared data source (.rsds), and published report items (.rsc) 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.

To Uninstall from Windows Control Panel

To start the wizard from Microsoft Windows Control Panel and remove the add-in:

1. In Control Panel, in **Programs**, select **Uninstall a Program**
2. Select **Microsoft SQL Server RS Add-in for SharePoint**. You can also start the uninstall wizard by running **rssharepoint.msi** from the command prompt with no switches.
3. Click **Remove**.

Uninstall from the command line

To uninstall the add-in from the command line:

1. Open a command prompt **with administrator permissions**.
2. Run the following command:

```
msiexec.exe /uninstall rsSharePoint.msi
```

3. You will see a confirmation message box. Click **Yes**.

Uninstall the add-in from the local server only

The previous methods of uninstalling the add-in will remove the Reporting Services features and object from the farm. If you have a multi-server farm and want to uninstall the add-in from only the local computer and leave the SharePoint farm in a functional state, complete the following steps:

1. Open a command prompt **with administrator permissions**.
2. Run the following command:

```
Msiexec.exe /uninstall rsSharePoint.msi LocalOnly=1
```

This will unregister the Reporting Services components from SharePoint and remove the files, but for the local computer only.

If you want to unregister the Reporting Services features from SharePoint but leave the files on the disk for use later, complete the following steps:

1. Open a command prompt **with administrator permissions**.
2. Run the following command:

```
rsCustomAction.exe /p
```

The above steps assume you completed an installation of the .msi with SkipCA=1 and the rscusstomaction.exe is

available. For more information, see the section describing the files only installation.

How to Repair rssharepoint.msi from the Command Line

To repair or uninstall the Reporting Services add-in using the command line, complete the following steps:

1. Open a command prompt **with administrator permissions**.
2. Run the following command:

```
msiexec.exe /f rssharepoint.msi
```

Setup Log Files

When Setup runs, it logs information to a log file in the **%temp%** folder for the user who installed the Reporting Services Add-in. For example **c:\Users\<username>\AppData\Local\Temp** .The file name is **RS_SP_<number>.log**, for example **RS_SP_0.log**. Each error in the log starts with the string "SSRSCustomActionError".

> **NOTE**
> AppData is a hidden folder in the Windows operating system. You may need to modify your Windows Explorer folder settings so you can see hidden files and folders.

View a log file with Windows Notepad

1. The following commands will change the command prompt path, list the rs log files and then open one of the files with Windows Notepad:

```
cd %temp%
```

```
dir rs_sp*.log
```

```
notepad rs_sp_3.log
```

View a Log file with PowerShell

1. Type the following command from the SharePoint Management Shell to return a filtered list of rows from the file, that contain "ssrscustomactionerror":

```
Get-Content -Path C:\Users\<UserName\AppData\Local\Temp\rs_sp_0.log | Select-String
"ssrscustomactionerror"
```

2. The output will look similar to the following:

```
2011-05-23 12:40:12: SSRSCustomActionError: SharePoint is installed, but not configured
```.

Upgrade

If you have an existing installation of the Reporting Services Add-in, you can upgrade to the current version. The add-in setup will detect the existing version and prompt you to confirm the update. The message will be similar to the following:

A Lower version of this product has been detected on your system. Would you like to upgrade your existing installation?

If you confirm, the older version of the add-in will be removed and then the new version will be installed.

Note that the Reporting Services Add-in is not instance-aware. You can only have one instance of the add-in on a computer. You cannot run different versions side-by-side the current version.

RsCustomAction.exe

The following table summarizes the rscustomaction.exe switches:

SWITCH	DESCRIPTION
i	Install the custom actions. This will register the Reporting Services components in SharePoint. This will restart the W3SVCservice.
r	Repair
u	Uninstall. This will unregister the Reporting Services components from the entire SharePoint farm but leave the files on disk. This will restart the W3SVCservice.
p	Local uninstall. This will unregister the Reporting Services components from only the local computer. The files will remain on disk. This will restart the W3SVCservice.
t	SQL Server Reporting Services 2005 only. The switch tests if the report server has a working connection to the report server database.

Configuring Reporting Services

After you have installed the add-in on all the necessary computers, you need to configure the report server from SharePoint Central Administration. The steps that are needed depend on the order which the different technologies were installed. For more information, see Install Reporting Services SharePoint Mode for SharePoint 2010 and Reporting Services Report Server (SharePoint Mode)

See Also

Install Reporting Services SharePoint Mode for SharePoint 2010 Reporting Services Report Server (SharePoint Mode)

Where to find the Reporting Services add-in for SharePoint Products

3/11/2020 • 2 minutes to read • Edit Online

The Microsoft SQL Server Reporting Services (SSRS) add-in for SharePoint Products and Technologies (rsSharePoint.msi) is a Web download that provides features to integrate a report server with a deployment of SharePoint.

> **IMPORTANT**
>
> For a list of the supported combinations of the Reporting Services add-in, report server, and SharePoint, see For more information, see Supported Combinations of SharePoint and Reporting Services Server and Add-in (SQL Server 2014).

SQL Server 2014 Reporting Services Add-in for SharePoint Products

To download and install the add-in see the Microsoft Download Center:

- Microsoft® SQL Server 2014 Reporting Services Add-in for Microsoft SharePoint

The SQL Server 2014 version of the add-in is also available in the SQL Server 2014 Installation wizard:

- On the **Feature Selection** page of the setup wizard, select **Reporting Services Add-in for SharePoint Products**
- From a command prompt installation, use the **RS_SHPWFE** option to install the add-in. For more information on Reporting Services command prompt installations, see Command Prompt Installation of Reporting Services SharePoint Mode and Native Mode.

SQL Server 2012 SP1 Reporting Services Add-in for SharePoint Products

The SQL Server 2012 SP1 version of the add-in and report server, add support for SharePoint Server 2013.

To download and install the add-in see the Microsoft Download Center:

- **SP1 add-in:** Microsoft® SQL Server® 2012 SP1 Reporting Services Add-in for Microsoft® SharePoint® (https://www.microsoft.com/download/details.aspx?id=35583).
- **SP1:** Microsoft® SQL Server® 2012 Service Pack 1 (SP1) (https://www.microsoft.com/download/details.aspx?id=35575).

SQL Server 2012 Reporting Services Add-in for SharePoint 2010 Products

Starting with the SQL Server 2014 release the add-in can be installed as part of the SQL Server Installation wizard, in the Feature Selection page. If you want to download and install the add-in separately, the most up-to-date version of this file is available online at the Microsoft Download Center at Microsoft® SQL Server® 2012 Reporting Services Add-in for Microsoft® SharePoint® Technologies 2010 page.

SQL Server 2008 R2 Reporting Services Add-in for SharePoint 2010

Products

The Microsoft SQL Server 2008 R2 Reporting Services Add-in for Microsoft SharePoint 2010 products is installed by the SharePoint 2010 Products Preparation tool (**PreRequisiteInstaller.exe**). If you want to download and install the add-in separately, the most up-to-date version of this file is available online at the Microsoft Download Center, SQL Server 2008 R2 Reporting Services Add-in for Microsoft SharePoint Technologies 2010.

> **TIP**
> If using this with SQL Server 2008, Service Pack 1 (SP1) with Cumulative Update #8 or higher, review the Known Issues with SharePoint 2010 Integration section of Integrating SQL Server 2008 SP2 Report Servers with SharePoint 2010

SQL Server 2008 SP2 Reporting Services Add-in for SharePoint 2007 Products and Technologies

Microsoft SQL Server 2008 SP2 Reporting Services Add-in is an updated version of the 2008 add-in that adds support integration of 2008 R2 report servers with SharePoint 2007 products.

The most up-to-date version of this file is available online at the Microsoft Download Center, Microsoft SQL Server 2008 SP2 Reporting Services Add-in for Microsoft SharePoint Technologies.

SQL Server 2008 Reporting Services Add-in for SharePoint 2007 Products and Technologies

The Microsoft SQL Server 2008 Reporting Services Add-in for Microsoft SharePoint Technologies provides features for running a report server within a deployment of Windows SharePoint Services 3.0 or Microsoft Office SharePoint Server 2007.

The most up-to-date version of this file is available online at the Microsoft Download Center, Microsoft SQL Server 2008 Reporting Services Add-in for Microsoft SharePoint Technologies.

See Also

Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013)
Add Report Server Content Types to a Library (Reporting Services in SharePoint Integrated Mode)
You are not able to browse SharePoint pages in non-default zone after uninstalling Reporting Services add-in

Add an Additional Report Server to a Farm (SSRS Scale-out)

5/22/2019 • 6 minutes to read • Edit Online

Adding a second or more SharePoint mode report servers to your SharePoint farm can improve the performance and response time of the report server processing. If you find performance slowing down as you added more users, reports, and other applications to the report server, then adding additions report servers can improve performance. It is also recommended to add a second report server to increase the availability of report servers when there are issues with hardware or you are conducting general maintenance on individual servers in your environment. Starting with the SQL Server 2012 release, the steps to scale-out a Reporting Services environment in SharePoint mode follows standard SharePoint farm deployment and leverages the SharePoint load balancing features.

IMPORTANT

Scale-out of Reporting Services is not supported on all editions of SQL Server. For more information, see the Reporting Services section of Features Supported by the Editions of SQL Server 2014.

TIP

Starting with SQL Server 2012 you do not use Reporting Services Configuration Manager to add servers and scale out report servers. SharePoint products manage the scale-out of reporting services as SharePoint servers with the Reporting Services service are added to the farm.

For information on how to scale-out native mode report servers, see Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager).

- Load Balancing
- Prerequisites
- Steps
- Additional Configuration

Load Balancing

The Load balancing of Reporting Services service applications will be managed automatically by SharePoint unless your environment has a custom or third-party load balancing solution. The default SharePoint load balancing behavior is that each Reporting Services Service Application will be balanced across all the application servers where you have started the Reporting Services service. To verify if the Reporting Services service is installed and started, click **Manage services on server** in SharePoint Central Administration.

Prerequisites

- You must be a local administrator to run SQL Server Setup.
- The computer must be joined to a domain.
- You need to know the name of the existing database server that is hosting the SharePoint configuration and

content databases.

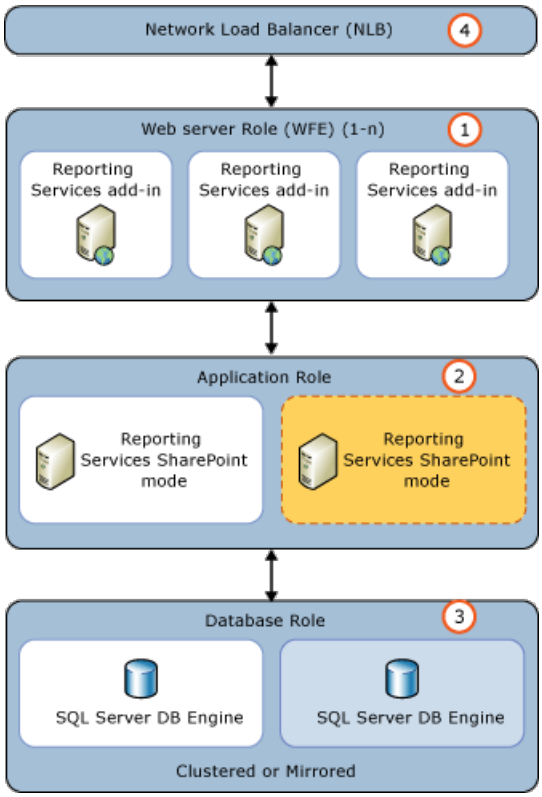
- The database server must be configured to allow for remote database connections. If it is not, you will not be able to join the new server to the farm because the new server will not be able to make a connection to the SharePoint configuration databases.
- The new server will need to have the same version of SharePoint installed that the current farm servers are running. For example if the farm already has SharePoint 2010 Service Pack 1 (SP1) installed, you will need to also install SP1 on the new server before it can join the farm.
- Review the following additional topics to understand system and version requirements:

  Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm

Steps

The steps in this topic assume that a SharePoint farm administrator is installing and configuring the server. The diagram shows a typical three tier environment and the numbered items in the diagram are described in the following list:

- (1) Multiple web front-end (WFE) servers. The WFE servers require the Reporting Services add-in for SharePoint 2010.
- (2) A single application server running Reporting Services and web sites, for example Central Administration. The following steps add a second application server to this tier.
- (3) Two SQL Server database servers.
- (4) Represents a software or hardware network load balancing solution (NLB)



The following steps assume that an administrator is installing and configuring the server. The server will be setup as a new application server in the farm and not used as a web front-end (WFE).

STEP	DESCRIPTION AND LINK
Run the SharePoint 2010 Products Preparation Tool	You must have the SharePoint 2010 installation media. The preparation tool is **PrerequisiteInstaller.exe** on the installation media.
Install a SharePoint 2010 product.	1) Select the **Server Farm** installation type. 2) Select **Complete** for the server type. 3) When the installation is complete, do not run the SharePoint Products Configuration wizard if your existing SharePoint farm has SharePoint 2010 SP1 installed. You should install SharePoint SP1 before running the SharePoint products configuration wizard.
Install SharePoint Server 2010 SP1.	If your existing SharePoint farm has SharePoint 2010 SP1 installed download and install SharePoint 2010 SP1 from:https://support.microsoft.com/kb/2460045. For more information on SharePoint 2010 SP1, see Known issues when you install Office 2010 SP1 and SharePoint 2010 SP1:
Run the SharePoint Products Configuration wizard to add the server to the farm.	1) In the **Microsoft SharePoint 2010 Products** program group, click **Microsoft SharePoint 2010 Products Configuration Wizard**. 2) On the **Connect to a Server Farm** page select **Connect to an existing Farm** and click **Next**. 3) On the **Specify Configuration Database Settings** page, type the name of the database server used for the existing farm and the name of the configuration database. Click **Next**. ** **Important** ** If you see an error message similar to the following and you have verified you have permissions, then verify what protocols are enabled for the SQL Server Network Configuration in **Sql Server Configuration Manager**:"Failed to connect to the database server. Ensure the database exists , is a Sql Server, and that you have the appropriate permissions to access the server." ** **Important** ** If you see the page **Server Farm Product and Patch Status**, you will need to review the information on the page and update the server with the needed files before you can proceed with joining the server to the farm. 4) On the **Specify Farm Security Settings** page type your farm passphrase and click **Next**. Click **Next** on the confirmation page to run the wizard. 5) Click **Next** to run the **Farm Configuration Wizard**.

STEP	DESCRIPTION AND LINK
Verify the server was added to the SharePoint farm.	1) In SharePoint Central Administration, click **Manage servers in this farm** in the **System Settings** group. 2) Verify the new server is added and the status is correct. 3) Note you do not see the service **SQL Server Reporting Services Service** running. The service will be installed in the next step. 4) To remove this server from the WFE role, click **Manage services on server** and stop the service **Microsoft SharePoint Foundation Web Application**.
Install and configure Reporting Services SharePoint mode.	Run SQL Server 2014 installation. For more information on the installation of Reporting Services SharePoint mode, see Install Reporting Services SharePoint Mode for SharePoint 2010 If the server will only be used as an application server and the server will not be used as a WFE, you do not need to select **Reporting Services add-in for SharePoint products** on: the **Setup Role** page, select **SQL Server Feature Installation** the **Feature Selection** page, select **Reporting Services - SharePoint** -OR- the **Reporting Services Configuration** page verify the **Install Only** option is selected for **Reporting Services SharePoint Mode**.
Verify that Reporting Services is operational.	1) In SharePoint Central Administration, click **Manage servers in this farm** in the **System Settings** group. 2) Verify the service **SQL Server Reporting Services Service**. For more information, see Verify a Reporting Services Installation

Additional Configuration

You can optimize individual Reporting Services servers in a scaled out deployment to perform background processing only so they do not compete for resources with interactive report execution. Background processing includes schedules, subscriptions, and data alerts.

To change the behavior of individual report servers, set **<IsWebServiceEnable>** to false in the **RSreportServer.config** configuration file.

By default reports servers are configured with <IsWebServiceEnable> set to TRUE. When all servers are configured for TRUE, interactive and background will be load balanced across all nodes in the farm.

If you configure all report servers with <IsWebServiceEnable> set to False, you will see an error message similar to the following when you try to use Reporting Services features:

The Reporting Services Web Service is not enabled. Configure at least one instance of the Reporting Services

SharePoint Service to have <IsWebServiceEnable> set to true. For more information, see Modify a Reporting Services Configuration File (RSreportserver.config)

See Also

Add web or application servers to farms in SharePoint 2013
Configure services (SharePoint Server 2010)

Add an Additional Reporting Services Web Front-end to a Farm

2/28/2020 • 4 minutes to read • Edit Online

Reporting Services SharePoint mode includes components needed for application servers and web front-end (WFE) servers. This topic focuses on installing the required Reporting Services components for a WFE server, including the application pages used by Reporting Services features such as subscriptions, data alerts, and Power View. The primary Reporting Services installation needed for a WFE is to install the Reporting Services add-in for SharePoint 2010 products.

Prerequisites

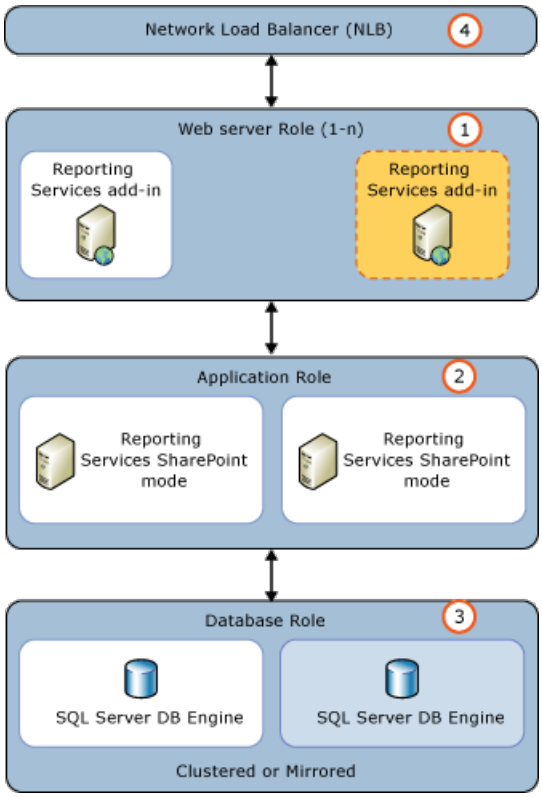
- You must be a local administrator to run SQL Server Setup.
- The computer must be joined to a domain.
- You need to know the name of the existing database server that is hosting the SharePoint configuration and content databases.
- The database server must be configured to allow for remote database connections. If it is not, you will not be able to join the new server to the farm because the new server will not be able to make a connection to the SharePoint configuration databases.
- The new server will need to have the same version of SharePoint installed that the current farm servers are running. For example if the farm already has SharePoint 2010 Service Pack 1 (SP1) installed, you will need to also install SP1 on the new server before it can join the farm.
- Review the following additional topics to understand system and version requirements:

  Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm

Steps

The steps in this topic assume that a SharePoint farm administrator is installing and configuring the server. The diagram shows a typical three tier environment and the numbered items in the diagram are described in the following list:

- (1) Multiple web front-end (WFE) servers. The WFE servers require the Reporting Services add-in for SharePoint 2010. The following steps add a second application server to this tier.
- (2) Two application servers running Reporting Services and web sites, for example Central Administration.
- (3) Two SQL Server database servers.
- (4) Represents a software or hardware network load balancing solution (NLB)



The following steps assume that an administrator is installing and configuring the server.

STEP	DESCRIPTION AND LINK
Run the SharePoint 2010 Products Preparation Tool	You must have the SharePoint 2010 installation media. The preparation tool is **PrerequisiteInstaller.exe** on the installation media.
Install a SharePoint 2010 product.	1) Select the **Server Farm** installation type. 2) Select **Complete** for the server type. 3) When the installation is complete, do not run the SharePoint Products Configuration wizard if your existing SharePoint farm has SharePoint 2010 SP1 installed. You should install SharePoint SP1 before running the SharePoint products configuration wizard.
Install SharePoint Server 2010 SP1.	If your existing SharePoint farm has SharePoint 2010 SP1 installed download and install SharePoint 2010 SP1 from:https://support.microsoft.com/kb/2460045. For more information on SharePoint 2010 SP1, see Known issues when you install Office 2010 SP1 and SharePoint 2010 SP1:

STEP	DESCRIPTION AND LINK
Run the SharePoint Products Configuration wizard to add the server to the farm.	1) In the **Microsoft SharePoint 2010 Products** program group, click **Microsoft SharePoint 2010 Products Configuration Wizard**. 2) On the **Connect to a Server Farm** page select **Connect to an existing Farm** and click **Next**. 3) On the **Specify Configuration Database Settings** page, type the name of the database server used for the existing farm and the name of the configuration database. Click **Next**. ** **Important** ** If you see an error message similar to the following and you have verified you have permissions, then verify what protocols are enabled for the SQL Server Network Configuration in **Sql Server Configuration Manager**. "Failed to connect to the database server. Ensure the database exists , is a Sql Server, and that you have the appropriate permissions to access the server." ** **Important** ** If you see the page **Server Farm Product and Patch Status**, you will need to review the information on the page and update the server with the needed files before you can proceed with joining the server to the farm. 4) On the **Specify Farm Security Settings** page type your farm passphrase and click **Next**. Click **Next** on the confirmation page to run the wizard. 5) Click **Next** to run the **Farm Configuration Wizard**.
Verify the server was added to the SharePoint farm.	1) In SharePoint Central Administration, click **Manage servers in this farm** in the **System Settings** group. 2) Verify the new server is added and the status is correct. 3) To remove this server from the WFE role, click **Manage services on server** and stop the service **Microsoft SharePoint Foundation Web Application**.
Install the SQL Server 2014 Reporting Services add-in for SharePoint 2010 products.	There are several methods for installing the add-in. The following steps use the SQL Server 2014 setup wizard. For more information on installing the add-in, see Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013). Run SQL Server 2014 installation: 1) On the **Setup Role** page, select **SQL Server Feature Installation** 2) On the **Feature Selection** page, select **Reporting Services add-in for SharePoint products** 3) Click **Next** on the next several pages to complete the setup options. For more information on installing Reporting Services, see Install Reporting Services SharePoint Mode for SharePoint 2010

STEP	DESCRIPTION AND LINK
Verify the new server is operational.	1) In SharePoint Central Administration, click **Manage servers in this farm** in the **System Settings** group. 2) Verify the new server is in the list.
Update your NLB solution.	If appropriate, update your hardware or software NLB environment to include the new server.

See Also

Add a Web or application server to the farm (SharePoint Server 2010) Configure services (SharePoint Server 2010)

Configure E-mail for a Reporting Services Service Application (SharePoint 2010 and SharePoint 2013)

10/23/2019 • 2 minutes to read • Edit Online

Reporting Services data alerting sends alerts in e-mail messages. To send e-mail you may need to configure your Reporting Services service application and you may need to modify the e-mail delivery extension for the service application. The e-mail settings are also required if you plan to use the e-mail delivery extension for the Reporting Services subscription feature.

Applies to: Reporting Services SharePoint mode | SharePoint 2010 and SharePoint 2013.

To configure e-mail for the shared service

1. In SharePoint Central Administration, click the **Application Management**.
2. In the **Service Applications** group, click **Manage service applications**.
3. In the **Name** list, click the name of your Reporting Services service application.
4. Click **E-mail Settings** on the **Manage Reporting Services Application** page.
5. Select **Use SMTP server**.
6. In the **Outbound SMTP server** box, type the name of an SMTP server.
7. In the **From address** box, type an e-mail address.

   This address is the sender of all alert e-mail messages.

   The account of the user specified in **From address** must be a managed account that you specified when you configured the application pool for the Reporting Services service application. If you have permission, you can view a list of existing managed accounts on the Service Accounts page in SharePoint Central Administration.
8. Click **OK**.

NTLM Authentication

1. If your email environment requires NTLM authentication and does not allow anonymous access, you need to modify the e-mail delivery extension configuration for your Reporting Services service applications. Change the **SMTPAuthenticate** to use a value of "2". This value cannot be changed from the user interface. The following PowerShell script example, updates the full configuration for the report server e-mail delivery extension for the service application named "SSRS_TESTAPPLICATION". Note some of the nodes listed in the script can also be set from the user interface, for example the "From" address.

```
$app = Get-SPRSServiceApplication | Where {$_.name -like "SSRS_TESTAPPLICATION *"}
$emailCfg = Get-SPRSExtension -Identity $app -ExtensionType "Delivery" -Name "Report Server Email" |
Select -ExpandProperty ConfigurationXml
$emailXml = [xml]$emailCfg
$emailXml.SelectSingleNode("//SMTPServer").InnerText = "your email server name"
$emailXml.SelectSingleNode("//SendUsing").InnerText = "2"
$emailXml.SelectSingleNode("//SMTPAuthenticate").InnerText = "2"
$emailXml.SelectSingleNode("//From").InnerText = "your FROM email address"
Set-SPRSExtension -Identity $app -ExtensionType "Delivery" -Name "Report Server Email" -
ExtensionConfiguration $emailXml.OuterXml
```

2. If you need to verify the name of your service application, run the **Get-SPRSServiceApplication** cmdlet.

```
Get-SPRSServiceApplication
```

3. The following example will return the current values of the e-mail extension for the service application named "SSRS_TESTAPPLICATION".

```
$app = get-sprsserviceapplication | Where {$_.name -like "SSRSTEST_APPLICATION*"}
Get-SPRSExtension -Identity $app -ExtensionType "Delivery" -Name "Report Server Email" | Select -
ExpandProperty ConfigurationXml
```

4. The following example will create a new file named "emailconfig.txt" with the current values of the e-mail extension for the service application named "SSRS_TESTAPPLICATION"

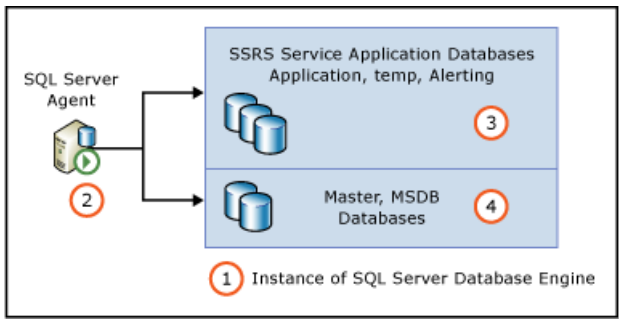
```
$app = Get-SPRSServiceApplication | Where {$_.name -like "SSRS_TESTAPPLICATION*"}
Get-SPRSExtension -identity $app -ExtensionType "Delivery" -name "Report Server Email" | Select -
ExpandProperty ConfigurationXml | Out-File c:\emailconfig.txt
```

Provision Subscriptions and Alerts for SSRS Service Applications

2/28/2020 • 3 minutes to read • Edit Online

Reporting Services subscriptions and data alerts require SQL Server Agent and require the configuration of permissions for SQL Server Agent. If you see error messages that indicate SQL Server Agent is required and you have verified SQL Server Agent is running, then update or verify permissions. The scope of this topic is Reporting Services in SharePoint mode and the topic describes three ways you can update the permissions of SQL Server Agent with Reporting Services subscriptions. The credentials you use for the steps in this topic need to have sufficient permissions to grant execute permissions to the RSExecRole for objects in the service application, msdb, and master databases.

Applies to: SharePoint 2013 | SharePoint 2010



	DESCRIPTION
1	The instance of SQL Server Database engine that is hosting the Reporting Services service application databases.
2	The instance of SQL Server agent for the instance of the SQL database engine.
3	The Reporting Services service application databases. The names are based on the information used for creating the service application. The following are example database names: ReportingService_2fbae157295d49df86d0b85760c704b0 ReportingService_2fbae157295d49df86d0b85760c704b0_Alerting ReportingService_2fbae157295d49df86d0b85760c704b0TempDB
4	The master and MSDB database of the instance of the SQL Server Database engine.

Use one the following three methods to update the permissions:

1. From the **Provisions and Subscriptions and Alerts** page, type credentials and click **ok**.

2. From the Provisions and Subscriptions and Alerts page, click the **Download Script** button to download a transact SQL script that can be used to configure permissions.

3. Run a PowerShell cmdlet to build a transact SQL script that can be used to configure permissions.

To update permissions using the provision page

1. From SharePoint Central Administration, in the **Application Management** group click **Manage Service Applications**

2. Find your service application in the list and click the name of the application or click the **Type** column to select the services application and click the **Manage** button in the SharePoint ribbon.

3. On the **Manage Reporting Services Application** page, click **Provision Subscriptions and Alerts**.

4. If the SharePoint administrator has enough privileges to the Master database and the service application databases, type those credentials.

5. Click the **OK** button.

To download the Transact-SQL Script

1. From SharePoint Central Administration, in the **Application Management** group click **Manage Service Applications**

2. Find your service application in the list and click the name of the application or click the **Type** column to select the services application and click the **Manage** button in the SharePoint ribbon.

3. On the **Manage Reporting Services Application** page, click **Provision Subscriptions and Alerts**.

4. In the **View Status** area, verify SQL Server Agent is running.

5. Click **Download Script** to download a transact SQL script you can run in SQL Server Management studio to grant permissions. The script file name that is created will contain the name of your Reporting Services service application name, for example **[name of the service application]-GrantRights.sql**.

To generate the Transact-SQL statement with PowerShell

1. You can also use a Windows PowerShell cmdlet in the SharePoint 2010 Management Shell to create the Transact-SQL script.

2. On the **Start** menu, click **All Programs**.

3. Expand **Microsoft SharePoint 2010 Products** and click **SharePoint 2010 Management Shell**.

4. Update the following PowerShell cmdlet by replacing the name of the report server database, application pool account, and the path of the statement.

   Syntax of cmdlet:

   ```
   Get-SPRSDatabaseRightsScript -DatabaseName <ReportingServices database name> -UserName <app pool
   account> -IsWindowsUser | Out-File <path of statement>
   ```

   Sample cmdlet:

   ```
   Get-SPRSDatabaseRightsScript -DatabaseName ReportingService_46fd00359f894b828907b254e3f6257c -UserName
   "NT AUTHORITY\NETWORK SERVICE" -IsWindowsUser | Out-File c:\SQLServerAgentrights.sql
   ```

Using the Transact-SQL Script

The following procedures can be used with scripts download from the provisions page or scripts created using PowerShell.

To load the Transact-SQL script in SQL Server Management Studio

1. To open SQL Server Management Studio, on the **Start** menu, click **Microsoft SQL Server 2012** and click **SQL Server Management Studio**.
2. In the **Connect to Server** dialog box set the following options:
   - In the **Server type** list, select **Database Engine**
   - In **Server Name**, type the name of the SQL Server instance on which you want to configure SQL Server Agent.
   - Select an authentication mode.
   - If connecting using SQL Server Authentication, provide a login and password.
3. Click **Connect**.

To run the Transact-SQL statement

1. On the toolbar of SQL Server Management Studio, click **New Query**.
2. On the **File** menu, click **Open**, and then click **File**.
3. Navigate to the folder where you saved the Transact-SQL statement that you generated in SharePoint 2010 Management Shell.
4. Click the file and then click **Open**.

   The statement is added to the query window.
5. Click **Execute**.

Claims to Windows Token Service (C2WTS) and Reporting Services

5/8/2020 • 3 minutes to read • Edit Online

The SharePoint Claims to Windows Token Service (c2WTS) is required with Reporting Services SharePoint mode if you want to use windows authentication for Data Sources that are outside the SharePoint farm. This is true even if the user accesses the data sources with Windows Authentication because the communication between the web front-end (WFE) and the Reporting Services shared service will always be Claims authentication.

c2WTS is needed even if your data source(s) are on the same computer as the shared service. However in this scenario, constrained delegation is not needed.

The tokens created by c2WTS will only work with constrained delegation (constrains to specific services) and the configuration option "using any authentication protocol". As noted earlier, if your data sources are on the same computer as the shared service, then constrained delegation is not needed.

If your environment will use Kerberos constrained delegation, then the SharePoint Server service and external data sources need to reside in the same Windows domain. Any service that relies on the Claims to Windows token service (c2WTS) must use Kerberos **constrained** delegation to allow c2WTS to use Kerberos protocol transition to translate claims into Windows credentials. These requirements are true for all SharePoint Shared Services. For more information, see Overview of Kerberos authentication for Microsoft SharePoint 2010 Products (https://technet.microsoft.com/library/gg502594.aspx).

The procedure is summarized in this topic.

Applies to: SharePoint 2013 | SharePoint 2010

Prerequisites

NOTE

Note: Some of the configuration steps may change, or may not work in certain farm topologies. For instance, a single server install does not support the Windows Identity Foundation c2WTS services so claims to windows token delegation scenarios are not possible with this farm configuration.

Basic steps needed to configure c2WTS

1. Configure the c2WTS service account. Add the service account to the local Administrators group on each application server running c2WTS. In addition, verify that the account has the following local security policy rights:

   - Act as part of the operating system
   - Impersonate a client after authentication
   - Log on as a service

   The account you use for c2WTS also needs to be configured for Constrained Delegation with Protocol Transitioning and needs permissions to delegate to the Services it is required to communicate with (i.e. SQL Server Engine, SQL Server Analysis Services).To configure delegation you can use the Active Directory Users

and Computer snap-in.

a. Right-click each service account and open the properties dialog. In the dialog click the **Delegation** tab.

> **NOTE**
> Note: the delegation tab is only visible if the object has an SPN assigned to it. c2WTS does not require an SPN on the c2WTS Account, however, without an SPN, the **Delegation** tab will not be visible. An alternative way to configure constrained delegation is to use a utility such as **ADSIEdit**.

b. Key configuration options on the delegation tab are the following:

- Select "Trust this user for delegation to specified services only"
- Select "Use any authentication protocol"

For more information, see the "configure Kerberos constrained delegation for computers and service accounts" section of the following white paper, Configuring Kerberos authentication for SharePoint 2010 and SQL Server 2008 R2 products

2. Configure c2WTS 'AllowedCallers'

c2WTS requires the 'callers' identities explicitly listed in the configuration file, **c2wtshost.exe.config**. c2WTS does not accept requests from all authenticated users in the system unless it is configured to do so. In this case the 'caller' is the WSS_WPG Windows group. The c2wtshost.exe.confi file is saved in the following location:

\Program Files\Windows Identity Foundation\v3.5\c2wtshost.exe.config

The following is an example of the configuration file:

```
<configuration>
  <windowsTokenService>
    <!--
        By default no callers are allowed to use the Windows Identity Foundation Claims To NT Token
Service.
        Add the identities you wish to allow below.
      -->
    <allowedCallers>
      <clear/>
      <add value="WSS_WPG" />
    </allowedCallers>
  </windowsTokenService>
</configuration>
```

3. Start the operating system c2WTS service:

a. Configure the service to use the service account you configured in the previous step.

b. Change the Startup type to "**Automatic**" and start the service.

4. Start the SharePoint 'Claims to Windows Token Service': Start the Claims to Windows Token Service through SharePoint Central Administration on the **Manage Services on Server** page. The service should be started on the server that will be performing the action. For example if you have a server that is a WFE and another server that is an Application Server that has the Reporting Services shared service running, you only need to start c2WTS on the Application Server. c2WTS is not needed on the WFE.

See Also

Claims to Windows Token Service (c2WTS) Overview (https://msdn.microsoft.com/library/ee517278.aspx)
Overview of Kerberos authentication for Microsoft SharePoint 2010 Products (https://technet.microsoft.com/library/gg502594.aspx)

PowerPivot for SharePoint 2013 Installation

4/15/2020 • 11 minutes to read • Edit Online

The procedures in this topic guide you through a single server installation of a Analysis Services server in SharePoint deployment mode. The steps include running the SQL Server installation wizard as well as configuration tasks that use SharePoint 2013 Central Administration.

Applies to: SharePoint 2013 | SharePoint 201

In this topic:

Background

Prerequisites

Step 1: Install PowerPivot for SharePoint

Step 2: Configure Basic Analysis Services SharePoint Integration

Step 3: Verify the Integration

Configure the Windows Firewall to Allow Analysis Services Access

Upgrade Workbooks and Scheduled Data Refresh

Beyond the Single-Server Installation -PowerPivot for Microsoft SharePoint

Background

PowerPivot for SharePoint is a collection of middle-tier and backend services that provide PowerPivot data access in a SharePoint 2013 farm.

- **Backend services:** If you use PowerPivot for Excel to create workbooks that contain analytical data, you must have PowerPivot for SharePoint to access that data in a server environment. You can run SQL Server Setup on a computer that has SharePoint Server 2013 installed, or on a different computer that has no SharePoint software. Analysis Services does not have any dependencies on SharePoint.

  Note: This topic describes the installation of the Analysis Services server and the backend services.

- **Middle-tier:** Enhancements to the PowerPivot experiences in SharePoint including PowerPivot Gallery, Schedule data refresh, Management dashboard, and data providers. For more information on installing and configuring the middle-tier, see the following:

  - Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013)
  - Configure PowerPivot and Deploy Solutions (SharePoint 2013)

Prerequisites

1. You must be a local administrator to run SQL Server Setup.
2. SharePoint Server 2013 enterprise edition is required for PowerPivot for SharePoint. You can also use the evaluation enterprise edition.
3. The computer must be joined to a domain in the same Active Directory forest as Excel Services.
4. The PowerPivot instance name must be available. You cannot have an existing PowerPivot-named instance

on the computer on which you are installing Analysis Services in SharePoint mode.

5. Review Hardware and Software Requirements for Analysis Services Server in SharePoint Mode (SQL Server 2014).

6. Review the release notes at SQL Server 2012 Service Pack 1 Release Notes (https://go.microsoft.com/fwlink/?LinkID=248389).

SQL Server Edition Requirements

Business intelligence features are not all available in all editions of SQL Server 2014. For details, see Features Supported by the Editions of SQL Server 2012 (https://go.microsoft.com/fwlink/?linkid=232473) and Editions and Components of SQL Server 2014.

The current release notes can be found at SQL Server 2012 SP1 Release Notes (https://go.microsoft.com/fwlink/?LinkID=248389).

Microsoft SQL Server 2012 Release Notes (https://go.microsoft.com/fwlink/?LinkId=236893).

Step 1: Install PowerPivot for SharePoint

In this step, you run SQL Server Setup to install an Analysis Services server in SharePoint mode. In a subsequent step, you configure Excel Services to use this server for workbook data models.

1. Run the SQL Server Installation Wizard (Setup.exe).

2. Click **Installation** in the left navigation.

3. Click **New SQL Server stand-alone installation or add features to an existing installation**.

4. If you see the **Product Key** page, specify the evaluation edition or enter a product key for a licensed copy of the enterprise edition. Click **Next**. For more information on editions, see Editions and Components of SQL Server 2014.

5. Review and accept the Microsoft Software License Terms of agreement, and then click **Next**.

6. If you see the **Global Rules** page, review any rules information the setup wizard displays.

7. On the **Microsoft Update** page, it is recommended you use Microsoft Update to check for updates, then click **Next**.

8. The **Install Setup Files** page runs for several minutes. Review any rule warnings or failed rules, and then click **Next**.

9. If you see another **Setup Support Rules**, review any warnings and click **Next**.

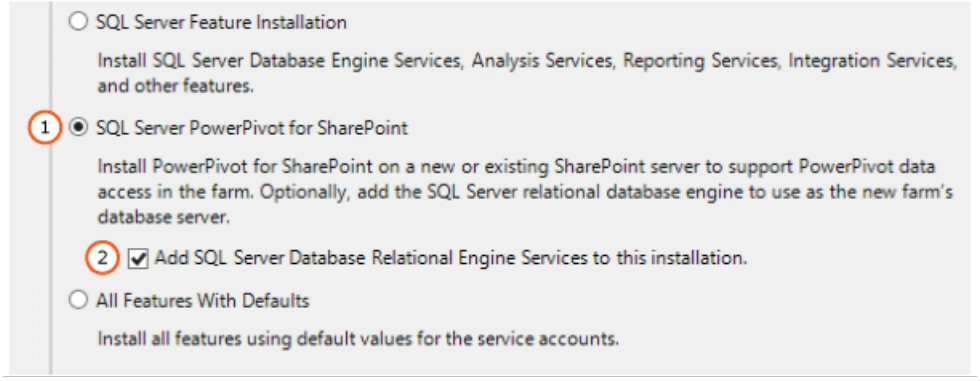
   Note: Because Windows Firewall is enabled, you see a warning to open ports to enable remote access.

10. On the **Setup Role** page, select **SQL Server PowerPivot for SharePoint**. This option installs Analysis Services in SharePoint mode.

    Optionally, you can add an instance of the Database Engine to your installation. You might add the Database Engine when setting up a new farm and need a database server to run the farm's configuration and content databases. This option also installs SQL Server Management Studio.

    If you add the Database Engine, it is installed as a **PowerPivot** named instance. Whenever you specify a connection to this instance, enter the database name in this format: [`servername`]\PowerPivot.

    Click **Next**.



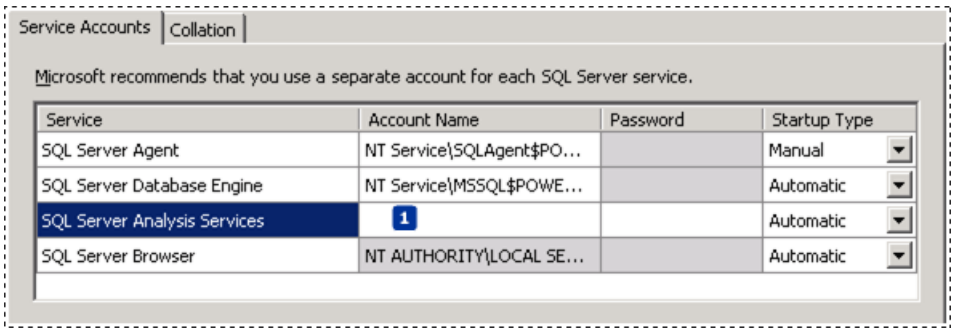
11. In Feature Selection, a read-only list of the features is displayed for informational purposes. You cannot add or remove items the preselected items for this role. Click **Next**.

12. On the **Instance Configuration** page, a read-only instance name of 'PowerPivot' is displayed for informational purposes. This instance name is required and it cannot be modified. However, you can enter a unique Instance ID to specify a descriptive directory name and registry keys. Click **Next**.

13. On the **Server Configuration** page, configure all of the services for Automatic **Startup Type**. Specify the desired domain account and password for **SQL Server Analysis Services**, **(1)** in the following diagram.

    - For Analysis Services, you can use a **domain user** account or **NetworkService** account. Do not use LocalSystem or LocalService accounts.
    - If you added the SQL Server Database Engine and SQL Server Agent, you can configure the services to run under domain user accounts or under the default virtual account.
    - Never provision service accounts with your own domain user account. Doing so grants the server the same permissions that you have to the resources in your network. If a malicious user compromises the server, that user is logged in under your domain credentials. The user has the permissions to download or use the same data and applications that you do.

    Click **Next**.



14. If you are installing the Database Engine, the **Database Engine Configuration** page appears. In Database Engine Configuration, click **Add Current User** to grant your user account administrator permissions on the Database Engine instance.

    Click **Next**.

15. On the **Analysis Services Configuration** page, click **Add Current User** to grant your user account administrative permissions. You will need administrative permission to configure the server after Setup is finished.

    - In the same page, add the Windows user account of any person who also requires administrative permissions. For example, any user who wants to connect to the Analysis Services service instance in SQL Server Management Studio to troubleshoot database connection problems must have system administrator permissions. Add the user account of any person who might need to troubleshoot or

administer the server now.

- **NOTE**

  All service applications that require access to the Analysis Services server instance need to have Analysis Services Administrative permissions. For example, add the service accounts for Excel Services, Power View, and Performance Point Services. Also, add the SharePoint farm account, which is used as the identity of the web application that hosts Central Administration.

Click **Next**.

16. On the **Error Reporting** page, click **Next**.
17. On the **Ready to Install** page, click **Install**.
18. If you see the dialog **Computer Restart Required**, click **OK**.
19. When the installation is complete, click **Close**.
20. Restart the computer.
21. If you have a firewall in your environment, review the SQL Server Books Online topic, Configure the Windows Firewall to Allow Analysis Services Access.

Verify the SQL Server Installation

Verify that the Analysis Services Service is running.

1. In Microsoft Windows click **Start**, click **All Programs**, and click the **Microsoft SQL Server 2012** group.
2. Click **SQL Server Management Studio**.
3. Connect to the Analysis Services instance, for example **[your server name]\POWERPIVOT**. If you can connect to the instance, you have verified the Service is running.

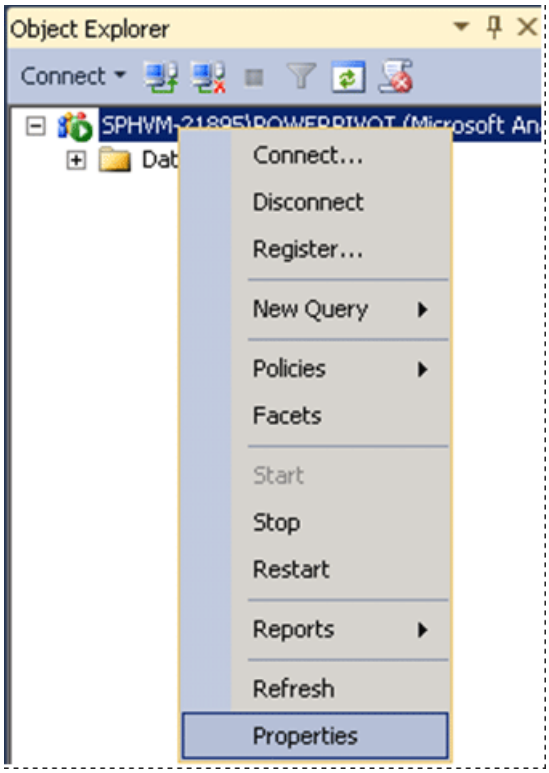
Step 2: Configure Basic Analysis Services SharePoint Integration

The following steps describe configuration changes needed so you can interact with Excel advanced data models inside a SharePoint document library. Complete these steps after you install SharePoint Server 2013 and SQL Server Analysis Services.

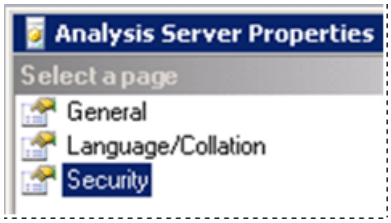
Grant Excel Services Server Administration Rights on Analysis Services

You do not need to complete this section if during the Analysis Services installation; you added the Excel Services Application service account as an Analysis Services administrator.

1. On the Analysis Services server, start SQL Server Management Studio and connect to the Analysis Services instance, for example `[MyServer]\POWERPIVOT`.
2. In Object Explorer, Right-click the instance name and click **Properties**.



3. In the left pane, click **Security**. Add the domain login you configured for the Excel Services Application in step 1.



Configure Excel Services for Analysis Services integration

1. In SharePoint Central Administration, in the Application Management group, click **Manage Service Applications**.
2. Click the name of your service application, the default is **Excel Services Application**.
3. On the **Manage Excel Services Application page**, click **Data Model Settings**.
4. Click **Add Server**.
5. In **Server Name**, type the Analysis Services server name and the PowerPivot instance name. For example `MyServer\POWERPIVOT`. The PowerPivot instance name is required.

   Type a description.
6. Click **Ok**.
7. The changes will take effect in a few minutes or you can **Stop** and **Start** the service **Excel Calculation Services**. To

   Another option is to open a command prompt with administrative privileges, and type `iisreset /noforce`.

   You can verify the server is recognized by Excel Services by reviewing entries in the ULS log. You will see entries similar to the following:

```
Excel Services Application            Data Model        27           Medium                Check
Administrator Access ([ServerName]\POWERPIVOT): Pass.        f127bd9b-bae3-e0e0-9b48-3f7b5ad1eae6
Excel Services Application            Data Model        27           Medium                Check Server
Version ([ServerName]\POWERPIVOT): Pass (11.0.2809.24 >= 11.0.2800.0).         f127bd9b-bae3-e0e0-9b48-
3f7b5ad1eae6
Excel Services Application            Data Model        27           Medium                Check
Deployment Mode ([ServerName]\POWERPIVOT): Pass.            f127bd9b-bae3-e0e0-9b48-3f7b5ad1eae6
```

Step 3: Verify the Integration

The following steps walk you through creating and uploading a new workbook to verify the Analysis Services integration. You will need a SQL Server database to complete the steps.

1. **Note:** If you already have an advanced workbook with slicers or filters, you can upload it to your SharePoint document library and verify you are able to interact with the slicers and filters from the document library view.
2. Start a new workbook in Excel.
3. On the Data tab, click **From Other Sources** on the ribbon in the **Get External Data**.
4. Select **From SQL Server**.
5. In the **Data Connection Wizard**, enter the name of the SQL Server instance that has the database you want to use.
6. or Log on credentials, verify that **Use Windows Authentication** is selected, and then click **Next**.
7. Select the database you want to use.
8. Verify that the **Connect to specific table** checkbox is selected.
9. Click the **Enable selection of multiple tables and add tables to the Excel Data Model** checkbox.
10. Select the tables you want to import.
11. Click the checkbox **Import relationships between selected tables**, and then click **Next**. Importing multiple tables from a relational database lets you work with tables that are already related. You save steps because you don't have to build the relationships manually.
12. In the **Save Data Connection File and Finish** page of the wizard,, type a dame for your connection and click **Finish**.
13. The **Import Data** dialog box will appear. Choose **PivotTable Report**, and then click **Ok**.
14. A PivotTable Field List appears in the workbook.
    On the field list, click the **All** tab
15. Add fields to the Row, Columns, and Value areas in the field list.
16. Add a slicer or a filter to the PivotTable. **Do not skip this step**. A slicer or filter is the element that will help you verify your Analysis Services installation.
17. Save the workbook to a document library on a SharePoint Server 2013 that has Excel Services configured. You can also save the workbook to a file share and then upload it to the SharePoint Server 2013 document library.
18. Click the name of your workbook to view it in SharePoint and click the slicer or change the filter that you previously added. If a data update occurs, you know that Analysis Services is installed and available to Excel

Services. If you open the workbook in Excel you will be using a cached copy and not using the Analysis Services server.

Configure the Windows Firewall to Allow Analysis Services Access

Use the information in the topic Configure the Windows Firewall to Allow Analysis Services Access to determine whether you need to unblock ports in a firewall to allow access to Analysis Services or PowerPivot for SharePoint. You can follow the steps provided in the topic to configure both port and firewall settings. In practice, you should perform these steps together to allow access to your Analysis Services server.

Upgrade Workbooks and Scheduled Data Refresh

The steps required to upgrade workbooks created in previous versions of PowerPivot depend on what version of PowerPivot created the workbook. For more information, see Upgrade Workbooks and Scheduled Data Refresh (SharePoint 2013).

Beyond the Single-Server Installation -PowerPivot for Microsoft SharePoint

Web front-end (WFE) or **Middle-tier**:: To use an Analysis Services server in SharePoint mode in a larger SharePoint farm and to install additional PowerPivot features into the farm, run the installer package **spPowerPivot.msi** on each of the SharePoint servers. The spPowerPivot.msi installs required data providers and the PowerPivot for SharePoint 2013 Configuration tool.

For more information on installing and configuring the middle-tier, see the following:

- Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013)
- To download the .msi, see Microsoft SQL Server 2014 PowerPivot for Microsoft SharePoint 2013
- Configure PowerPivot and Deploy Solutions (SharePoint 2013)

Redundancy and server load: Installing a second, or more Analysis Services servers in SharePoint mode will provide redundancy of the Analysis Services server functionality. Additional servers will also spread the load across servers. For more information, see the following:

- Configure Analysis Services for processing data models in Excel Services (https://technet.microsoft.com/library/jj614437(v=office.15)).
- Manage Excel Services data model settings (SharePoint Server 2013) (https://technet.microsoft.com/library/jj219780(v=office.15)).

Submit feedback and contact information through Microsoft SQL Server Connect (https://connect.microsoft.com/SQLServer/Feedback).

See Also

Migrate PowerPivot to SharePoint 2013
Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013)
Upgrade Workbooks and Scheduled Data Refresh (SharePoint 2013)

Hardware and Software Requirements for Analysis Services Server in SharePoint Mode (SQL Server 2014)

12/16/2019 • 4 minutes to read • Edit Online

PowerPivot for SharePoint supports both SharePoint 2010 and SharePoint 2013. PowerPivot for SharePoint 2013 runs outside the SharePoint farm though it can be installed on SharePoint servers. PowerPivot for SharePoint 2010 runs on application servers in a SharePoint 2010 farm and uses SharePoint features and infrastructure to support server operations. To install PowerPivot for SharePoint for either version of SharePoint use the SQL Server 2014 installation wizard. Following the installation, complete the following:

- Run the appropriate version of the PowerPivot for SharePoint configuration tool for the version of SharePoint.
- For SharePoint 2013, install the Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013).

SQL Server Edition Requirements

Business intelligence features are not all available in all editions of SQL Server 2014. For details, see Features Supported by the Editions of SQL Server 2014 and Editions and Components of SQL Server 2014.

The current release notes can be found at Microsoft SQL Server 2014 Release Notes.

SQL Server licensing

For more information on SQL Server licensing, see the following:

- SQL Server 2014 Licensing Datasheet (https://download.microsoft.com/download/6/6/F/66FF3259-1466-4BBA-A505-2E3DA5B2B1FA/SQL_Server_2014_Licensing_Datasheet.pdf).
- How To Buy: SQL Server licensing models support (https://www.microsoft.com/licensing/product-licensing/sql-server-2014?activetab=sql-server-2014-pivot%3aprimaryr2).

Analysis Services Installed on SharePoint 2013

If you install Analysis Services server in SharePoint mode on a server by itself, then minimum system requirements are based on SQL Server 2014 rather than SharePoint Server requirements.

Hardware and Software Requirements for Installing SQL Server 2014

PowerPivot for SharePoint runs best on new generation business servers that offer higher RAM thresholds and more processing capability. Large amounts of RAM are used to store PowerPivot data in memory. The RAM supports the ability to adapt to structural changes. Extra processors support long running scans of raw, un-aggregated data. The data assumes its structure in a dynamic environment, in response to user-driven data analysis initiated through an Excel client or front-end interface.

> **TIP**
> PowerPivot for SharePoint utilizes L2 and L3 caches. To improve performance, consider using Processors with larger L2 and L3 caches.

The following table describes the minimum and recommended hardware configuration for a standalone PowerPivot for SharePoint 2013 server that is not part of the SharePoint farm:

COMPONENT	MINIMUM	RECOMMENDED
Processor	64-bit dual-core processor, 3 gigahertz.	16 Cores
RAM	8 gigabytes of RAM	64 gigabytes of RAM
Storage	80 gigabytes of storage	80 gigabytes or more

If you install Analysis Services server in SharePoint mode on a SharePoint farm server, review the minimum system requirements for both SQL Server 2014 and SharePoint Server at the following links:

- Hardware and Software Requirements for Installing SQL Server 2014
- Hardware and software requirements for SharePoint 2013.

The standard SharePoint 2013 hardware and software recommendations are for a web-based document management solution for a workgroup or team. Because PowerPivot processing is data intensive, the standard recommendation is sufficient if the overall workload is small, for example, less than 100 users or workbooks. A larger PowerPivot deployment requires more computing power.

Analysis Services Installed on a SharePoint 2010 Server

PowerPivot for SharePoint 2010 runs on application servers in a SharePoint 2010 farm and uses SharePoint features and infrastructure to support server operations. The following table summarizes requirements related to SharePoint 2010 deployments:

COMPONENT	REQUIREMENT
SharePoint version	SharePoint 2010 Enterprise, with Excel Services, Secure Store Service, and the Claims to Windows Token Service configured in the same server farm. SharePoint must be installed using the Server Farm option in SharePoint Setup (SharePoint's Standalone installation option is not supported). PowerPivot requires server farm infrastructure to support administrative and data access. The standalone installation does not provide these services. The developer edition, which runs on Windows 7 or Windows Vista, is not supported for PowerPivot server installations.
Service Packs	SharePoint Server 2010 Service Pack 1 (SP1) is required. SharePoint 2010 Service Pack 1 is required for SQL Server 2012PowerPivot for SharePoint features. SharePoint 2010 August 2010 Cumulative Update, or later is required when Upgrading from a previous version of PowerPivot for SharePoint to SQL Server 2012. The August 2010 Cumulative update or later should be installed after installing SharePoint Service Pack 1. A new installation of SQL Server 2012 PowerPivot for SharePoint does not require the cumulative update. For more information, see August 2010 Cumulative Update for SharePoint has been released.

COMPONENT	REQUIREMENT
SharePoint web application	PowerPivot for SharePoint 2010 only supports SharePoint web applications that are configured for classic-mode authentication. If you are adding PowerPivot for SharePoint to an existing farm, be sure that the web application you plan to use it with is configured for classic-mode authentication. For instructions on how to check authentication mode, see the section "Verify the Web application uses Classic mode authentication" in Deploy PowerPivot Solutions to SharePoint.
Data providers required for PowerPivot server-side data refresh	Server-side data refresh repeats the same data retrieval steps used to originally import the data. This means that the data providers used to import the data on a client workstation must also be present on the PowerPivot for SharePoint server. In addition, using data feeds on a SharePoint server requires that you have ADO.NET Data Services. The SharePoint Prerequisite Installer program does not install this software for you. The following software must be installed manually. ADO.NET Data Services 3.5 SP1 runtime assemblies, used to export a SharePoint list as a data feed. Download and install the version that matches your operating system: For Windows Server 2008 R2, use ADO.NET Data Services Update for .NET Framework 3.5 SP1 for Windows 7 and windows Server 2008 R2. Windows Server 2008 R2 **SP1** already contains the updated provider. For Windows Server 2008, use ADO.NET Data Services Update for .NET Framework 3.5 SP1 for Windows 2000, Windows Server 2003, Windows XP, Windows Vista and Windows Server 2008 (https://go.microsoft.com/fwlink/?LinkId=158125).

Determine Hardware and Software Requirements (SharePoint 2010) (https://go.microsoft.com/fwlink/?LinkId=169734)

Additional Information

Example of a Minimum-Privilege Configuration for PowerPivot for SharePoint 2013

10/22/2019 • 2 minutes to read • Edit Online

This topic describes an example PowerPivot for SharePoint 2013 configuration with minimum privileges. The configuration utilizes a different account for each of the three components and each account has the minimum level of privileges.

Summary of Accounts

PowerPivot for SharePoint 2013 supports the use of the Network Service account for the Analysis Services service account. The Network Service account is not a supported scenario with SharePoint 2010. For more information on Service accounts, see Configure Windows Service Accounts and Permissions (https://msdn.microsoft.com/library/ms143504.aspx).

The following table summarizes the three accounts used in this example of a minimum privileged configuration.

SCOPE	NAME
SharePoint Administrator account	**SPAdmin**
SharePoint Farm account	**SPFarm**
Analysis Services service account	**SPsvc**

The SharePoint Administrator account (SpAdmin)

SPAdmin is a domain account you use to install and configure the farm. It is the account used to run the SharePoint Configuration Wizard and the PowerPivot Configuration Tool for SharePoint 2013.The **SPAdmin** account is the only account that requires local Administrator rights. Before running the PowerPivot Configuration tool, grant the **SPAdmin** account privileges to the SQL Server database instance where SharePoint creates content and configuration databases. To configure the SPAdmin account in a minimum privilege scenario, it should be a member of the roles **securityadmin** and **dbcreator**.

The Farm account (SPFarm)

SPFarm is a domain account that the SharePoint Timer service and the web application for Central Administration use to access the SharePoint content database. This account does not need to be a local administrator. The SharePoint configuration wizard grants the proper minimal privilege in the back-end SQL Server database.The minimum SQL Server privilege configuration is membership in the roles **securityadmin** and **dbcreator**.

The Service Account for PowerPivot Service (SPsvc)

If a new SharePoint farm is not configured before you run the PowerPivot Configuration tool, then by default the PowerPivot Configuration tool will create the following:

- PowerPivot Service application.
- Excel Services application.
- Secure Store application.

The PowerPivot configuration tool configures all three of the service applications in the default application pool.

That application pool is typically configured to run as the SPFarm account, which has access to many resources that a service account does not require.To make the environment a minimum-privileged environment, configure a new domain account to be use by the appropriate application pool and web application.

To create a new domain account SPsvc to be used as a SharePoint Service account:

1. In SharePoint Central Administration, click **Security**.
2. Click **Configure Service Accounts**
3. Click **Register new managed account**.

The **SPSvc** account has no local administrator privileges and SPsvc will not have any privileges in the SharePoint database. The only privileges SPsvc requires is administrative rights to the PowerPivot Instance of the Analysis Services.

To configure the appropriate application pool to use the SPsvc account :

1. In SharePoint Central Administration, click **Security**.
2. Click **Configure Service Accounts**.
3. Select the service application pool used by the PowerPivot Service application. Then select the SPSvc account.

To Grant access to the web application with PowerShell:

1. Run the SharePoint 2013 Management Shell with administrator privileges.
2. Run the following PowerShell code:

```
$webApp = Get-SPWebApplication "http://<servername>"
$webApp.GrantAccessToProcessIdentity("DOMAIN\<ServiceAccountName>")
```

Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013)

4/14/2020 • 7 minutes to read • Edit Online

PowerPivot for SharePoint 2013 is a collection of application server components and back-end services that provide PowerPivot data access in a SharePoint Server 2013 farm. The PowerPivot for SharePoint add-in (**spPowerpivot.msi**) is an installer package used to install the application server components.

- The add-in is not required for SharePoint 2010 deployments.
- The add-in is not required on a single server deployment that includes SharePoint 2013 and Analysis Services in SharePoint mode. The components installed by the add-in are included when you install an Analysis Services server in SharePoint mode. For diagrams of example deployments with the add-in, see Deployment Topologies for SQL Server BI Features in SharePoint.

Note: This topic describes installing the PowerPivot solution files and PowerPivot for SharePoint 2013 Configuration tool. After the installation, see the following topic for information on the configuration tool and additional features, Configure PowerPivot and Deploy Solutions (SharePoint 2013).

For information on how to download **spPowerPivot.msi**, see Microsoft® SQL Server® 2014 PowerPivot® for Microsoft SharePoint®.

In this topic:

- Background
- Where to Install spPowerPivot.msi?
- Requirements and Prerequisites
- To Install PowerPivot for SharePoint
- Deploy the SharePoint Solution Files with the PowerPivot for SharePoint 2013 Configuration Tool
- Uninstall or repair the add-in

Background

- **Application Server:** PowerPivot functionality in SharePoint 2013 includes using workbooks as a data source, scheduled data refresh, and the PowerPivot Management Dashboard.

  PowerPivot for SharePoint 2013 is a Microsoft Windows Installer package (**spPowerpivot.msi**) that deploys Analysis Services client libraries and copies PowerPivot for SharePoint 2013 installation files to the computer. The installer does not deploy or configure PowerPivot features in SharePoint. The following components install by default:

  - PowerPivot for SharePoint 2013. This component includes PowerShell scripts (.ps1 files), SharePoint solution packages (.wsp), and the PowerPivot for SharePoint 2013 configuration tool to deploy PowerPivot in a SharePoint 2013 farm.
  - Microsoft OLE DB Provider for Analysis Services (MSOLAP).
  - ADOMD.NET data provider.
  - SQL Server Analysis Management Objects.

- **Backend services:** If you use PowerPivot for Excel to create workbooks that contain analytical data, you must have Excel Services configured with a BI server running Analysis Services in SharePoint mode to access that data in a server environment. You can run SQL Server Setup on a computer that has SharePoint Server 2013 installed, or on a different computer that has no SharePoint software. Analysis Services does not have any dependencies on SharePoint.

  For more information on installing, uninstalling, and configuring the backend services, see the following:

  - PowerPivot for SharePoint 2013 Installation
  - Uninstall PowerPivot for SharePoint

Where to Install spPowerPivot.msi?

A recommended best practice is to install **spPowerPivot.msi** on all servers in the SharePoint farm for configuration consistency, including application servers and web-front end servers. The installer package includes the Analysis Services data providers as well as the PowerPivot for SharePoint 2013 configuration tool. When you install **spPowerPivot.msi** you can customize the installation by excluding individual components.

Data providers: Several SharePoint and SQL Server technologies use the Analysis Services data providers including Excel Services, PerformancePoint Services, and Power View. Installing **spPowerPivot.msi** on all SharePoint servers ensures the full set of Analysis Services data providers and PowerPivot connectivity is consistently available across the farm.

> **NOTE**
>
> You must install the Analysis Services data providers on a SharePoint 2013 server using **spPowerPivot.msi**. Other installer packages available in the SQL Server 2014 Feature Pack are not supported because these packages do not include the SharePoint 2013 support files that the data providers require in this environment.

Configuration Tool: The PowerPivot for SharePoint 2013 configuration tool is required on only one of the SharePoint servers. However a recommended best practice in multi-server farms is to install the configuration tool on at least two servers so you have access to the configuration tool if one of the two servers is offline.

Requirements and Prerequisites

- Microsoft SharePoint Server 2013.
- **spPowerPivot.msi** is 64-bit only, in accordance with the requirements of SharePoint products and technologies.
- A SQL Server 2014 Analysis Services (SSAS) server in PowerPivot mode. Excel Services will use the SQL Server Analysis Services instance as a PowerPivot server. Analysis Services can run on the local or a remote computer.
- **Permissions:** To install PowerPivot for SharePoint 2013, the current user is required to be an administrator on the computer and a SharePoint Farm Administrators group.
- For more information on PowerPivot for SharePoint requirements and pre-requisites, go to Hardware and Software Requirements for Analysis Services Server in SharePoint Mode (SQL Server 2014).

To Install PowerPivot for SharePoint

The **spPowerpivot.msi** installer package supports both a graphical user interface and a command-line mode. Both methods of installation require that you run the .msi with administrator privileges. After the installation, see the following topic for information on the configuration tool and additional features, Configure PowerPivot and Deploy

Solutions (SharePoint 2013).

User interface installation

To install PowerPivot for SharePoint 2013 with the graphical user interface, complete the following steps:

1. Run **SpPowerPivot.msi**.
2. Click **Next** on the Welcome page.
3. Review and accept the license agreement, then click **Next**.
4. On the **Feature Selection** page, all of the features are selected by default.
5. Click **Next**.
6. Click **Install** to install to finish the installation.

Command Line Installation

For a command-line installation, open a command prompt with administrative permissions, and then run the **spPowerPivot.msi**. For example:

`Msiexec.exe /i SpPowerPivot.msi` .

To create an installation log, use the standard MsiExec logging switches. The following example creates the log file "Install_Log.txt" using the "v" verbose logging switch.

```
Msiexec.exe /i SpPowerPivot.msi /L v c:\test\Install_Log.txt
```

Quiet Command Line Installation for scripting

You can use the **/q** or **/quiet** switches for a "quiet" installation that will not display any dialogs or warnings. The quiet installation is useful if you want to script the installation of the add-in.

> **IMPORTANT**
> If you use the **/q** switch for a silent command line installation, the end-user license agreement will not be displayed. Regardless of the installation method, the use of this software is governed by a license agreement and you are responsible for complying with the license agreement.

To perform a quiet installation

1. Open a command prompt **with administrator permissions**.
2. Run the following command:

```
Msiexec.exe /i spPowerPivot.msi /q
```

Command Line Installation to include specific components

The PowerPivot for SharePoint 2013 Configuration tool is not required on every SharePoint server, however it is recommended to install it on at least two servers so the configuration tool is available when you need it.

When you install the spPowerPivot.msi, you can use the command line options to install specific items, such as the data providers and not the PowerPivot for SharePoint 2013 Configuration tool. The following command line is an example of installing all components except the configuration tool:

```
Msiexec /i spPowerPivot.msi AGREETOLICENSE="yes" ADDLOCAL=" SQL_OLAPDM,SQL_ADOMD,SQL_AMO,SQLAS_SP_Common"
```

OPTION	DESCRIPTION
Analysis_Server_SP_addin	PowerPivot Configuration
SQL_OLAPDM	MSOLAP
SQL_ADOMD	ADOMD.net provider
SQL_AMO	AMO provider
SQLAS_SP_Common	Analysis Services common components for SharePoint 2013

Deploy the SharePoint Solution Files with the PowerPivot for SharePoint 2013 Configuration Tool

Three of the files copied to the hard drive by spPowerPivot.msi are SharePoint solution files. The scope of one solution file is the Web application level while the scope of the other files is the farm level. The files are the following:

- `PowerPivotFarmSolution.wsp`
- `PowerPivotFarm14Solution.wsp`
- `PowerPivotWebApplicationSolution.wsp`

The solution files are copied to the following folder:

```
C:\Program Files\Microsoft SQL Server\120\Tools\PowerPivotTools\SPAddinConfiguration\Resources
```

Following the .msi installation, run the PowerPivot for SharePoint 2013 Configuration Tool to configure and deploy the solutions in the SharePoint farm.

To start the configuration tool

From the Windows Start screen type "power" and in the Apps search results, click **PowerPivot for SharePoint 2013 Configuration**. Note that the search results may include two links because SQL Server setup installs separate PowerPivot configuration tools for SharePoint 2010 and SharePoint 2013. Make sure you start the PowerPivot for SharePoint 2013 Configuration tool.

Or

1. Go to **Start**, **All Programs**.
2. Click **Microsoft SQL Server 2014**.
3. Click **Configuration Tools**.
4. Click **PowerPivot for SharePoint 2013 Configuration**.

For more information on the configuration tool, see PowerPivot Configuration Tools.

Uninstall or repair the add-in

Caution

If you uninstall **spPowerPivot.msi** the data providers and the configuration tool are uninstalled. Uninstalling the data providers will cause the server to be unable to connect to PowerPivot.

You can uninstall or repair PowerPivot for SharePoint 2013 using one of the following methods:

1. **Windows control panel:** Select **Microsoft SQL Server 2012 PowerPivot for SharePoint 2013**. Click either **Uninstall** or **Repair**.
2. Run the spPowerPivot.msi and select the **Remove** option or the **Repair** option.

Command Line: To repair or uninstall PowerPivot for SharePoint 2013 using the command line, open a command prompt **with administrator permissions** and run one of the following commands:

- To Repair, run the following command:

```
msiexec.exe /f spPowerPivot.msi
```

OR

- To uninstall, run the following command:

```
msiexec.exe /uninstall spPowerPivot.msi
```

Configure PowerPivot and Deploy Solutions (SharePoint 2013)

8/15/2019 • 4 minutes to read • Edit Online

This topics describes the deployment and configuration of middle-tier enhancements to the PowerPivot features in SharePoint Server 2013 including PowerPivot Gallery, Schedule data refresh, Management Dashboard, and data providers. Run **PowerPivot for SharePoint 2013 Configuration** tool to complete the following:

- Deploy SharePoint solution files.
- Create a PowerPivot service application.
- Configure an Excel Services Application to use an Analysis Services server in SharePoint mode. For information on backend services and installing a Analysis Services server in SharePoint mode, see PowerPivot for SharePoint 2013 Installation.

For information on installing the PowerPivot for SharePoint 2013 Configuration tool, see Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013)

This topic contains the following sections:

Run PowerPivot for SharePoint 2013 configuration

Verify PowerPivot Configuration

Troubleshoot Issues

Run PowerPivot for SharePoint 2013 configuration

Note: The SQL Server 2014 setup wizard installs two different configuration tools for Microsoft SQL Server 2014 PowerPivot for SharePoint. They each support a different version of SharePoint.

NAME	DESCRIPTION
PowerPivot for SharePoint 2013 Configuration	SharePoint 2013
PowerPivot Configuration Tool	SharePoint 2010 with SharePoint 2010 Service Pack 1 (SP1)

Note: To complete the following steps, you must be a farm administrator. If you see an error message similar to the following:

- "The user is not a farm administrator. Please address the validation failures and try again."

Either login as the account that installed SharePoint or configure the setup account as the primary administrator of the SharePoint Central Administration Site.

1. On the **Start** menu, click **All Programs**, and then click **Microsoft SQL Server 2014**, click **Configuration Tools**, and then click **PowerPivot For SharePoint 2013 Configuration**. Toold is listed only when PowerPivot for SharePoint is installed on the local server.
2. Click **Configure or Repair PowerPivot for SharePoint** and then click **OK**.
3. The tool runs validation to verify the current state of PowerPivot and what steps are required to complete configuration. Expand the window to full size. You should see a button bar at the bottom of the window that

includes **Validate**, **Run**, and **Exit** commands.

4. On the **Parameters** tab:

   a. **Default Account UserName**: Enter a domain user account for the default account. This account will be used to provision services, including the PowerPivot service application pool. Do not specify a built-in account such as Network Service or Local System. The tool blocks configurations that specify built-in accounts.

   b. **Database Server**: You can use SQL Server Database engine that is supported for the SharePoint farm.

   c. **Passphrase**: Enter a passphrase. If you are creating a new SharePoint farm, the passphrase is used whenever you add a server or application to the SharePoint farm. If the farm already exists, enter the passphrase that allows you to add a server application to the farm.

   d. **PowerPivot Server for Excel Services**: Type the name of an Analysis Services SharePoint mode server. In a single-server deployment, it is the same as the database server. `[ServerName]\powerpivot`

   e. Click **Create Site Collection** in the left window. Note **Site URL** so you can reference it in later steps. If the SharePoint server is not already configured, then the configuration wizard defaults the web application, and site collection URLs to the root of `http://[ServerName]`. To modify the defaults review the following pages in the left window: **Create Default Web application** and **Deploy Web Application Solution**

5. Optionally, review the remaining input values used to complete each action. Click each action in the left window to see and review the details of the action. For more information about each one, see the section "Input values used to configure the server in Configure or Repair PowerPivot for SharePoint 2010 (PowerPivot Configuration Tool) in this topic.

6. Optionally, remove any actions that you do not want to process at this time. For example, if you want to configure Secure Store Service later, click **Configure Secure Store Service**, and then clear the checkbox **Include this action in the task list**.

7. Click **Validate** to check whether the tool has sufficient information to process the actions in the list. If you see validation errors, click the warnings in the left pane to see details of the validation error. Correct any validation errors and then click **Validate** again.

8. Click **Run** to process all of the actions in the task list. Note that **Run** becomes available after you validate the actions. If **Run** is not enabled, click **Validate** first.

For more information, see Configure or Repair PowerPivot for SharePoint 2010 (PowerPivot Configuration Tool)

Verify PowerPivot Configuration

Services:

1. In Central Administration, in System Settings, click **Manage services on server**.

2. Verify that **SQL Server Analysis Services** and **SQL Server PowerPivot System Service** are started.

Farm Feature:

1. In Central Administration, in System Settings, click **Manage farm features**.

2. Verify that **PowerPivot Integration Feature** is **Active**.

Site Collection Feature:

1. Browse to your site URL that was created by the Configuration tool.

Click **Settings**, and then click **Site Settings**.

Click **Site Collection Features**.

2. Verify that **PowerPivot Feature Integration for Site Collections** is **Active**.

PowerPivot Service Application:

1. In Central Administration, in the **Application Management**, click **Manage service applications**.
2. Verify the service application status is **started**. The default name is **Default PowerPivot Service Application**.

   Click the name of the services application to open the PowerPivot Management Dashboard for the service application opens. On first use, the dashboard takes several minutes to load.

For more information, see Verify a PowerPivot for SharePoint Installation.

Troubleshoot Issues

To assist in troubleshooting issues, it is a good idea to verify the diagnostic logging is enabled.

1. In SharePoint Central Administration, click **Monitoring** and then click **Configure usage and health data collection**.
2. Verify **Enable usage data collection** is selected.
3. Verify the following events are selected:
   - Definition of usage fields for Education telemetry
   - PowerPivot Connects
   - PowerPivot Load Data Usage
   - PowerPivot Query Usage
   - PowerPivot Unload Data Usage
4. Verify **Enable health data collection** is selected.
5. Click **OK**.

For more information on trouble shooting data refresh, see Troubleshooting PowerPivot Data Refresh (https://social.technet.microsoft.com/wiki/contents/articles/3870.troubleshooting-powerpivot-data-refresh.aspx).

For more information on the configuration tool, see PowerPivot Configuration Tools.

Migrate PowerPivot to SharePoint 2013

2/28/2020 • 7 minutes to read • Edit Online

SharePoint 2013 does not support in-place upgrade. However the procedure of **database-attach upgrade is supported**. The behavior is different from upgrading to SharePoint 2010, where a customer could choose between the two basic upgrade approaches, in-place upgrade and database-attach upgrade.

If you have a PowerPivot for SharePoint installation integrated with SharePoint 2010, you cannot upgrade in-place the SharePoint server. However you can migrate content databases and service application databases from the SharePoint 2010 farm to a SharePoint 2013 farm. This topic is an overview of the steps required to complete a database-attach upgrade and complete a migration related to PowerPivot:

Applies to: SharePoint 2013

Migration Overview

1	2	3	4
Prepare the SharePoint 2013 farm	Backup, copy, restore databases.	Mount content databases	Migrate PowerPivot Schedules
	SQL Server Management Studio	SharePoint Central Administration Windows PowerShell	SharePoint application Pages Windows PowerShell

In this topic:

- 1) Prepare the SharePoint 2013 Farm
- 2) Backup, Copy, Restore the Databases
- 3) Prepare Web Applications and Mount Content Databases
- 4) Upgrade PowerPivot Schedules
- Additional Resources

1) Prepare the SharePoint 2013 Farm

1. **TIP**

   Review the authentication method your existing web applications are configured for. SharePoint 2013 web applications default to claims-based authentication. SharePoint 2010 web applications configured for classic-mode authentication require additional steps to migrate databases from SharePoint 2010 to SharePoint 2013. If your web applications are configured for classic-mode authentication, review the SharePoint 2013 documentation.

2. Install a new SharePoint Server 2013 farm.
3. Install an instance of a SQL Server 2014 Analysis Services server in SharePoint mode. For more information, see PowerPivot for SharePoint 2013 Installation.
4. Run the PowerPivot for SharePoint 2013 installation package **spPowerPivot.msi** on each server in the SharePoint farm. For more information, see Install or Uninstall the PowerPivot for SharePoint Add-in

(SharePoint 2013).

5. In SharePoint 2013 Central Administration, configure the Excel Services service application to use the Analysis Services SharePoint mode server created in the previous step. For more information, see the "Configure Basic Analysis Services SharePoint Integration" section of PowerPivot for SharePoint 2013 Installation.

2) Backup, Copy, Restore the Databases

The "SharePoint database-attach upgrade" process is a sequence of steps to back up, copy, and restore PowerPivot related content and service application databases to the SharePoint 2013 farm.

1. **Set Database to read-only:** In SQL Server Management Studio, right-click the database name and click **Properties**. On the **Options** page, set the **Database read-Only** property to **True**.
2. **Back up:** Back up each content database and service application database that you want to migrate to the SharePoint 2013 farm. In SQL Server Management Studio, right-click the database name, click **Tasks**, and click **Back up**.
3. File copy the database backup files (.bak) to the desired destination server.
4. **Restore:** Restore the databases to the destination SQL Server Database Engine. This step can be completed using SQL Server Management Studio.
5. **Set Database to read-write:** Set the **Database read-Only** to **False**.

3) Prepare Web Applications and Mount Content Databases

For a more detailed explanation of the following procedures, see Upgrade databases from SharePoint 2010 to SharePoint 2013 (https://go.microsoft.com/fwlink/p/?LinkId=256690).

1. **Take Databases Offline:**

   Take each SharePoint 2013 content database offline, using SharePoint Central Administration. The content databases are replaced by the databases you copied over. Consider what is the best sequence for your environment. Consider taking each database offline and mount its relevant replacement database before taking the next content database offline. Another option is to take all the content databases offline as a group.

   a. In SharePoint Central Administration, click **Application Management**.

   b. Click **Manage Content Databases**.

   c. Click the name of the database.

   d. On the **Manage Content Database Settings**, set **Database status** to **Offline**.

   e. Select **Remove Content Database**. Note the warning that sites stored in the content database will no longer be accessible.

- **Mount content databases:**

  Use PowerShell cmdlets in the SharePoint 2013 Management shell to mount the migrated content database.

  The service application database does not need to be mounted, only the content databases:

```
Mount-SPContentDatabase "SharePoint_Content_014-KJSP1" -DatabaseServer "[server name]\powerpivot" -
WebApplication [web application URL]
```

For more information, see Attach or detach content databases (SharePoint Server 2010) (https://technet.microsoft.com/library/ff628582.aspx).

Status when the step is complete: When the mount operation is complete, users can see files that were in the old content database. Therefore users can see and open the workbooks in the document library.

> **TIP**
> It is possible at this point in the migration process to create new schedules for the migrated workbooks. However, the schedules are created in the new PowerPivot service application database, and not the database you copied from the old SharePoint farm. Therefore it will not contain any of the old schedules. After you complete the following steps to use the old database and migrate the old schedules, the new schedules are not available.

Troubleshoot issues when you attempt to mount databases

This section summarizes possible issues encountered when mounting the database.

1. **Authentication errors:** If you see errors related to authentication, review what authentication mode the source web applications are using. The error could be caused by a mismatch in authentication between the SharePoint 2013 web application and the SharePoint 2010 web application. See the 1) Prepare the SharePoint 2013 Farm for more information.
2. **Missing PowerPivot.Files:** If you see errors related to missing PowerPivot .dlls, the **spPowerPivot.msi** has not been installed or the PowerPivot Configuration Tool has not been used to configure PowerPivot.

4) Upgrade PowerPivot Schedules

This section describes the details and options for migrating PowerPivot schedules. Schedule migration is a two-step process. First configure the PowerPivot service application to use the migrated service application database. Second, choose one of two options for schedule migration.

Configure the service application to use the migrated service application database.

In SharePoint Central Administration to configure the PowerPivot services application to use the old service application database you copied over. The PowerPivot Service upgrades the service application database to the new schema.

1. In SharePoint Central Administration, click **Manage Service Applications**.
2. Find the PowerPivot service application, for example "Default PowerPivot Service Application", click the name of the service application and click **Properties** in the SharePoint ribbon.
3. Update the database server name-instance and the database name. To the correct names for the database you backed up, copied, and restored. Once you click **Ok**, the service application database is upgraded. Errors will be in the ULS log.

Upgrade PowerPivot schedules

Configure the PowerPivot service application to migrate refresh schedules.

- **Migrate Schedules option1: SharePoint farm administrator**
  1. In the SharePoint 2013 Management Run the `Set-PowerPivotServiceApplication` cmdlet with the `-StartMigratingRefreshSchedules` switch to enable automatic on demand schedule migration . The following Windows PowerShell script assumes that there is only one PowerPivot service application.

```
$app = Get-PowerPivotServiceApplication
Set-PowerPivotServiceApplication $app -StartMigratingRefreshSchedules
```

After the Windows PowerShell script is run, the schedules are active and the schedules will run at the next appropriate time. However, the status one the schedule refresh page is not enabled. When the schedule runs the first time it will be migrated and on the schedule refresh page, **Enabled** will be true.

2. If you want to check the current value of the StartMigratingRefreshSchedules property, run the following PowerShell script. The Script loops through all PowerPivot service application objects and display the name and property values:

```
$apps = Get-PowerPivotServiceApplication
foreach ($app in $apps){ Get-PowerPivotServiceApplication $app | Format-Table -Property
displayname, id, StartMigratingRefreshSchedules }
```

Migrate Schedules option2: User updates each workbook

1. Another option to migrate schedules is to enable the schedule refresh for each workbook. Navigate to the document library that contains the workbooks.
2. Open the context menu and click **Manage PowerPivot Data Refresh**.
3. In the **schedule refresh** section, click **Enable**.
4. You can select **Also refresh as soon as possible**. This option adds one instance of the refresh to the queue as soon as you click ok. The regular refresh schedule still triggers at the appropriate time.
5. Click **Ok**. The refresh history is now visible in the refresh page, the schedule will fire at the normal time.

SQL Server 2008 R2 PowerPivot workbooks

- SQL Server 2008 R2 PowerPivot workbooks do not automatically upgrade when they are used in SQL Server 2012 SP1 PowerPivot for SharePoint 2013. After you migrate a content database containing the 2008 R2 workbooks, you can use the workbooks but the schedules do not upgrade.
- For more information, see Upgrade Workbooks and Scheduled Data Refresh (SharePoint 2013).

Additional Resources

> **NOTE**
> For more information on PowerPivot and SharePoint database-attach upgrade, see the following:

- Upgrade Workbooks and Scheduled Data Refresh (SharePoint 2013).
- Overview of the upgrade process to SharePoint 2013 (https://go.microsoft.com/fwlink/p/?LinkId=256688).
- Clean up preparations before an upgrade to SharePoint 2013 (https://go.microsoft.com/fwlink/p/?LinkId=256689).
- Upgrade databases from SharePoint 2010 to SharePoint 2013 (https://go.microsoft.com/fwlink/p/?LinkId=256690).

Upgrade Workbooks and Scheduled Data Refresh (SharePoint 2013)

5/22/2019 • 8 minutes to read • Edit Online

This topic explains the user experience of workbooks created in previous PowerPivot environments and how to upgrade PowerPivot workbooks so that you can take advantage of new features introduced in this release. To learn more about new features, see What's New in PowerPivot.

WARNING

You cannot rollback upgrade for workbooks that are upgraded automatically on the server. Once a workbook is upgraded, it remains upgraded. To use a previous version, you can republish the previous workbook to SharePoint, restore a previous version, or recycle the workbook. For more information about restoring or recycling a document in SharePoint, see Plan to protect content by using recycle bins and versioning.

This topic contains the following sections:

- Overview of Upgrading Workbooks
- Upgrade to SQL Server 2012 Service Pack 1 (SP1) workbooks from 2008 R2 Workbooks
- Upgrade to Office 2013 workbooks from Versions created by using the 2012 PowerPivot Add-In for Excel
- Upgrade to SQL Server 2012 workbooks from Versions created by using the 2008 R2 PowerPivot Add-In for Excel 2010
- Running Multiple Workbook Versions on a Newer Server

Overview of Upgrading Workbooks

A PowerPivot workbook is an Excel workbook that contains embedded PowerPivot data. Upgrading a workbook has two benefits:

- Use new features in PowerPivot for Excel.
- Enables scheduled data refresh for workbooks that run with a SQL Server 2012 SP1 Analysis Services server in SharePoint mode.

IMPORTANT

You cannot rollback an upgraded workbook, so be sure to make a copy of the file if you want to use it in the previous version of PowerPivot for Excel, or on a previous version of PowerPivot for SharePoint.

The following table lists the support and behavior of PowerPivot workbooks based on the environment in which the workbook was created. The behavior described includes the general user experience, the supported upgrade options to upgrade the workbook to the particular environment, and the behavior of scheduled data refresh of a workbook that has not yet been upgraded.

Workbook Behavior and Upgrade Options

CREATED IN	<	SUPPORT AND BEHAVIOR	>
	2008 R2 PowerPivot for SharePoint 2010	**2012 PowerPivot for SharePoint 2010**	**2012 SP1 PowerPivot for SharePoint 2013**
2008 R2 PowerPivot for Excel 2010	All Features	**Experience:** Users can interact with the workbook in the browser and use it as a data source for other solutions. **Upgrade:** Workbooks will auto upgrade in the document library if Auto Upgrade is enabled for the PowerPivot system Service in the SharePoint farm, **Schedule data refresh:** NOT supported. Workbook needs to be upgraded.	**Experience:** Users can interact with the workbook and use it as a data source for other solutions. **Upgrade:** Auto upgrade is not available. Users must manually upgrade their 2008 R2 workbooks to the 2012 version or to the office 2013 version. **Schedule data refresh:** NOT supported. Workbook needs to be upgraded.
2012 PowerPivot for Excel	Not supported	All Features	**Experience:** Users can interact with the workbook in the browser and use it as a data source for other solutions. Schedule data refresh is available. **Upgrade:** Auto upgrade is not supported. Users can manually upgrade their workbooks to the Office 2013 version. **Schedule data refresh:** supported.
Excel 2013	Not supported	Not supported	All Features

Upgrade to SQL Server 2012 Service Pack 1 (SP1) workbooks from 2008 R2 Workbooks

This section describes upgrading to SQL Server 2012 SP1 PowerPivot for Excel 2013 workbooks from SQL Server 2008 R2 PowerPivot for Excel 2010 workbooks.

Behavior Change: SQL Server 2008 R2 PowerPivot workbooks will not be automatically upgraded when they are used in SQL Server 2012 SP1 PowerPivot for SharePoint 2013. Therefore, scheduled data refreshes will not work for SQL Server 2008 R2 PowerPivot workbooks

2008 R2 workbooks will open in PowerPivot for SharePoint 2013, however scheduled data refreshes will not work. If you review the refresh history you will see an error message similar to the following:

"The workbook contains an unsupported PowerPivot model. The PowerPivot model in the workbook is in the SQL Server 2008 R2 PowerPivot for Excel 2010 format. Supported PowerPivot models are the following:

- SQL Server 2012 PowerPivot for Excel 2010.
- SQL Server 2012 PowerPivot for Excel 2013.

How to upgrade a workbook: The Scheduled data refresh will not work until you upgrade the workbook to a 2012 workbook. To upgrade the workbook and model it contains, complete one of the following:

- Download and open the workbook in Microsoft Excel 2010 with the SQL Server 2012 PowerPivot for Excel add-in installed.

  Open the PowerPivot window and upgrade the PowerPivot model.

  Then save the workbook and republish it to SharePoint.

- Download and open the workbook in Microsoft Excel 2013.

  Open the PowerPivot window and upgrade the PowerPivot model.

  Then save the workbook and republish it to the SharePoint server.

For more information on Changes to Analysis Services features, see Behavior Changes to Analysis Services Features in SQL Server 2014

For more information on refresh history, see View Data Refresh History (PowerPivot for SharePoint).

Upgrade to Office 2013 workbooks from Versions created by using the 2012 PowerPivot Add-In for Excel

This section describes Upgrading **to** SQL Server 2012 SP1 PowerPivot in Excel 2013 **from** SQL Server 2012 PowerPivot for Excel 2010 workbooks.

Upgrading a workbook resolves the following error that occurs when attempting scheduled data refresh on the previous workbook version workbook:

"Refresh operation for workbooks created with earlier version of PowerPivot is not available."

How to upgrade a workbook

1. Upgrade each workbook manually by opening it in Microsoft Excel 2013.
2. To upgrade the workbook and model it contains, download and open the workbook in Microsoft Excel 2013.
3. Open the PowerPivot window and upgrade the PowerPivot model.
4. Then save the workbook and republish it to the SharePoint 2013 server.

Upgrade to SQL Server 2012 workbooks from Versions created by using the 2008 R2 PowerPivot Add-In for Excel 2010

This section describes Upgrading **to** SQL Server 2012 PowerPivot for Excel 2010 **from** SQL Server 2008 R2 PowerPivot for Excel 2010 workbooks.

Upgrading a workbook resolves the following error that occurs when attempting scheduled data refresh on the previous workbook version workbook:

"Refresh operation for workbooks created with earlier version of PowerPivot is not available."

How to upgrade a workbook

There are two ways to upgrade:

1. Upgrade each workbook manually by opening it in Excel on a computer that has the SQL Server 2012 version of PowerPivot for Excel, and then republish it to the server. When you open the workbook in the newer version of the add-in, the following internal operations occur: the data provider in the workbook data

connection string is updated to MSOLAP.5, metadata is updated, and relationships are recreated to conform to a newer implementation.

2. Alternatively, a SharePoint Administrator can enable the auto-upgrade feature for the PowerPivot System Service in a SharePoint farm to automatically upgrade a SQL Server 2008 R2 PowerPivot workbook when schedule data refresh runs (only workbooks that are configured for scheduled data refresh are upgraded).

> **NOTE**
> Automatic upgrade is a server configuration feature; you cannot enable or disable it for specific workbooks, libraries, or site collections.

How to configure automatic upgrade during data refresh

To use automatic upgrade, you must select the **Automatically upgrade PowerPivot workbooks to enable data refresh from the server** checkbox in the PowerPivot Configuration Tool. Within the tool, the checkbox is on the **Upgrade PowerPivot System Service** page, and on the **Create PowerPivot Service Application** page if you are configuring a new installation.

You can run the following cmdlet to verify whether automatic upgrade is enabled:

```
PS C:\Windows\system32> Get-PowerPivotSystemService
```

The output from Get-PowerPivotSystemService is a list of properties and corresponding values. You should see `WorkbookUpgradeOnDataRefresh` in the property list. It will be set to **true** if automatic upgrade is enabled. If it is **false**, continue to the next step, enabling automatic workbook upgrade.

To enable automatic workbook upgrade, run the following command:

```
PS C:\Windows\system32> Set-PowerPivotSystemService -WorkbookUpgradeOnDataRefresh:$true -Confirm:$false
```

After you upgrade the workbook, you can use scheduled data refresh and new features in the PowerPivot for Excel add-in.

Running Multiple Workbook Versions on a Newer Server

You can run older and newer versions of PowerPivot workbooks side by side on a SQL Server 2012 SP1 instance of PowerPivot for SharePoint.

Depending on how you installed the server, **you might need** to install a previous version of the Analysis Services OLE DB provider before you can access older and newer workbooks on the same server.

Note that Publishing newer version workbooks on previous SQL Server instances of PowerPivot for SharePoint is not supported. A SQL Server 2008 R2 instance will not load a workbook that you created in the SQL Server 2012 version of PowerPivot for Excel, and a SQL Server 2012 instance will not load Office 2013 workbooks with advanced data models that you created using the SQL Server 2012 SP1 version of PowerPivot in Excel.

How to Check for MSOLAP Data Provider Information in a PowerPivot Workbook

Use the following instructions to check which OLE DB provider is used in a PowerPivot workbook. Checking the data connection information does not require the PowerPivot for Excel add-in to be installed.

1. In Excel, on the Data tab, click **Connections**. Click **Properties**.

2. On the **Definition** tab, the provider version appears at the beginning of the connection string.

Provider=MSOLAP.5 indicates the workbook is SQL Server 2012.

Provider=MSOLAP.4 indicates SQL Server 2008 R2.

Data Source=$Embedded$ indicates that the workbook is a PowerPivot workbook, using an embedded database.

How to Check for the Current Version of the MSOLAP Data Provider on a Local Computer

Use the following instructions to check which OLE DB provider is the current version on the server or workstation that runs PowerPivot workbooks. Knowing the current version can help you troubleshoot data connection errors after upgrading.

1. In the Registry Editor, go to HKEY_CLASSES_ROOT
2. Scroll down to MSOLAP. Verify that MSOLAP.5 is listed among the OLAP providers installed on the system. Verify that MSOLAP | CurVer is set to MSOLAP.5

See Also

Migrate PowerPivot to SharePoint 2013
Upgrade PowerPivot for SharePoint
What's New in Analysis Services and Business Intelligence
View Data Refresh History (PowerPivot for SharePoint)

PowerPivot for SharePoint 2010 Installation

10/3/2019 • 2 minutes to read • Edit Online

PowerPivot for SharePoint is a collection of server components that provide query processing and management control over PowerPivot workbooks that you publish to SharePoint. Services include the Analysis Services engine and PowerPivot System Service.

> **NOTE**
>
> For information regarding SQL Server 2012 SP1 and installation with SharePoint Server 2013, see the following:
>
> - The "SQL Server 2012 SP1" section of Overview of SQL Server Servicing Installation.

Analysis Services provides server-side processing for Excel workbooks that contain PowerPivot data. PowerPivot System Service works alongside Analysis Services, adding SharePoint integration, load balancing and connection management. PowerPivot for SharePoint extends Excel Services by pairing its large scale data processing capability with the data rendering services that Excel provides.

To install PowerPivot for SharePoint, use the SQL Server 2012installation media.

For instructions on advanced deployment scenarios, see Deployment Checklist: Reporting Services, Power View, and PowerPivot for SharePoint and Deployment Checklist: Scale-out by adding PowerPivot Servers to a SharePoint 2010 farm.

In This Section

Install PowerPivot for SharePoint 2010

Install the Analysis Services OLE DB Provider on SharePoint Servers

Install ADOMD.NET on Web Front-End Servers Running Central Administration

Install ADO.NET Data Services to support data feed exports of SharePoint lists

Install PowerPivot from the Command Prompt

Uninstall PowerPivot for SharePoint

Repair PowerPivot for SharePoint

Initial Configuration (PowerPivot for SharePoint)

See Also

PowerPivot Server Administration and Configuration in Central Administration

Install PowerPivot for SharePoint 2010

2/28/2020 • 7 minutes to read • Edit Online

PowerPivot for SharePoint is a collection of middle-tier and backend services that provide PowerPivot data access in a SharePoint 2010 farm. If your organization uses the client application, PowerPivot for Excel 2010, to create workbooks that contain analytical data, you must have PowerPivot for SharePoint to access that data in a server environment. This topic walks you through the basic installation process and it links you additional topics to help you configure PowerPivot.

Applies to: SharePoint 2010

For instructions on how to install PowerPivot and Reporting Services on the same server, see Deployment Checklist: Reporting Services, Power View, and PowerPivot for SharePoint.

Prerequisites

1. You must be a local administrator to run SQL Server Setup.
2. SharePoint Server 2010 enterprise edition is required for PowerPivot for SharePoint. You can also use the evaluation enterprise edition.
3. SharePoint Server 2010 SP2 must be installed. Without it, you cannot configure the farm to use SQL Server 2014 features.
4. The computer must be joined to a domain.
5. You must have a domain user account to provision Analysis Services. In a PowerPivot for SharePoint installation, the Analysis Services service account must be a domain user account so that you can manage it from Central Administration. You will type the account and credentials on the **Server Configuration** page as part of the steps in this document.
6. The PowerPivot instance name must be available. You cannot have an existing PowerPivot named instance on the computer on which you are installing PowerPivot for SharePoint.
7. The PowerPivot for SharePoint instance cannot be part of a SQL Server Failover Cluster. Use high availability features of the SharePoint Product. For example, Excel Services manages load balancing of PowerPivot for SharePoint servers. For more information, see Manage Excel Services data model settings (SharePoint Server 2013) (https://technet.microsoft.com/library/jj219780.aspx).
8. If you are installing PowerPivot for SharePoint on an existing farm, you must have one or more SharePoint web applications that are configured for classic mode authentication. PowerPivot data access will only work if the web application supports classic mode authentication. For more information about classic mode requirements, see PowerPivot Authentication and Authorization.
9. Review the following additional topics to understand system and version requirements:
   - Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm

Step 1: Install PowerPivot for SharePoint

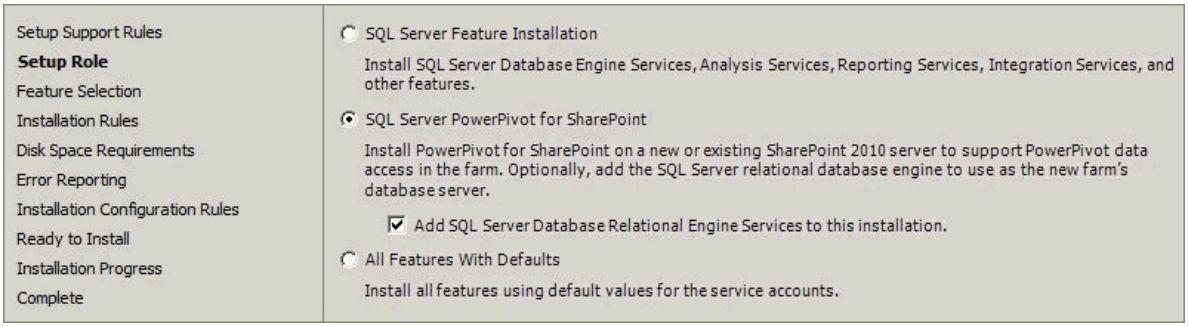
In this step, you run SQL Server Setup to install PowerPivot for SharePoint. In a subsequent step, you will configure the server as a post-installation task.

1. Insert the installation media or open a folder that contains the setup files for SQL Server and then double-click **setup.exe**.
2. Click **Installation** in the left navigation pane.
3. Click **New SQL Server stand-alone installation or add features to an existing installation**.
4. On the **Product Key** page, specify the evaluation edition or enter a product key for a licensed copy of the enterprise edition.

   Click **Next**.
5. Accept the Microsoft Software License Terms of agreement and We appreciate it if you also turn on the customer experience and error reporting. Click **Next**.
6. Update the setup files if you are prompted to do so.
7. On the **Install Rules** page, setup identifies any problems that might prevent it from installing. Review the list to determine whether Setup detected potential problems on the system.

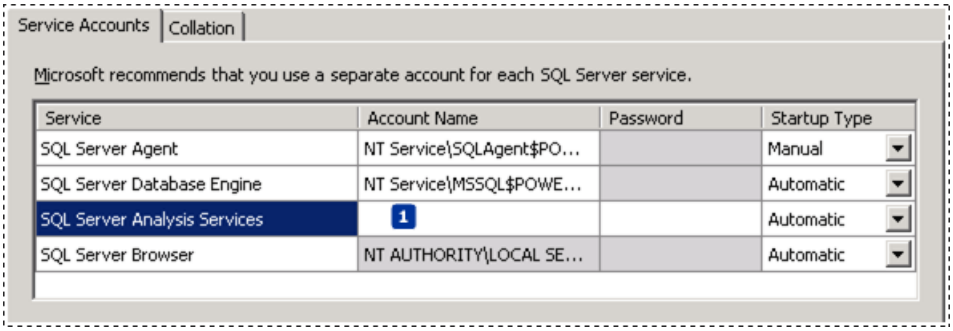
   > **NOTE**
   >
   > Because Windows Firewall is enabled, you will be warned to open ports to enable remote access. This warning is generally not applicable to PowerPivot installations. Connections to PowerPivot services and data files are made using the SharePoint ports that are already open for SharePoint service-to-service communication.

   Click **Next**. Wait while SQL Server Setup program files are installed on the server.
8. On the **Setup Role** page, select **SQL Server PowerPivot for SharePoint**.
9. Optionally, you can add an instance of the Database Engine to your installation. You might do this if you are setting up a new farm and need a database server to run the farm's configuration and content databases. If you add the Database Engine, it will be installed as a PowerPivot named instance. Whenever you need to specify a connection to this instance (for example, in the farm configuration wizard if you are using that wizard to configure the farm), enter the database name in this format: < `servername` >\PowerPivot.

10. Click **Next**.
11. On the **Feature Selection** page, a read-only list of the features that will be installed is displayed for informational purposes. You cannot add or remove items that are preselected for this role. Click **Next**.
12. On the **Feature Rules** page, click **Next**. The page may be skipped.
13. On the **Instance Configuration** page, a read-only instance name of 'PowerPivot' is displayed for informational purposes. This instance name of **POWERPIVOT** is **required and cannot be modified**. However, you can enter a unique Instance ID to specify a descriptive directory name and registry keys. Click **Next**.
14. On the **Server Configuration** page, type desired account information.

For SQL Server Analysis Services, you must specify a domain user account. Do not specify a built-in account. Domain accounts are required for managing the Analysis Services service account as a *managed account* in SharePoint Central administration.



If you added the SQL Server Database Engine and SQL Server Agent, you can configure the services to run under domain user accounts or under the default virtual account.

Never use your own domain user account to provision any service. Doing so grants the server the same permissions that you have to the resources in your network. If the server is compromised by a malicious user, that user will be logged in under your domain credentials, with the ability to download or use the same data and applications that you do.

15. Click **Next**.
16. If you are installing the Database Engine, the Database Engine Configuration page appears. In Database Engine Configuration, click **Add Current User** to grant your user account administrator permissions on the Database Engine instance. Click **Add** to add additional accounts. Click **Next**.
17. On the **Analysis Services Configuration** page, click **Add Current User** to grant your user account administrative permissions. You will need administrative permission to configure the server after Setup is finished.
18. On the same page, add the Windows user account of any person who also requires administrative permissions. For example, any user who wants to connect to the Analysis Services service instance in SQL Server Management Studio to troubleshoot database connection problems or get version information must have system administrator permissions on the server. Add the user account of any person who might need to troubleshoot or administer the server now.
19. Click **Next**.
20. Click **Next** on each of the remaining pages until you get to the Ready to Install page.
21. Click **Install**.

> **TIP**
> If you need to trouble shoot the SQL Server installation, see View and Read SQL Server Setup Log Files.

Step 2: Configure the Server

> **IMPORTANT**
> SharePoint 2010 SP2 must be installed before you can configure PowerPivot for SharePoint or a SharePoint farm that uses a SQL Server 2014 database server. If you have not yet installed the service pack, do so now before you begin configuring the server.

Installation is not complete until the server is configured. In this release, server configuration is always performed as post-installation task, using one of the following approaches: PowerPivot Configuration Tool, Central Administration, or PowerShell. To continue, choose one of the following approaches:

- Configure or Repair PowerPivot for SharePoint 2010 (PowerPivot Configuration Tool)
- PowerPivot Server Administration and Configuration in Central Administration
- PowerPivot Configuration using Windows PowerShell

Connecting to the Database Engine Instance. When you installed PowerPivot for SharePoint, SQL Server Setup gave you the option of adding an instance of the Database Engine to your installation. You might have added a Database Engine instance to your installation if you are setting up a new farm and need a database server to run the farm's configuration and content databases. If you added the Database Engine, it was installed as a PowerPivot named instance. Whenever you need to specify a connection to this instance (for example, in the farm configuration wizard if you are using that wizard to configure the farm), remember to enter the database name in this format: < `servername` >\PowerPivot.

Step 3: Install Analysis Services OLE DB providers on Excel Services Application Servers

Additional installation steps are required if you run Excel Calculation Services and PowerPivot on separate application servers. On the application servers running Excel Calculation Services, install the appropriate version of the Analysis Services OLE DB (MSOLAP) provider.

- The SQL Server 2014 version of MSOLAP is included in SQL Server Setup, therefore explicitly installing the SQL Server 2014 version of MSOLAP is only required if your application server is not a PowerPivot application server.

  > **NOTE**
  > The Excel Calculation Services application server also needs an instance of the file **Microsoft.AnalysisServices.Xmla.dll** in the global assembly. To install the .dll on the application server, install SQL Server Management Studio. Select the "Management Tools - Complete" on the **Feature Selection** page of the SQL Server Setup wizard.

- If you want the application server to support older PowerPivot workbooks, you need to install the SQL Server 2008 R2 version of MSOLAP.

For more information about installing the provider, including verification steps, see Install the Analysis Services OLE DB Provider on SharePoint Servers

Step 4: Verify the Installation

In this last step, you will verify that both SharePoint 2010 and PowerPivot for SharePoint are fully functional. For instructions, see Verify a PowerPivot for SharePoint Installation.

See Also

PowerPivot for SharePoint 2010 Installation Deployment Checklist: Reporting Services, Power View, and PowerPivot for SharePoint Deployment Checklist: Scale-out by adding PowerPivot Servers to a SharePoint 2010 farm Deployment Checklist: Multi-Server Installation of PowerPivot for SharePoint 2010

Install the Analysis Services OLE DB Provider on SharePoint Servers

3/11/2020 • 7 minutes to read • Edit Online

The Microsoft OLE DB Provider for Analysis Services (MSOLAP) is an interface that client applications use to interact with Analysis Services data. In a SharePoint environment that includes PowerPivot for SharePoint, the provider handles connection requests for PowerPivot data.

The data provider is included in the PowerPivot for SharePoint installation package (spPowerPivot.msi), but might require manual installation. There are two reasons why you might need to manually install a client library or data provider on a SharePoint server.

- **Enable backwards compatibility**. SQL Server 2008 R2 workbooks specify the SQL Server 2008 R2 version of the Analysis Services OLE DB provider in their connection string. As such, this provider version must be present on the computer in order for the request to succeed.
- **Enable data access on a dedicated Excel Services instance**. If your SharePoint farm includes Excel Services on a server that does not also have PowerPivot for SharePoint, install the SQL Server 2012 SP1 version of the provider and other client connectivity components by using the PowerPivot for SharePoint installation package.

> **NOTE**
> These scenarios are not mutually exclusive. Hosting multiple workbook versions on a farm that includes application servers running Excel Services without a PowerPivot for SharePoint instance will require that you install both older and newer versions of the data provider on each Excel Services computer.

Versions of the OLE DB Provider Supporting PowerPivot Data Access

A SharePoint farm might include multiple versions of the Analysis Services OLE DB provider, including older versions that do not support PowerPivot data access.

By default, SharePoint 2010 installs the SQL Server 2008 version of the provider. Although it is identified as MSOLAP.4 (the same version number used for SQL Server 2008 R2), this version does not work for PowerPivot data access. In order for connections to succeed, you must have the SQL Server 2008 R2 or SQL Server 2014 version of the provider.

A post SQL Server 2008 version of the OLE DB provider includes transports and connection support for PowerPivot data structures. PowerPivot workbooks use newer versions of this provider to request query processing from PowerPivot servers in the farm. To get an updated version, you can download and install it through a SQL Server Feature Pack page.

The following table describes the valid versions:

PRODUCT VERSION	FILE VERSION	VALID FOR:

PRODUCT VERSION	FILE VERSION	VALID FOR:
SQL Server 2008 R2	MSOLAP100.dll in the file system MSOLAP.4 in an Excel connection string 10.50.1600 or later in file version details	Use for data models created using the SQL Server 2008 R2 version of PowerPivot for Excel.
SQL Server 2012	MSOLAP110.dll in the file system MSOLAP.5 in an Excel connection string 11.0.0000 or later in file version details	Use for data models created using the SQL Server 2012 or SQL Server 2014 version of PowerPivot for Excel.
SQL Server 2014	MSOLAP120.dll in the file system 12.0.20000 or later in file version details	Use for data models other than PowerPivot models.

Why you need to install the OLE DB Provider

There are two scenarios that call for manually installing the OLE DB provider on servers in the farm.

The most common scenario is when you have older and newer versions of PowerPivot workbooks that are saved in document libraries in the farm. If analysts in your organization are using the SQL Server 2008 R2 version of PowerPivot for Excel, and they save those workbooks to a SQL Server 2012PowerPivot for SharePoint installation, the older workbook will not work. Its connection string will reference an older version of the provider, which won't be on the server unless you install it. Installing both versions will enable data access for PowerPivot workbooks created in older and newer versions of PowerPivot for Excel. SQL Server 2012 Setup does not install the SQL Server 2008 R2 version of the provider, so you must install it manually if you are using workbooks from a previous version.

The second scenario is when you have a server in a SharePoint farm that runs Excel Services, but not PowerPivot for SharePoint. In this case, the application server that runs Excel Services must be manually updated to use a newer version of the provider. This is necessary for connecting to a PowerPivot for SharePoint instance. If Excel Services is using an older version of the provider, the connection request will fail. Note that the provider must be installed by using SQL Server setup or the PowerPivot for SharePoint installation package (spPowerPivot.msi) in order to ensure that all components required support PowerPivot are installed.

Install the SQL Server 2012 OLE DB Provider on an Excel Services server by using SQL Server Setup

Use the following instructions to add the OLE DB provider and other client connectivity components to SharePoint servers that do not already have them installed, such as application servers that run Excel Services without PowerPivot for SharePoint on the same hardware.

Use these instructions to install the current Analysis Services OLE DB provider and to add the **Microsoft.AnalysisServices.Xmla.dll** to the global assembly.

Run SQL Server Setup and Install the Client Connectivity Tools

1. On the application server that hosts Excel Services, run SQL Server Setup.
2. On the Installation page, choose **New SQL Server stand-alone installation or add features to an existing installation**.
3. On the Installation Type page, choose **Perform a new installation of SQL Server 2012**.

4. On the Setup Role page, choose **SQL Server Feature Installation**.
5. On the **Feature Selection** page, click **Client Tools Connectivity**. This option installs **Microsoft.AnalysisServices.Xmla.dll**

   Do not select any other features.
6. Click **Next** to finish the wizard, and then click **Install** to run Setup.
7. Repeat the previous steps if you have other servers running Excel Services, without a PowerPivot for SharePoint installation on the same server.

Verify MSOLAP.5 is a trusted provider

1. In Central Administration, click **Manage service applications**, and then click the Excel Services service application.
2. Click **Trusted Data Providers**.
3. Verify that MSOLAP.5 appears in the list. Depending on how you configured PowerPivot for SharePoint, MSOLAP.5 might already be trusted. If you used the PowerPivot Configuration tool, but then excluded this action from the task list, MSOLAP.5 will not be trusted by Excel Services and now needs to be added manually.
4. If MSOLAP is not listed, click **Add Trusted Data Provider**.
5. In Provider ID, type `MSOLAP.5`.
6. For Provider Type, ensure that OLE DB is selected.
7. In Provider Description, type **Microsoft OLE DB Provider for OLAP Services 11.0**.

Verify installation

1. Go to Program files\Microsoft Analysis Services\AS OLEDB\110.
2. Right-click msolap110.dll and select **Properties**.
3. Click **Details**.
4. View the file version information. The version should include 11.00.<buildnumber>.
5. In the Windows\assembly folder, verify that Microsoft.AnalysisServices.Xmla.dll, version 11.0.0.0, is listed.

Use the PowerPivot for SharePoint Installation package (spPowerPivot.msi) to install the SQL Server 2012 OLE DB Provider

Install the SQL Server 2012 SP1 OLE DB Provider on and Excel Services Server by using the PowerPivot for SharePoint installation package **(spPowerPivot.msi)**.

Download the MSOLAP.5 provider from the SQL Server 2012 SP1 Feature Pack.

1. Browse to Microsoft® SQL Server® 2012 SP1 Feature Pack
2. Click **Install Instructions**.
3. See the section "Microsoft Analysis Services OLE DB Provider for Microsoft SQL Server 2012 SP1". Download the file and start the installation.
4. On the **Feature Selection** page, select **Analysis Services OLE DB Provider for SQL Server**. Unselect the other components and complete the installation. For more information on spPowerPivot.msi, see Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013).
5. Register MSOLAP.5 as a trusted provider with SharePoint Excel Services. For more information, see Add

MSOLAP.5 as a Trusted Data Provider in Excel Services.

Install the SQL Server 2008 R2 OLE DB Provider to host earlier version workbooks

Use the following instructions to install the SQL Server 2008 R2 version of the MSOLAP.4 provider, and register the Microsoft.AnalysisServices.ChannelTransport.dll file. The ChannelTransport is a subcomponent of the Analysis Services OLE DB provider. The SQL Server 2008 R2 version of the provider reads the registry when using ChannelTransport to make a connection. Registering this file is a post-installation step required only for connections handled by the SQL Server 2008 R2 provider on a SQL Server 2012 server.

Step 1: Download and install the client library

1. On the SQL Server 2008 R2 Feature Pack page, find Microsoft Analysis Services OLE DB Provider for Microsoft SQL Server 2008 R2.
2. Download the x64 Package of the `SQLServer2008_ASOLEDB10.msi` installation program. Although the file name contains SQLServer2008, it is the correct file for the SQL Server 2008 R2 version of the provider.
3. On the computer that has an installation of PowerPivot for SharePoint, run the .msi to install the library.
4. If you have other servers in the farm that run just Excel Services, without PowerPivot for SharePoint on the same server, repeat the previous steps to install the 2008 R2 version of the provider on the Excel Services computer.

Step 2: Register the Microsoft.AnalysisServices.ChannelTransport.dll file

1. Use the regasm.exe utility to register the file. If you have not run regasm.exe before, add its parent folder, C:\Windows\Microsoft.NET\Framework64\v4.0.30319\, to the system path variable.
2. Open a command prompt with administrator permissions.
3. Go to this folder C:\Windows\assembly\GAC_MSIL\Microsoft.AnalysisServices.ChannelTransport\10.0.0.0__89845dcd8080cc91
4. Enter the following command: `regasm microsoft.analysisservices.channeltransport.dll`
5. Repeat the previous steps for any computer on which you manually installed the 2008 R2 version of the provider.

Verify installation

1. You should now be able to slice or filter SQL Server 2008 R2 workbooks. If an error occurs, verify that you used the 64-bit version of regasm.exe to register the file.
2. Additionally, you can check the file version.

   Go to `C:\Program files\Microsoft Analysis Services\AS OLEDB\10`. Right-click **msolap100.dll** and select **Properties**. Click **Details**.

   View the file version information. The version should include 10.50.<buildnumber>.

See Also

PowerPivot for SharePoint 2010 Installation

Install ADO.NET Data Services to support data feed exports of SharePoint lists

10/3/2019 • 2 minutes to read • Edit Online

ADO.NET Data Services is required for a data feed export of SharePoint lists. SharePoint 2010 does not include this component in the SharePoint Prerequisite Installer program, so you must install it manually.

Without this prerequisite, you will get the following error when you attempt to use a SharePoint list that was exported as a data feed: "For security reasons DTD is prohibited in this XML document. To enable DTD processing set the ProhibitDtd property on XmlReaderSettings to false and pass the settings into XmlReader.Create method."

Use the following instructions to install ADO.NET Data Services on every SharePoint server for which you want to allow lists to be exported as data feeds.

Download and install ADO.NET Data Services

1. Go to the hardware and software requirements documentation for SharePoint 2010, Hardware and Software Requirements (SharePoint 2010)
2. In **Access to applicable software**, find the link for ADO.NET Data Services 3.5 that corresponds to the operating system you are using (either Windows Server 2008 SP2 or Windows Server 2008 R2).
3. Click the link and run the setup program that installs the service.

See Also

PowerPivot for SharePoint 2010 Installation

Install ADOMD.NET on Web Front-End Servers Running Central Administration

10/3/2019 • 2 minutes to read • Edit Online

If you install PowerPivot for SharePoint into a farm that has the topology of Central Administration, without Excel Services or PowerPivot for SharePoint, download and install the Microsoft ADOMD.NET client library if you want full access to the built-in reports in the PowerPivot management dashboard. Some reports in the dashboard use ADOMD.NET to access internal data that provides reporting data on PowerPivot query processing and server health in the farm.

Applies to: SharePoint 2010

For SharePoint 2013, the provider is included in the SQL Server feature pack. For information on how to download spPowerPivot.msi, see Microsoft SQL Server 2014 Feature Pack

Download and install the client library

1. On the SQL Server 2014 Feature Pack page, find Microsoft ADOMD.NET.
2. Download the x64 Package of the `SQL_AS_ADOMD.msi` installation program.
3. Run the .msi to install the library.
4. Reset IIS after installation is finished. Open an administrative command prompt and type **IISRESET**.

Verify installation

1. Go to Windows\Assembly.
2. Right-click Microsoft.AnalysisServices.AdomdClient and select **Properties**.
3. Click **Version**.
4. Verify that the version includes 12.00.<build number> and that the description is Microsoft.AnalysisService.AdomdClient.

See Also

PowerPivot Management Dashboard and Usage Data

Install PowerPivot from the Command Prompt

10/23/2019 • 3 minutes to read • Edit Online

You can run Setup from the command line to install SQL Server PowerPivot for SharePoint. You must include the `/ROLE` parameter in your command and exclude the `/FEATURES` parameter.

Prerequisites

SharePoint Server 2010 enterprise edition with Service Pack 1 (SP1) must be installed.

You must use domain accounts to provision Analysis Services.

The computer must be joined to the same domain as the SharePoint farm.

/ROLE based installation options

For PowerPivot for SharePoint deployments, the `/ROLE` parameter is used in place of the `/FEATURES` parameter. Valid values include:

- `SPI_AS_ExistingFarm`
- `SPI_AS_NewFarm`

Both roles install application, configuration, and deployment files that enable a PowerPivot for SharePoint to run in a SharePoint farm. Specifying either role will cause Setup to check for hardware and software requirements necessary for SharePoint integration.

The existing farm option assumes that a SharePoint farm is already in place. The new farm option assumes that you will create a new farm; it supports the addition of a Database Engine instance in the command line syntax so that you can use the Database Engine instance as the farm's database server.

In contrast with the previous releases, all server configuration tasks are performed as post-installation tasks. If you are automating installation and configuration steps, you can use PowerShell to configure the server. For more information, see PowerPivot Configuration using Windows PowerShell.

Example Commands

The following examples illustrate the usage of each option. Example 1 shows `SPI_AS_ExistingFarm`.

```
Setup.exe /q /IAcceptSQLServerLicenseTerms /ACTION=install /ROLE=SPI_AS_ExistingFarm /INSTANCENAME=PowerPivot
/INDICATEPROGRESS/ASSVCACCOUNT=<DomainName\UserName> /ASSVCPASSWORD=<StrongPassword> /ASSYSADMINACCOUNTS=
<DomainName\UserName>
```

Example 2 shows `SPI_AS_NewFarm`. Notice that it includes parameters for provisioning the Database Engine.

```
Setup.exe /q /IAcceptSQLServerLicenseTerms /ACTION=install /ROLE=SPI_AS_NewFarm /INSTANCENAME=PowerPivot
/INDICATEPROGRESS/SQLSVCACCOUNT=<DomainName\UserName> /SQLSVCPASSWORD=<StrongPassword> /SQLSYSADMINACCOUNTS=
<DomainName\UserName> /AGTSVCACCOUNT=<DomainName\UserName> /AGTSVCPASSWORD=<StrongPassword> /ASSVCACCOUNT=
<DomainName\UserName> /ASSVCPASSWORD=<StrongPassword> /ASSYSADMINACCOUNTS=<DomainName\UserName>
```

Modifying the command syntax

Use the following steps to modify the example command syntax.

1. Copy the following command into Notepad:

   ```
   Setup.exe /q /IAcceptSQLServerLicenseTerms /ACTION=install /ROLE=SPI_AS_ExistingFarm
   /INSTANCENAME=PowerPivot /INDICATEPROGRESS/ASSVCACCOUNT=<DomainName\UserName> /ASSVCPASSWORD=
   <StrongPassword> /ASSYSADMINACCOUNTS=<DomainName\UserName>
   ```

   The `/q` parameter runs Setup in quiet mode, which suppresses the user interface.

   The `/IAcceptSQLServerLicenseTerms` is required when the `/q` or `/qs` parameter is specified for un-attended installations.

   The `/action` parameter instructs Setup to perform an installation.

   The `/role` parameter instructs Setup to install the Analysis Services program and configuration files required for PowerPivot for SharePoint. This role also detects and uses the existing farm connectivity information to access the SharePoint configuration database. This parameter is required. Use it instead of the `/features` parameter to specify the components to install.

   The `/instancename` parameter specifies 'PowerPivot' as a named instance. This value is hard-coded and cannot be changed. It is specified in the command for educational purposes so that you know how the service is installed.

   The `/indicateprogress` parameter allows you to monitor progress in the command prompt window.

2. The `PID` parameter is omitted from the command, which causes the Evaluation edition to be installed. If you want to install the Enterprise edition, add the PID to your Setup command and provide a valid product key.

   ```
   /PID=<product key for an Enterprise installation>
   ```

3. Replace the placeholders for <domain\username> and <StrongPassword>with valid user accounts and passwords.

   The `/assvaccount` and **/assvcpassword** parameters are used to configure the Analysis Services service instance on the application server. Replace these placeholders with valid account information.

   The **/assysadminaccounts** parameter must be set to the identity of the user who is running SQL Server Setup. You must specify at least one system administrator. Note that SQL Server Setup no long grants automatic sysadmin permissions to members of the built-in Administrators group.

4. Remove line breaks.

5. Select the entire command and then click **Copy** on the Edit menu.

6. Open an administrator command prompt. To do this, click **Start**, right-click the command prompt, and select **Run as administrator**.

7. Navigate to the drive or shared folder that contains the SQL Server installation media.

8. Paste the revised command into the command line. To do this, click the icon in the top left corner of the command prompt window, point to **Edit**, and then click **Paste**.

9. Press **Enter** to run the command. Wait for setup to complete. You can monitor Setup's progress in the command prompt window.

10. To verify installation, check the summary.txt file at \Program Files\SQL Server\120\Setup Bootstrap\Log.

Final result should say "Passed" if the server installed without errors.

11. Configure the server. At a minimum, you must deploy solutions, create a service application, and enable the feature for each site collection. For more information, see Configure or Repair PowerPivot for SharePoint 2010 (PowerPivot Configuration Tool) or PowerPivot Server Administration and Configuration in Central Administration.

See Also

Configure PowerPivot Service Accounts
PowerPivot for SharePoint 2010 Installation

Repair PowerPivot for SharePoint

10/3/2019 • 2 minutes to read • Edit Online

You can use the PowerPivot Configuration Tool to repair an existing installation of PowerPivot for SharePoint. The tool reads the configuration settings for a local PowerPivot for SharePoint instance and provides actions for re-configuring a setting that is inconsistent with a PowerPivot for SharePoint installation. For example, if you stop a service or retract a solution, the tool will offer actions to start the service and deploy the solution.

For more information, see Configure or Repair PowerPivot for SharePoint 2010 (PowerPivot Configuration Tool).

See Also

Drop a SQL Server 2014 Installation
Install PowerPivot for SharePoint 2010
Uninstall an Existing Instance of SQL Server (Setup)
PowerPivot for SharePoint 2010 Installation

Initial Configuration (PowerPivot for SharePoint)

3/11/2020 • 17 minutes to read • Edit Online

Use the steps in this topic to configure an initial installation of PowerPivot for SharePoint. The easiest way to configure an initial installation is to use the PowerPivot Configuration tool. It automates all of the configuration steps that are described below.

Alternatively, if you want more control over how the server is configured, you can use Central Administration and the information in this topic to perform each step individually.

Prerequisites

The SharePoint server must have been installed using the Server Farm installation option in SharePoint setup. A standalone SharePoint server that uses a built-in database is not supported. For more information, see Guidance for Using SQL Server BI Features in a SharePoint 2010 Farm.

> **IMPORTANT**
> SharePoint 2010 SP1 must be installed before you can configure either PowerPivot for SharePoint, or a SharePoint farm that uses a SQL Server 2012 database server. If you have not yet installed the service pack, do so now before you begin configuring the server.

PowerPivot for SharePoint must be installed. At a minimum, the farm solution must be deployed. Use either the PowerPivot Configuration tool or PowerShell script to deploy the farm solution. Instructions for this step are provided in this topic.

The computer must be joined to a domain. The accounts that you specify for services must be domain user accounts. At a minimum, you will need one domain account for the PowerPivot service application. If you are configuring additional services (such as Excel Services), you should have separate accounts for each service you provision.

You must be a farm administrator to add PowerPivot for SharePoint to the farm. You must know the passphrase for adding servers and applications to the farm.

Step 1: Deploy PowerPivot Solutions

There are two solutions that must be installed and deployed: a farm solution and a web application solution.

Install and Deploy the Farm Solution

In the previous release, SQL Server Setup installed and deployed the farm solution. In this release, you must either use the PowerPivot Configuration Tool or PowerShell script to deploy the farm solution. You cannot use Central Administration to deploy the farm solution. This is the only step in PowerPivot for SharePoint configuration that cannot be performed in Central Administration. After the farm solution is deployed, you can use Central Administration and the steps in this topic to configure a PowerPivot for SharePoint installation.

In this step, you run PowerShell to install and deploy the farm solution. For more information, see PowerPivot Configuration using Windows PowerShell.

1. Open a SharePoint 2010 Management Shell using the **Run as Administrator** option.
2. Run the first cmdlet:

```
Add-SPSolution -LiteralPath "C:\Program Files\Microsoft SQL
Server\120\Tools\PowerPivotTools\ConfigurationTool\Resources\PowerPivotFarm.wsp"
```

The cmdlet returns the name of the solution, its solution ID, and Deployed=False. In the next step, you will deploy the solution.

3. Run the second cmdlet to deploy the solution:

```
Install-SPSolution -Identity PowerPivotFarm.wsp -GACDeployment -Force
```

Deploy the web application solution

1. Click the Start button, select **All Programs**, select **Microsoft SharePoint Products 2010**, and then select **SharePoint 2010 Central Administration**.
2. In SharePoint 2010 Central Administration, in System Settings, click **Manage farm solutions**.

   You should see two separate solution packages: powerpivotfarm.wsp and powerpivotwebapp.wsp. The first solution (powerpivotfarm.wsp) must already be deployed. Once it is deployed, it never needs to be deployed again. The second solution (powerpivotwebapp.wsp) is deployed for Central Administration, but you must deploy this solution manually for each SharePoint web application that will support PowerPivot data access.
3. Click **powerpivotwebapp.wsp**.
4. Click **Deploy Solution.**
5. In **Deploy To?**, select the SharePoint web application to which you want to add PowerPivot feature support.
6. Click **OK**.
7. Repeat for other SharePoint web applications that will also support PowerPivot data access.

Step 2: Start Services on the Server

A PowerPivot for SharePoint deployment requires that your farm include the following services: Excel Calculation Services, Secure Store Service, and Claims to Windows token service.

The Claims to Windows Token Service is required for Excel Services and PowerPivot for SharePoint. It is used to establish connections to external data sources using the Windows identity of the current SharePoint user. This service must run on every SharePoint server that has Excel Services or PowerPivot for SharePoint enabled. If the service is not already started, you must start it now to enable Excel Services to forward authenticated requests to the PowerPivot System Service.

1. In Central Administration, in System Settings, click **Manage services on server**.
2. Start the Claims to Windows Token Service.
3. Start Excel Calculation Services.
4. Start Secure Store Service.
5. Verify that both SQL Server Analysis Services and SQL Server PowerPivot System Service are started.

Step 3: Create a PowerPivot Service Application

The next step is to create a PowerPivot service application.

1. In Central Administration, in Application Management, click **Manage service applications**.

2. In the **Service Applications** ribbon, click **New**.
3. Select **SQL Server PowerPivot Service Application**. If it does not appear in the list, PowerPivot for SharePoint is not installed or the solution is not deployed.
4. In the **Create New PowerPivot Service Application** page, enter a name for the application. The default is PowerPivotServiceApplication<number>. If you are creating multiple PowerPivot service applications, a descriptive name will help other administrators understand how the application is used.
5. In Application Pool, create a new application pool and select a security account for it. A domain user account is required.
6. In **Database Server**, choose a database server on which to create the service application database. The default value is the SQL Server Database Engine instance that hosts the farm configuration databases.
7. In **Database Name**, the default value is PowerPivotServiceApplication1_<guid>. The default database name corresponds to the default name of the service application. If you entered a unique service application name, follow a similar naming convention for your database name so that you can manage them together.
8. In **Database Authentication**, the default is Windows Authentication. If you choose **SQL Authentication**, refer to the SharePoint administrator guide for best practices on how to use this authentication type in a SharePoint deployment.
9. Select the checkbox for **Add the proxy for this PowerPivot service application to the default proxy group.** This adds the service application connection to the default service connection group. You must have at least one PowerPivot service application in the default connection group.

   If a PowerPivot service application is already listed in the default connection group, do not add a second service application to that group. Adding two service applications of the same type of the default connection group is not a supported configuration. For more information about how to use additional service applications in a connection group, see Connect a PowerPivot Service Application to a SharePoint Web Application in Central Administration.
10. Click **OK.** The service will appear alongside other managed services in the farm's service application list.

Step 4: Enable Excel Services

PowerPivot for SharePoint requires Excel Services to support PowerPivot data access in the farm. You can determine whether Excel Services is already enabled by confirming whether Excel Services Application appears in the list of service applications in Central Administration. If Excel Services is not listed, follow these steps to enable it now.

1. In Central Administration, in Application Management, click **Manage service applications**.
2. In the Service Applications ribbon, in Create, click **New**.
3. Select **Excel Services Application**.
4. In Create New Excel Services Application, specify a name (for example, Excel Services Application).
5. In Application Pool, select Create new application pool and give it a descriptive name (for example, Excel Services Application Pool).
6. In Configurable, select a Windows domain user account for this application pool identity.
7. Keep the default checkbox that adds the service application proxy to the default service connection list.
8. Click **OK**.
9. Click the Excel Services application you just created.

10. Click **Trusted File Locations** and on this page, select your trusted location. (Typically, this is listed as **http://** in the Address column.) To ensure that both Excel Services and PowerPivot service have access to the workbook, you must include SharePoint as an Excel Services trusted location. PowerPivot System Service cannot access workbooks that are stored outside of a SharePoint farm.
11. In the Workbook Properties area, set **Maximum Workbook Size** to 50.
12. In External Data, set **Allow External data** to **Trusted data connection libraries and embedded**. This setting is required for PowerPivot data access in a workbook.
13. Clear the **Warn on Data Refresh** checkbox to allow preview images of individual worksheets in PowerPivot Gallery. If you choose to keep the warning and workbook settings specify refresh on open, you might get a single preview image of the warning instead of the pages in your workbook.
14. Click **OK**.

Step 5: Enable Secure Store Service and Configure Data Refresh

PowerPivot for SharePoint requires Secure Store Service to store credentials and the unattended execution account for data refresh. You can determine whether Secure Store Service is already enabled by confirming whether it appears in the list of service applications.

> **IMPORTANT**
> If Secure Store Service is enabled, you should still verify that a master key has been generated for it. For instructions, see Part 2: Generate the Master Key in the following procedure.

If Secure Store Service is not listed, follow these steps to enable it now. By enabling Secure Store, workbook authors and document owners can access a broader range of data source connection options when scheduling data refresh for their published workbooks.

Part 1: Enable Secure Store Service

1. In Central Administration, in Application Management, click **Manage service applications**.
2. In the Service Applications ribbon, in Create, click **New**.
3. Select **Secure Store Service**.
4. In the **Create Secure Store Application** page, enter a name for the application.
5. In **Database**, specify the SQL Server instance that will host the database for this service application. The default value is the SQL Server Database Engine instance that hosts the farm configuration databases.
6. In **Database Name**, enter the name of the service application database. The default value is Secure_Store_Service_DB_<guid>. The default name corresponds to the default name of the service application. If you entered a unique service application name, follow a similar naming convention for your database name so that you can manage them together.
7. In **Database Authentication**, the default is Windows Authentication. If you choose SQL Authentication, refer to the SharePoint administrator guide for guidance on how to use the authentication type in your farm.
8. In Application Pool, select **Create new application pool.** Specify a descriptive name that will help other server administrators identify how the application pool is used.
9. Select a security account for the application pool. Specify a managed account to use a domain user account.
10. Accept the remaining default values, and then click **OK.** The service application will appear alongside other managed services in the farm's service application list.

Part 2: Generate the Master Key

1. Click the Secure Store Service application from the list.
2. In the Service Applications ribbon, click **Manage**.
3. In Key Management, click **Generate New Key**.
4. Enter and then confirm a pass phrase. The pass phrase will be used to add additional secure store shared service applications.
5. Click **OK**.

Part 3: Configure the Unattended PowerPivot Data Refresh Account

Creating an unattended data refresh account for PowerPivot data access is often required for external data access during data refresh. For example, if Kerberos is not enabled, you must create an unattended account that the PowerPivot service can use to connect to external data sources.

For instructions about how to create the unattended PowerPivot data refresh account or other stored credentials that are used in data refresh, see Configure the PowerPivot Unattended Data Refresh Account (PowerPivot for SharePoint) and Configure Stored Credentials for PowerPivot Data Refresh (PowerPivot for SharePoint).

Step 6: Enable Usage Data Collection

PowerPivot for SharePoint uses the SharePoint usage data collection infrastructure to gather information about PowerPivot usage throughout the farm. Although usage data is always part of a SharePoint installation, you might need to enable it before it can be used. For instructions, see Configure Usage Data Collection for (PowerPivot for SharePoint.

Step 7: Increase Maximum Upload Size for SharePoint Web Applications and Excel Services

Because PowerPivot workbooks can be large, you might want to increase the maximum file size. There are two file size settings to configure: Maximum Upload Size for the web application, and Maximum Workbook Size in Excel Services. The maximum file size should be set to the same value in both applications. For instructions, see Configure Maximum File Upload Size (PowerPivot for SharePoint).

Step 8: Activate PowerPivot Feature Integration for Site Collections

Feature activation at the site collection level makes application pages and templates available to your sites, including configuration pages for scheduled data refresh and application pages for PowerPivot Gallery and Data Feed libraries.

1. On a SharePoint site, click **Site Actions**.

   By default, SharePoint web applications are accessed through port 80. This means that you can often access a SharePoint site by entering http://<computer name> to open the root site collection.
2. Click **Site Settings**.
3. In Site Collection Administration, click **Site collection features**.
4. Scroll down the page until you find **PowerPivot Integration Site Collection Feature**.
5. Click **Activate**.
6. Repeat for additional site collections by opening each site and clicking **Site Actions**.

For more information, see Activate PowerPivot Feature Integration for Site Collections in Central Administration.

Step 9: Install the SQL Server 2008 R2 Version of the OLE DB provider on a SQL Server 2012 PowerPivot for SharePoint instance

If you want to run older and newer versions of PowerPivot workbooks side-by-side on the same server, you must install the Analysis Services OLE DB provider that ships in SQL Server 2008 R2 on a SQL Server 2012 PowerPivot for SharePoint server.

Installing the provider will allow workbooks that reference MSOLAP.4 in the data connection string to work as expected on a SQL Server 2012 PowerPivot server. Installing the SQL Server 2008 R2 OLE DB provider is an alternative approach to upgrading workbooks that were created in an earlier version of PowerPivot for Excel.

You can download the provider from SQL Server 2008 R2 Feature Pack page. Look for **Microsoft® Analysis Services OLE DB Provider for Microsoft® SQL Server® 2008 R2**, and then download the x64 Package of the `SQLServer2008_ASOLEDB10.msi` installation program.

For more information about installing the provider, including verification steps, see Install the Analysis Services OLE DB Provider on SharePoint Servers.

Step 10: Verify Installation

PowerPivot query processing in the farm occurs when a user or application opens an Excel workbook that contains PowerPivot data. At a minimum, you can check pages on SharePoint sites to verify that PowerPivot features are available. However, to fully verify an installation, you must have a PowerPivot workbook that you can publish to SharePoint and access from a library. For testing purposes, you can publish a sample workbook that already contains PowerPivot data and use it to confirm that SharePoint integration is correctly configured.

To verify PowerPivot integration with a SharePoint site, do the following:

1. In a browser, open the Web application you created. If you used default values, you can specify http://<your computer name> in the URL address.
2. Verify that PowerPivot data access and processing features are available in the application. You can do this by verifying the presence of PowerPivot-provided library templates:
    a. On Site Actions, click **More Options...**
    b. In Libraries, you should see **Data Feed Library** and **PowerPivot Gallery**. These library templates are provided by the PowerPivot feature and will be visible in the Libraries list if the feature is integrated correctly.

To verify PowerPivot data access on the server, do the following:

1. Upload a PowerPivot workbook to PowerPivot Gallery or any SharePoint library.
2. Click on the document to open it from the library.
3. Click on a slicer or filter the data to start a PowerPivot query. The server will load PowerPivot data in the background and return the results. In the next step, you will connect to the server to verify the data is loaded and cached.
4. Start SQL Server Management Studio from the Microsoft SQL Server 2008 R2 program group in the Start menu. If this tool is not installed on your server, you can skip to the last step to confirm the presence of cached files.
5. In Server Type, select **Analysis Services**.
6. In Server Name, enter **<server-name>\powerpivot**, where **<server-name>** is the name of the computer that has the PowerPivot for SharePoint installation.

7. Click **Connect**.
8. In Object Explorer, click **Databases** to view the list of PowerPivot data files that are loaded.
9. On the computer file system, check the following folder to determine whether files are cached to disk. The presence of cached files is further verification that your deployment is operational. To view the file cache, go to the \Program Files\Microsoft SQL Server\MSAS10_50.POWERPIVOT\OLAP\Backup folder.

Post-Installation Steps

After you verify the installation, finish service configuration by creating a PowerPivot Gallery or tuning individual configuration settings. To make full use of the server components you just installed, you can download PowerPivot for Excel to create and then publish your first PowerPivot workbook.

Install Data Providers Used for Data Refresh

If you enabled data refresh, the server will need the same data providers for external data access that were used by the PowerPivot client application to import the original data (for example, if the data was originally imported using a 32-bit provider, server-side data refresh will also require the 32-bit provider when it accesses the same external data source). For more information, see PowerPivot Data Refresh with SharePoint 2010.

Install ADO.NET Data Services

You must install ADO.NET Data Services 3.5 SP1 if you want to export SharePoint lists as data feeds. For instructions, see Install ADO.NET Data Services to support data feed exports of SharePoint lists.

Create a PowerPivot Gallery

PowerPivot Gallery is a library that includes preview and presentation options for viewing PowerPivot workbooks on a SharePoint site. Using PowerPivot Gallery to publish and view PowerPivot workbooks is recommended for its preview capability. In addition, if you also deployed Reporting Services to the same SharePoint server, a PowerPivot Gallery provides ease of use in creating reports. You can launch Report Builder from within PowerPivot Gallery to base a new report on a published PowerPivot workbook. For more information about creating and using the library, see Create and Customize PowerPivot Gallery and Use PowerPivot Gallery.

Create Additional Trusted Sites in Excel Services

You can add trusted sites in Excel Services to vary permissions and configuration settings on sites that provide Excel workbooks and PowerPivot data. For more information, see Create a trusted location for PowerPivot sites in Central Administration.

Tune Configuration Settings

A PowerPivot service application is created using default properties and values. You can modify configuration settings for individual service applications to change the methodology by which requests are allocated, set server timeouts, change the thresholds for query response report events, or specify how long usage data is retained. For more information about configuration in Central Administration or about using PowerPivot features in SharePoint Web applications, see PowerPivot Server Administration and Configuration in Central Administration.

Install PowerPivot for Excel and Build a PowerPivot Workbook

After you have the server components installed in a farm, you can create your first Excel 2010 workbook that uses embedded PowerPivot data, and then publish it to a SharePoint library in a Web application. Before you can build Excel workbooks that include PowerPivot data, you must start with an installation of Excel 2010, followed by the PowerPivot for Excel add-in that extends Excel to support PowerPivot data import and enrichment.

Add Servers or Applications

When you deploy the PowerPivot solution, feature integration is activated at the site collection level for all site collections in the web application. As you create new Web applications over time, you must deploy the powerpivotwebapp solution to each one. For instructions, see Deploy PowerPivot Solutions to SharePoint.

Depending on how you configure the PowerPivot service application, the PowerPivot System Service will be added to the default connection group, making it available to all web applications that use default connections. However, if you configured your Web applications to use custom service application connection lists, you will need to add the PowerPivot service application to each SharePoint web application for which you want to enable PowerPivot data processing. For more information, see Connect a PowerPivot Service Application to a SharePoint Web Application in Central Administration.

Over time, if you determine that additional data storage and processing capability is needed, you can add a second PowerPivot for SharePoint server instance to the farm. The installation process is almost identical to the steps you followed to add the first server, except for requirements in how you specify instance names and service account information. For instructions, see Deployment Checklist: Scale-out by adding PowerPivot Servers to a SharePoint 2010 farm.

See Also

Features Supported by the Editions of SQL Server 2014
Configure PowerPivot Service Accounts
Create and Configure a PowerPivot Service Application in Central Administration
Deploy PowerPivot Solutions to SharePoint
Activate PowerPivot Feature Integration for Site Collections in Central Administration
PowerPivot for SharePoint 2010 Installation

Command Prompt Installation of Reporting Services SharePoint Mode and Native Mode

5/22/2019 • 3 minutes to read • Edit Online

Reporting Services supports a command-line installation from the SQL Server setup program. This topic contains several examples of command-line installations that are specific to Reporting Services. For a complete description of the command-line options available for all SQL Server components, see Install SQL Server 2014 from the Command Prompt. This topic does not describe command-line options for the Reporting Services add-in for SharePoint products. For information on command installation of the add-in, see Install the add-in using the installation file rsSharePoint.msi.

Applies to: Reporting Services SharePoint mode | Reporting Services Native mode

RSINSTALLMODE (Native Mode)

The primary input setting for installing Reporting Services is the **/RSINSTALLMODE** input setting. The setting has two options: **DefaultNativeMode** and **FilesOnlyMode**

If the installation includes the SQL Server Database engine, the default RSINSTALLMODE is DefaultNativeMode.If the installation does not include the SQL Server Database engine, the default RSINSTALLMODE is FilesOnlyMode.If you choose DefaultNativeMode but the installation does not include the SQL Server Database engine, the installation automatically changes the RSINSTALLMODE to FilesOnlyMode. For more information on the input settings, see Install SQL Server 2014 from the Command Prompt.

RSSHPINSTALLMODE (SharePoint Mode)

The input setting to install Reporting Services in SharePoint mode is **/RSSHPINSTALLMODE**. The input setting has one option: SharePointFilesOnlyMode. The option installs all the files needed for SharePoint mode but, configuration is required following installation. The additional configuration steps are completed using SharePoint Central Administration. For more information, see Install Reporting Services SharePoint Mode for SharePoint 2010.

Examples of SharePoint Mode Installation

The following example installs SQL Server the database engine service and Reporting Services in SharePoint mode as well as the Reporting Services add-in for SharePoint (RS_SHPWFE).

```
setup /q /ACTION=install /FEATURES=SQL, RS_SHP, RS_SHPWFE,TOOLS /INSTANCENAME=MSSQLSERVER
/SQLSYSADMINACCOUNTS="BUILTIN\ADMINISTRATORS" /RSSVCACCOUNT="NT AUTHORITY\NETWORK SERVICE" /SQLSVCACCOUNT="NT
AUTHORITY\NETWORK SERVICE" /AGTSVCACCOUNT="NT AUTHORITY\NETWORK SERVICE"
```

The following example installs only Reporting Services SharePoint mode.

```
Setup.exe /q /ACTION="Install" /IACCEPTSQLSERVERLICENSETERMS /FEATURES="RS_SHP" /INSTANCEDIR="C:\Program
Files\Microsoft SQL Server" /INSTALLSHAREDDIR="C:\Program Files\Microsoft SQL Server"
/INSTALLSHAREDWOWDIR="C:\Program Files (x86)\Microsoft SQL Server" /INSTALLSQLDATADIR="C:\Program
Files\Microsoft SQL Server" /SECURITYMODE="SQL" /SAPWD="*****" /PID="[Your PID Value]" /SQLSYSADMINACCOUNTS="
[Account Name]" "AutoSql Admin Group" /ASSYSADMINACCOUNTS="[Account Name]" /UPDATEENABLED="False"
```

The following example installs all of the SQL Server features, including both Reporting Services Native mode and

SharePoint mode.

```
Setup.exe /q /ACTION="Install" /INSTANCENAME="MSSQLSERVER"
/FEATURES="SQLEngine,Replication,SNAC,SNAC_SDK,Browser,Writer,AS,IS,MDS,Adv_SSMS,BC,BOL,Conn,SDK,DReplay_CTLR,
DReplay_CLT, RS_SHP,RS_SHPWFE,DQC,BIDS,FullText, RS,DQ,SSMS" /INSTANCEDIR="C:\Program Files\Microsoft SQL
Server" /INSTALLSHAREDDIR="C:\Program Files\Microsoft SQL Server" /INSTALLSHAREDWOWDIR="C:\Program Files
(x86)\Microsoft SQL Server" /INSTALLSQLDATADIR="C:\Program Files\Microsoft SQL Server" /SQLSVCACCOUNT="
[Account Name]" /SQLSVCPASSWORD="********" /AGTSVCACCOUNT="[Account Nam]" /AGTSVCPASSWORD="********"
/CTLRSVCACCOUNT="[Account Nam]" /CTLRSVCPASSWORD="********" /CLTSVCACCOUNT="[Account Nam]"
/CLTSVCPASSWORD="********" /ASSVCACCOUNT="[Account Nam]" /ASSVCPASSWORD="********" /RSSVCACCOUNT="[Account
Nam]" /RSSVCPASSWORD="********" /FTSVCACCOUNT="[Account Nam]" /FTSVCPASSWORD="********" /SECURITYMODE="SQL"
/SAPWD="********" /PID="[Your PID Value]" /SQLSYSADMINACCOUNTS="[Account Nam]" "AutoSql Admin Group"
/ASSYSADMINACCOUNTS="BUILTIN\ADMINISTRATORS" /UPDATEENABLED="False" /IACCEPTSQLSERVERLICENSETERMS
/RSSHPINSTALLMODE="SharePointFilesOnlyMode" /RSINSTALLMODE="DefaultNativeMode"
```

Examples of SharePoint Mode Upgrade

The following example upgrades Reporting Services SharePoint mode. **RSUPGRADEPASSWORD** is the password of the existing Report Server service account. RSUPGRADEPASSWORD is a required field in an upgrade scenario unless the Reporting Services service account is a built-in account.

```
Setup.exe /q /ACTION="Upgrade" /INSTANCENAME="MSSQLSERVER" /PID="[PID value]" /FTSVCACCOUNT="[DOMAIN\ACCOUNT]"
/FTSVCPASSWORD="[ACCOUNTPASSSWORD]" /UPDATEENABLED="False" /IACCEPTSQLSERVERLICENSETERMS
/RSUPGRADEPASSWORD="******"
```

The following example can be used to upgrade a SharePoint Mode installation that is based on the SharePoint shared service architecture. The example uses switch ALLOWUPGRADEFORSSRSSHAREPOINTMODE. The switch is not needed for upgrading older versions that are not based on the shared service architecture:

- SQL Server 2008 R2
- SQL Server 2008

```
Setup.exe /q /ACTION="Upgrade" /INSTANCENAME="MSSQLSERVER" /PID="[Your PID Value]" /FTSVCACCOUNT="[ACCOUNT
Name]" /FTSVCPASSWORD="****" /UPDATEENABLED="False" /IACCEPTSQLSERVERLICENSETERMS
/ALLOWUPGRADEFORSSRSSHAREPOINTMODE="True"
```

Examples of Native Mode Installation

```
Setup.exe /q /ACTION="Install" /INSTANCENAME="MSSQLSERVER"
/FEATURES="SQLEngine,Replication,SNAC,SNAC_SDK,Browser,Writer,AS,IS,MDS,Adv_SSMS,BC,BOL,Conn,SDK,DReplay_CTLR,
DReplay_CLT,DQC,BIDS,FullText, RS,DQ,SSMS" /INSTANCEDIR="C:\Program Files\Microsoft SQL Server"
/INSTALLSHAREDDIR="C:\Program Files\Microsoft SQL Server" /INSTALLSHAREDWOWDIR="C:\Program Files
(x86)\Microsoft SQL Server" /INSTALLSQLDATADIR="C:\Program Files\Microsoft SQL Server" /SQLSVCACCOUNT="
[Account Name]" /SQLSVCPASSWORD="********" /AGTSVCACCOUNT="[Account Nam]" /AGTSVCPASSWORD="********"
/CTLRSVCACCOUNT="[Account Nam]" /CTLRSVCPASSWORD="********" /CLTSVCACCOUNT="[Account Nam]"
/CLTSVCPASSWORD="********" /ASSVCACCOUNT="[Account Nam]" /ASSVCPASSWORD="********" /RSSVCACCOUNT="[Account
Nam]" /RSSVCPASSWORD="********" /FTSVCACCOUNT="[Account Nam]" /FTSVCPASSWORD="********" /SECURITYMODE="SQL"
/SAPWD="********" /PID="[Your PID Value]" /SQLSYSADMINACCOUNTS="[Account Nam]" "AutoSql Admin Group"
/ASSYSADMINACCOUNTS="BUILTIN\ADMINISTRATORS" /UPDATEENABLED="False" /IACCEPTSQLSERVERLICENSETERMS
/RSINSTALLMODE="DefaultNativeMode"
```

See Also

Install SQL Server 2014 from the Command Prompt
SysPrep Parameters

Install PowerPivot from the Command Prompt

Verify or Troubleshoot SQL Server BI Feature Installation Problems

5/22/2019 • 2 minutes to read • Edit Online

This section includes information for verifying an installation or troubleshooting errors that arise during installation.

Verify a Reporting Services Installation

5/22/2019 • 4 minutes to read • Edit Online

Reporting Services report servers can be installed in one of two modes, Native or SharePoint. The steps you should follow for verifying the installation depend on the report server mode.

This topic contains the following information:

- Verify SharePoint Mode Installation
- Verify a Native Mode Installation

Verify SharePoint Mode Installation

To verify the Reporting Services Service

1. From SharePoint central administration, click **Manage services on server** in the **System Settings** group.
2. Verify the **SQL Server Reporting Services Service** is installed and in the **Running** state.

   If you do not see the Reporting Services service in the list, verify the service is installed. For more information, see the "Install and start the Reporting Services SharePoint Service" section of Install Reporting Services SharePoint Mode for SharePoint 2010.

To verify the Service Application

1. To verify from Central Administration you have at least one Reporting Services service application, click **Manage Service Applications** in the **Application Management** group.
2. Verify there is a service application of type **SQL Server Reporting Services Service Application** and a corresponding application proxy.
3. Click **near** the name of the service application and then click **Properties** in the SharePoint toolbar. If you click the name of the service application it will open the Management pages of the service application, not the properties page.
4. Verify the **Web Application Association** is configured to point to the desired web application.

To verify the Site collection Feature

1. In site settings, click **Site collection Features** in the **Site Collection Administration** group.
2. Verify the **Report Server Integration Feature** is active.

To Verify Reporting Server content types

1. To verify or add Reporting Services report server content types, see Add Report Server Content Types to a Library (Reporting Services in SharePoint Integrated Mode).

To verify you can launch Report Builder

1. From a document library, click **Documents** in the SharePoint ribbon.
2. Click **New Document** and click **Report Builder Report**. If you do not see this option, review the previous procedure on adding the report server content types to a library.

Create a basic report

1. In a SharePoint document library, create a basic Reporting Services report that only contains a text box, for example a title. The report does not contain any data sources or datasets. The goal is to verify you can open Report Builder and a basic report will preview.

2. Save the report to the document library and the run the report from the library. For more information on creating reports with Report Builder, see Start Report Builder (Report Builder).

Reporting Services samples

1. Complete one of the Reporting Services tutorials. For more information, see Reporting Services Tutorials (SSRS).

2. Download the Adventure works sample database and the Reporting Services sample reports from CodePlex. For more information, see AdventureWorks Report Samples.

Verify a Native Mode Installation

When you install a Native mode 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.

To verify that the report server is installed and running

1. 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.

2. 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**.

3. Open a browser and type the report server URL in the address bar. The address consists of the server name and the virtual directory name that you specified for the report server during setup. By default, the report server virtual directory is named **ReportServer**. You can use the following URL to verify report server installation: http://*<computer name>*/ReportServer*<_instance name>*. The URL will be different if you installed the report server as a named instance. For more information about the URL format, see Configure Report Server URLs (SSRS Configuration Manager). If you are a local administrator on Windows Vista or Windows Server 2008, see Configure a Native Mode Report Server for Local Administration (SSRS).

4. Run reports to test report server operations. For this step, you can create a sample report from a tutorial. For more information, see Create a Basic Table Report (SSRS Tutorial).

To verify that Report Manager is installed and running

1. Open a browser and type the Report Manager URL in the address bar. The address consists of the server name and the virtual directory name that you specified for the Report Manager during setup or in the Report Manager URL page in the Reporting Services Configuration tool. By default, the Report Manager virtual directory is **Reports**. You can use the following URL to verify Report Manager installation:

   http://*<computer name>*/Reports*<_instance name>*.

2. Use Report Manager to create a new folder or upload a file to test whether definitions are passed back to the report server database. If these operations are successful, the connection is functional.

   For more information, see Report Manager (SSRS Native Mode).

To verify that Report Designer is installed and running

1. Open SQL Server Data Tools (SSDT), and create a new project based on a Report Server project type. For more information on using the Report Server Project Wizard, see Reporting Services in SQL Server Data Tools (SSDT) in SQL Server Books Online.

2. If you installed report samples, open the sample report project files and publish the reports to a report server.

See Also

Troubleshoot a Reporting Services Installation
Cause and Resolution of Reporting Services Errors

Troubleshoot a Reporting Services Installation

12/16/2019 • 12 minutes to read • Edit Online

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 The latest information regarding issues with SQL Server 2014, see Reporting Services SQL Server 2012 Tips Tricks, and Troubleshooting

For information about other errors and issues related to Reporting Services see Troubleshoot SSRS issues and errors.

Review the Online Release Notes in case the issue you encounter is described in the release notes.

This topic contains the following information:

- Check Setup Logs
- Check Prerequisites
- Troubleshoot Problems with SharePoint Mode installations
- Troubleshoot Problems with the Native Mode Installations
- Additional Resources

Check Setup Logs

Setup errors are recorded in log files in the **Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log** folder. A subfolder is created each time you run Setup. The subfolder name is the time and date you ran Setup. For instructions on how to view the Setup log files, see View and Read SQL Server Setup Log Files.

- 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.

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.
- 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 SharePoint Mode installations

- Reporting Services Configuration Manager Does not start
- You do not see the SQL Server Reporting Services service in SharePoint Central Administration after installing SQL Server 2012 SSRS in SharePoint mode
- Reporting Services PowerShell cmdlets are not available and commands are not recognized
- You see an error message indicating the URL is not configured
- Setup fails on a computer with SharePoint installed but it is not configured
- SharePoint Central Administration Page is blank
- You see an error Message when you try to create a new Report Builder Report
- You see an error message that RS_SHP is not supported with PREPAREIMAGE

Reporting Services Configuration Manager Does not start

Description: This issue is by design in SQL Server 2014. The Reporting Services is now architected for the SharePoint service architecture. The Configuration Manager is no longer needed to configure and administer Reporting Services in SharePoint mode.

Workaround: Use SharePoint Central Administration to configure a report server in SharePoint mode. For more information, see Manage a Reporting Services SharePoint Service Application

You do not see the SQL Server Reporting Services service in SharePoint Central Administration after installing SQL Server 2012 SSRS in SharePoint mode

Description: If after successfully installing SQL Server 2014 Reporting Services in SharePoint mode and the SQL Server 2014 Reporting Services Add-in for SharePoint 2010, you do not see "SQL Server Reporting Services" in the following two menus, then the Reporting Services service has not been registered:

- SharePoint 2010 Central Administration -> "Application Management" -> "Manage Services on Server" page
- SharePoint 2010 Central Administration -> "Application Management" -> "Manage Service Applications" ->"New" menu

Workaround: To register and start the Reporting Services SharePoint Services, complete the following:

1. On the computer that runs SharePoint 2010 Central Administration
    a. Open the SharePoint 2010 Management Shell with administrator privileges. Right click the icon and click, "Run As Administrator". Run the following three cmdlets from the shell:

```
Install-SPRSService
```

```
Install-SPRSServiceProxy
```

```
Get-SPServiceInstance -all |where {$_.TypeName -like "SQL Server Reporting*"} | Start-
SPServiceInstance
```

2. Verify the Reporting Services Service shows status as "**Started**" on the page: SharePoint 2010 Central Administration -> "**Application Management**" -> "**Manage Services on Server**"

Reporting Services PowerShell cmdlets are not available and commands are not recognized

Description: When you try to run a Reporting Services PowerShell cmdlet, you see an error message similar to the following:

- The term 'Install-SPRSServiceInstall-SPRSService' **is not recognized** as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.At line:1 char:39+ Install-SPRSServiceInstall-SPRSService <<<< + CategoryInfo : ObjectNotFound: (Install-SPRSServiceInstall-SPRSService:String) [], CommandNotFoundExcep

Workaround: Complete one of the following:

- Run the Reporting Services add-in for SharePoint products. **rssharepoint.msi**.
- Install Reporting Services SharePoint mode from the SQL Server installation media.

Note: If the **SharePoint 2013 Management Shell** is open when you complete one of the workarounds, close and reopen the management shell.

For more information, see the following:

- Where to find the Reporting Services add-in for SharePoint Products
- Install Reporting Services SharePoint Mode for SharePoint 2010
- Install Reporting Services SharePoint Mode for SharePoint 2013

You see an error message indicating the URL is not configured

Description: You see and error message similar to the following:

This SQL Server Reporting Services (SSRS) functionality is not supported. Use Central Administration to verify and fix one or more of the following issues:•A report server URL is not configured. Use the SSRS Integration page to set it.•The SSRS service application proxy is not configured. Use the SSRS service application pages to configure the proxy.•The SSRS service application is not mapped to this web application. Use the SSRS service application pages to associate the SSRS service application proxy to the Application Proxy Group for this web application.

Workaround: The error message contains three suggested steps to correct this issue. The first suggestion in the message 'A report server URL is not configured..' is relevant when integrating with report server version previous to SQL Server 2012. SharePoint Configuration for the previous report server versions is completed on the **General Application Settings** page, using the **SQL Server Reporting Services (2008 and 2008 R2)**..

More Information: You will see this error message when attempting to use any of the Reporting Services functionality that requires a connection to the Reporting Services service. This includes:

- Opening SQL Server Report Builder from a SharePoint document library.
- Manage Subscriptions.
- Manage a service application.

Setup fails on a computer with SharePoint installed but it is not configured

Description: If you select to install Reporting Services SharePoint Mode on a computer that has SharePoint installed but SharePoint is not configured, you will see a message similar to the following and setup will stop:

SQL Server Setup has stopped working

Workaround: Configure SharePoint and then run SQL Server Installation.

More Information: When installing Reporting Services into and existing SharePoint installation, setup attempts to install and start the Reporting Services SharePoint service. If SharePoint is not configured, the service installation fails, causing setup to fail.

SharePoint Central Administration Page is blank

Description: You were able to successfully install SharePoint 2010, with no installation errors. However when you browse to Central Administration, you only see a blank page:

Workaround: This issue is not specific to Reporting Services but is related to the configuration of permissions in your overall SharePoint installation. The following is a list of suggested:

- Review the SharePoint topic on development environments. Setting Up the Development Environment for SharePoint 2010 on Windows Vista, Windows 7, and Windows Server 2008
- Review the forum post: Central Administration returns blank page after installation on Windows 7
- The Service account you are using for SharePoint services such as the SharePoint 2010 Central Administration Service, should have administrative privileges in the local operating system.

You see an error Message when you try to create a new Report Builder Report

Description: You see an error message similar to the following when you attempt to create a Report Builder report inside a document library:

This functionality is not supported because a SQL Server Reporting Services service application does not exist or a report server URL has not been configured in Central Administration.

Workaround: Verify you have an Reporting Services service application and it is correctly configured. For more information see, the section 'Create a Reporting Services Service Application' in Install Reporting Services SharePoint Mode for SharePoint 2010

You see an error message that RS_SHP is not supported with PREPAREIMAGE

Description: When you try to run PREPAREIMAGE for Reporting Services you see an error message similar to the following:

"The specified feature 'RS_SHP' is not supported when running the PREPAREIMAGE action, since it does not support SysPrep. Remove the features that are not compatible with SysPrep and run setup again."

Workaround: There is no work around. Reporting Services does not support SYSPREP (PREPAREIMAGE). Reporting Services Native mode does support SYSPREP.

Troubleshoot Problems with the Native Mode Installations

Performance counters are not visible after upgrading to Windows Vista or Windows Server 2008

If you upgrade the operating system to Windows Vista or Windows Server 2008 on a computer that runs Reporting Services, Reporting Services performance counters will not be set after the upgrade.

To reinstate Reporting Services performance counters

1. Delete the following registry keys:
   - HKLM\SYSTEM\CurrentControlSet\Services\MSRS 2011 Web Service
   - HKLM\SYSTEM\CurrentControlSet\Services\MSRS 2011 Windows Service
2. Open a command window and type the following command at the prompt:

- **run** < *.NET 2.0 Framework directory* >**\InstallUtil.exe** < *Report Server Bin directory* >**\ReportingServicesLibrary.dll**

  > **NOTE**
  > Replace <*.NET 2.0 Framework directory*> with the physical path of the .NET Framework 2.0 files and replace <*Report Server Bin directory*> with the physical path of the report server bin files.

3. Restart the Reporting Services service.

To verify that the steps worked, open a Web browser and navigate to the Report Manager URL or the Report Server URL. Then open Performance Monitor to verify that the counters are working.

To re-add the performance registry keys by using Registry Editor

1. Open the Registry Editor:
   a. Click **Start**, and click **Run**.
   b. In the **Run** dialog box, in the **Open** box, type `regedit`.
2. In Registry Editor, select the following registry key:
   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSRS 2011 Web Service\Performance`
3. Right-click the **Performance** node, point to **New**, and click **Multi-String Value**.
4. Type `Counter Names` and then press ENTER.
5. Repeat to add the `Counter Types` registry key in this node.
6. Navigate to the following registry key:
   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSRS 2011 Web Service\Performance`
7. Right-click the **Performance** node, point to **New**, and click **Multi-String Value**.
8. Type `Counter Names` and then press ENTER.
9. Repeat to add the `Counter Types` registry key in this node.

After you repair the 64-bit instance or re-add the registry keys manually, you can use Performance Monitor to configure the Reporting Services performance objects that you want to monitor.

ReportServerExternalURL and PassThroughCookies configuration properties are not configured after an upgrade from SQL Server 2005

When you upgrade from SQL Server 2005 to SQL Server 2014 Reporting Services (SSRS), the `ReportServerExternalURL` and `PassThroughCookies` configuration properties are not configured by the upgrade process. `ReportServerExternalURL` is an optional property, and it should be set only if you are using SharePoint 2.0 Web Parts and you want users to be able to retrieve a report and open it in a new browser window. For more information about `ReportServerExternalURL`, see URLs in Configuration Files (SSRS Configuration Manager). `PassThroughCookies` is required only when using Custom authentication method. For more information about `PassThroughCookies`, see Configure Report Manager to Pass Custom Authentication Cookies.

> **NOTE**
> When you use Custom authentication, it is recommended that you migrate your installation rather than performing an upgrade. For more information about migrating Reporting Services, see Migrate a Reporting Services Installation (Native Mode).

By default, these properties do not exist in the SQL Server 2014 Reporting Services (SSRS) configuration. If you configured these properties in SQL Server 2005 and you continue to require the functionality that they provide, you must manually add them to the **RSReportServer.config** file following the upgrade process. For more information, see Modify a Reporting Services Configuration File (RSreportserver.config).

Installation fails for a default instance of SQL Server 2005 Reporting Services on a computer that runs SQL Server 2012Reporting Services

If you attempt to install a default instance of SQL Server 2005 Reporting Services on a computer that already runs an instance of SQL Server 2014 Reporting Services (SSRS), the SQL Server 2005 Reporting Services instance will fail to install with the following error message:

"An instance with the same name is already installed on this computer. To proceed with SQL Server Setup, provide a unique instance name."

This issue happens regardless of whether the SQL Server 2014 Reporting Services (SSRS) instance is a default or named instance, and regardless of whether a SQL Server 2014 Reporting Services (SSRS) instance with that name already exists on the computer.

To work around this issue, you have one of the following options:

- If you must run SQL Server 2005 Reporting Services as the default instance on the computer, you must install the SQL Server 2005 Reporting Services instance before the SQL Server 2014 Reporting Services (SSRS) instance.
- If the SQL Server 2005 Reporting Services instance does not need to be a default instance, you can install the SQL Server 2005 Reporting Services instance as a named instance after you install the SQL Server 2014 Reporting Services (SSRS) instance.

401-Unauthorized error when using Windows authentication after an upgrade from SQL Server 2005 to SQL Server 2012

If you upgrade from SQL Server 2005 Reporting Services to SQL Server 2014 Reporting Services (SSRS), and you use NTLM authentication with a built-in account for the Report Server service account, you might encounter a 401-Unauthorized error when you access the report server or Report Manager after the upgrade.

This happens because of a change in the default SQL Server 2014 Reporting Services (SSRS) configuration for Windows authentication. Negotiate is configured when the Report Server service account is either Network Service or Local System. NTLM is configured when the Report Server service account is not one of those built-in accounts. To fix this issue after you upgrade, you can edit the RSReportServer.config file and configure the `AuthenticationType` to be `RSWindowsNTLM`. For more information, see Configure Windows Authentication on the Report Server.

Uninstalling 32-bit instance of SQL Server 2012 Reporting Services in side-by-side deployment with a 64-bit instance breaks the 64-bit instance

When you install a 32-bit instance and a 64-bit instance of SQL Server 2014 Reporting Services (SSRS) side-by-side on a computer, and you uninstall the 32-bit instance, four Reporting Services registry keys are removed. This breaks the 64-bit instance of Reporting Services. The Reporting Services registry keys that are removed when you uninstall the 32-bit instance are:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSRS 2011 Web Service\Performance:Counter Names
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSRS 2011 Windows Service\Performance:Counter Names
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSRS 2011 Web Service\Performance:Counter Types
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSRS 2011 Windows Service\Performance:Counter Types
```

To fix this issue, you can repair the 64-bit instance. Although it is recommended to use repair, you can re-add the registry keys manually by using Registry Editor.

Caution

Incorrectly editing the registry can severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

Additional Resources

The following are additional resources you can review to assist you with trouble shooting issues:

- TechNet Wiki: Trouble Shooting topics Troubleshoot SQL Server Reporting Services (SSRS) in SharePoint Integrated Mode
- Forum: SQL Server Reporting Services

Verify a PowerPivot for SharePoint Installation

5/22/2019 • 4 minutes to read • Edit Online

A PowerPivot for SharePoint instance that you install in a SharePoint farm is administered through SharePoint Central Administration. At a minimum, you can check pages in Central Administration and on SharePoint sites to verify that PowerPivot server components and features are available. However, to fully verify an installation, you must have a PowerPivot workbook that you can publish to SharePoint and access from a library. For testing purposes, you can publish a sample workbook that already contains PowerPivot data and use it to confirm that SharePoint integration is correctly configured.

Verify Central Administration Integration

To verify PowerPivot integration with Central Administration, do the following:

1. On the Start menu, click **All Programs**, open Microsoft SharePoint 2010 Products, and click **SharePoint 2010 Central Administration**.
2. Enter your user name and password, and then click **OK**.

   Optionally, you can modify browser settings to avoid having to enter a user name and password each time you open Central Administration. To add Central Administration as a trusted site, do the following.

   a. In Internet Explorer, on the Tools menu, click **Internet options**.

   b. On the Security tab, in the **Select a zone to view or change security settings** section, click Trusted Sites, and then click Sites.

   c. Clear the **Require server verification (https:) for all sites in this zone** checkbox.

   d. In **Add this Web site to the zone**, type the URL to your site, and then click **Add**.

   e. Click **Close**, and then click **OK**.

   > **NOTE**
   > SharePoint installation documentation includes additional instructions for working around proxy server errors and for disabling Internet Explorer Enhanced Security Configuration so that you can download and install updates. For more information, see the **Perform additional tasks** section in Deploy a single server with SQL Server on the Microsoft web site.

3. In Central Administration, in System Settings, click **Manage farm features**.
4. Verify that **PowerPivot Integration Feature** is **Active**.
5. In Central Administration, in System Settings, click **Manage services on server**.
6. Verify that **SQL Server Analysis Services** and **SQL Server PowerPivot System Service** are started.
7. In Central Administration, in Application Management, click **Manage service applications**.
8. Click **Default PowerPivot Service Application** to open PowerPivot Management Dashboard for this application. On first use, the dashboard takes several minutes to load.

   Alternatively, click the empty space next to **Default PowerPivot Service Application** to select the row, and click **Properties** to view the configuration settings for this service application. You can modify both

configuration settings and application properties to change your server configuration. For more information about these settings, see Create and Configure a PowerPivot Service Application in Central Administration.

Verify Integration at the Site Level

To verify PowerPivot integration with a SharePoint site, do the following:

1. In a browser, open the Web application you created. If you used default values, you can specify http://<your computer name> in the URL address.
2. Verify that PowerPivot data access and processing features are available in the application. You can do this by verifying the presence of PowerPivot-provided library templates:
   a. On Site Actions, click **More Options...**
   b. In Libraries, you should see **Data Feed Library** and **PowerPivot Gallery**. These library templates are provided by the PowerPivot feature and will be visible in the Libraries list if the feature is integrated correctly.

Verify Data Access on the Server

To verify PowerPivot data access on the server, do the following:

1. Download the Picnic data sample that accompanies a Reporting Services tutorial. You will use the sample workbook in this download to verify PowerPivot data access. Extract the files.
2. Upload the Excel workbook (.xlsx) to Shared Documents. The workbook contains embedded PowerPivot data.
3. Click on the document to open it from the library.
4. Click on a slicer or filter at the top of the workbook. Month, color, and type are slicers in this workbook. Clicking a slicer starts a PowerPivot query and proves that your server is operational. The server will load PowerPivot data in the background and return the results.
5. Go back to the library. Select the down arrow to the right of the workbook, and then click **Launch Power View**. This step confirms that the Power View feature in Reporting Services is operational. If you did not install Reporting Services, skip this step.

   In the next step, you will connect to the server in Management Studio to verify the data is loaded and cached.
6. Start SQL Server Management Studio from the SQL Server 2012 program group in the Start menu. If this tool is not installed on your server, you can skip ahead to the last step to confirm the presence of cached files.
7. In Server Type, select **Analysis Services**.
8. In Server Name, enter **<server-name>\powerpivot**, where **<server-name>** is the name of the computer that has the PowerPivot for SharePoint installation.
9. Click **Connect**. This verifies that the Analysis Services server is available.
10. In Object Explorer, you can click **Databases** to view the list of PowerPivot data files that are loaded.
11. On the computer file system, check the following folder to determine whether files are cached to disk. The presence of cached files is further verification that your deployment is operational. To view the file cache, go to the <drive>:\Program Files\Microsoft SQL Server\MSAS11.POWERPIVOT\OLAP\Backup\Sandboxes\Default PowerPivot Service Application folder. Each cached database is stored in its own folder, using a GUID-based naming convention to ensure a unique name.

Troubleshoot a PowerPivot for SharePoint Installation

10/3/2019 • 2 minutes to read • Edit Online

If you get errors instead of the pages and features you expect, do the following.

- Review release notes for both SharePoint and SQL Server 2014 to get workarounds for known installation problems. Release notes are provided with the installation media or on the Microsoft site from which you downloaded the software.
  - SQL Server 2014 Release Notes.
- See the Technet wiki topic, Troubleshooting Installations of PowerPivot (and other add-ins).

Issues

PowerPivot Gallery Thumbnail images show as a red X

One Possible cause is the **PowerPivot features Integration for Site Collections** is not active. Complete the following:

1. In the PowerPivot Gallery library, click **Site Settings** from either the gear icon or the **Home** list.
2. In the **Site Collection Administration** section, click **Site Collection Features**.
3. Click **Site Collection Features**.
4. Verify **PowerPivot features Integration for Site Collections** is **Active**.

For additional causes of this issue, see PowerPivot Gallery shows Red X's for Icons (https://support.microsoft.com/kb/2361559).

Upgrade to SQL Server 2014

3/12/2020 • 2 minutes to read • Edit Online

You can upgrade instances of, SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, or SQL Server 2012 to SQL Server 2014. Before running SQL Server Setup to upgrade to SQL Server 2014, check out the SQL Server 2014 Upgrade Technical Guide (PDF download), review the topics about the upgrade process in this section, and read the SQL Server 2014 Release Notes.

In This Section

This section contains the following topics:

- Supported Version and Edition Upgrades
- Use Upgrade Advisor to Prepare for Upgrades
- Use the Distributed Replay Utility to Prepare for Upgrades
- Upgrade Analysis Services
- Upgrade Database Engine
- Upgrade Data Quality Services
- Upgrade Integration Services
- Upgrade Master Data Services
- Upgrade PowerPivot for SharePoint
- Upgrade Replicated Databases
- Upgrade and Migrate Reporting Services
- Upgrade SQL Server Management Tools
- Upgrade How-to Topics

See Also

Upgrade Database Engine
Upgrade Analysis Services
Upgrade and Migrate Reporting Services
Upgrade Integration Services
Upgrade Replicated Databases
Upgrade Master Data Services
SQL Server 2012 Best Practices Analyzer
SQL Server 2008 R2 Best Practices Analyzer
Backward Compatibility

Supported Version and Edition Upgrades

11/5/2018 • 9 minutes to read • Edit Online

You can upgrade from SQL Server 2008, and SQL Server 2008 R2, and SQL Server 2012. This topic lists the supported upgrade paths from these SQL Server versions, and the supported edition upgrades for SQL Server 2014.

Pre upgrade Checklist

- Before upgrading from one edition of SQL Server 2014 to another, verify that the functionality you are currently using is supported in the edition to which you are moving.
- 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.
- To upgrade to SQL Server 2014, you must be running a supported operating system. For more information, see Hardware and Software Requirements for Installing SQL Server 2014.
- Upgrade will be blocked if there is a pending restart.
- Upgrade will be blocked if the Windows Installer service is not running.

Unsupported Scenarios

- Cross-version instances of SQL Server 2014 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 2014.
- Cross-platform upgrade is not supported. You cannot upgrade a 32-bit instance of SQL Server to native 64-bit using SQL Server Setup. However, you can back up or detach databases from a 32-bit instance of SQL Server, and then restore or attach them to a new instance of SQL Server (64-bit) if the databases are not published in replication. You must re-create any logins and other user objects in master, msdb, and model system databases.
- You cannot add new features during the upgrade of your existing instance of SQL Server. After you upgrade an instance of SQL Server to SQL Server 2014, you can add features by using the SQL Server 2014 Setup. For more information, see Add Features to an Instance of SQL Server 2014 (Setup).
- Failover Clusters are not supported in WOW mode.
- Upgrade from an Evaluation edition of a previous SQL Server version is not supported.

Upgrades from Earlier Versions to SQL Server 2014

> **NOTE**
> Support for SQL Server 2005 is described in more detail in the next section, 'SQL Server 2014 support for SQL Server 2005'.

- SQL Server 32-bit editions can be upgraded to SQL Server 2014 on the 32-bit subsystem (WOW64) of a 64-bit server.

- SQL Server 64-bit versions can be upgraded to SQL Server 2014 64-bit server only.

> **NOTE**
>
> When you upgrade to SQL Server 2014 from a prior version of SQL Server Enterprise edition, choose between Enterprise Edition: Core-based Licensing and Enterprise Edition. These Enterprise editions differ only with respect to the licensing modes and the maximum number of cores supported. For more information, see Compute Capacity Limits by Edition of SQL Server.

SQL Server 2014 supports upgrade from the following versions of SQL Server:

- SQL Server 2005 SP4 or later
- SQL Server 2008 SP3 or later
- SQL Server 2008 R2 SP2 or later
- SQL Server 2012 SP1 or later

The table below lists the supported upgrade scenarios from earlier versions of SQL Server to SQL Server 2014.

UPGRADE FROM	SUPPORTED UPGRADE PATH
SQL Server 2005 SP4 Enterprise	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence
SQL Server 2005 SP4 Developer	SQL Server 2014 Developer
SQL Server 2005 SP4 Standard	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard
SQL Server 2005 SP4 Workgroup	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard
SQL Server 2005 SP4 Express, SQL Server 2005 SP4 Express with Tools, and SQL Server 2005 SP4 Express with Advanced Services	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard SQL Server 2014 Web SQL Server 2014 Express
SQL Server 2008 SP3 Enterprise	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence
SQL Server 2008 SP3 Developer	SQL Server 2014 Developer

UPGRADE FROM	SUPPORTED UPGRADE PATH
SQL Server 2008 SP3 Standard	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard
SQL Server 2008 SP3 Small Business	SQL Server 2014 Standard
SQL Server 2008 SP3 Web	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard SQL Server 2014 Web
SQL Server 2008 SP3 Workgroup	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard
SQL Server 2008 SP3 Express, SQL Server 2008 SP3 Express with Tools, and SQL Server 2008 SP3 Express with Advanced Services	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard SQL Server 2014 Web SQL Server 2014 Express
SQL Server 2008 R2 SP2 Datacenter	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence
SQL Server 2008 R2 SP2 Enterprise	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence
SQL Server 2008 R2 SP2 Developer	SQL Server 2014 Developer
SQL Server 2008 R2 SP2 Small Business	SQL Server 2014 Standard
SQL Server 2008 R2 SP2 Standard	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard
SQL Server 2008 R2 SP2 Web	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard SQL Server 2014 Web

UPGRADE FROM	SUPPORTED UPGRADE PATH
SQL Server 2008 R2 SP2 Workgroup	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard
SQL Server 2008 R2 SP2 Express, SQL Server 2008 R2 SP2 Express with Tools, and SQL Server 2008 R2 SP2 Express with Advanced Services	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard SQL Server 2014 Web SQL Server 2014 Express
SQL Server 2012 SP1 Enterprise	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence
SQL Server 2012 SP1 Developer	SQL Server 2014 Developer
SQL Server 2012 SP1 Standard	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard
SQL Server 2012 SP1 Web	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard SQL Server 2014 Web
SQL Server 2012 SP1 Express, SQL Server 2012 SP1 Express with Tools, and SQL Server 2012 SP1 Express Management Studio, and SQL Server 2012 SP1 Express with Advanced Services	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence SQL Server 2014 Standard SQL Server 2014 Web SQL Server 2014 Express
SQL Server 2012 SP1 Business Intelligence	SQL Server 2014 Enterprise SQL Server 2014 Business Intelligence

SQL Server 2014 Support for SQL Server 2005

This section discusses SQL Server 2014 support for SQL Server 2005. In SQL Server 2014, you will be able to do the following:

- Upgrade a SQL Server 2005 instance of database engine to SQL Server 2014 by running SQL Server 2014 setup using the installation wizard or from the command prompt.

- Attach a SQL Server 2005 database (mdf/ldf files) to SQL Server 2014 instance of database engine.
- Restore a SQL Server 2005 database to SQL Server 2014 instance of database engine from a backup.
- Upgrade a SQL Server 2005 Integration Services (SSIS) package to SQL Server 2014. Execute packages with automatic-in-place upgrade.
- Upgrade a SQL Server 2005 Analysis Services (SSAS) to SQL Server 2014 by running SQL Server 2014 setup.
- Back up a SQL Server 2005 Analysis Services (SSAS) cube and restoring on SQL Server 2014.
- Upgrade SQL Server 2005 Reporting Services (SSRS) to SQL Server 2014 by running SQL Server 2014 setup.
- Connect to SQL Server 2005by using SQL Server Profiler 2014.

When a SQL Server 2005 database is upgraded to SQL Server 2014, the database compatibility level will be changed from 90 to 100. (In SQL Server 2014, valid values for the database compatibility level are 100, 110 and 120.) ALTER DATABASE Compatibility Level (Transact-SQL) discusses how the compatibility level change could affect SQL Server applications.

Any scenarios not specified in the list above are not supported, including but not limited to the following:

- Installing SQL Server 2005 and SQL Server 2014 on same computer (side by side).
- Using a SQL Server 2005 instance as a member of the replication topology that involves a SQL Server 2014 instance.
- Configuring database mirroring between SQL Server 2014 and SQL Server 2005 instances.
- Backing up the transaction log with log shipping between SQL Server 2014 and SQL Server 2005 instances.
- Configuring linked servers between SQL Server 2014 and SQL Server 2005 instances.
- Managing a SQL Server 2005 instance from a SQL Server 2014 Management Studio.
- Attaching a SQL Server 2005 Analysis Services (SSAS) cube in SQL Server 2014 Management Studio.
- Connecting to SQL Server 2005 Integration Services (SSIS) from SQL Server 2014 Management Studio.
- Managing a SQL Server 2005 Integration Services (SSIS) service from SQL Server 2014 Management Studio.
- Support for SQL Server 2005 third party custom Integration Services components, such as execute and upgrade.

SQL Server 2014 Edition Upgrade

The following table lists the supported edition upgrade scenarios in SQL Server 2014.

For step-by-step instructions on how to perform an edition upgrade, see Upgrade to a Different Edition of SQL Server 2014 (Setup).

UPGRADE FROM	UPGRADE TO
SQL Server 2014 Enterprise (Server+CAL and Core) [2]	SQL Server 2014 Business Intelligence
SQL Server 2014 Business Intelligence	SQL Server 2014 Enterprise (Server+CAL or Core License)

UPGRADE FROM	UPGRADE TO
SQL Server 2014 Evaluation Enterprise [2]	SQL Server 2014 Enterprise (Server+CAL or Core License) SQL Server 2014 Business Intelligence SQL Server 2014 Standard SQL Server 2014 Developer SQL Server 2014 Web Upgrading from Evaluation Enterprise (a free edition) to any of the paid editions is supported for stand-alone installations, but is not supported for clustered installations.
SQL Server 2014 Standard [2]	SQL Server 2014 Business Intelligence SQL Server 2014 Enterprise (Server+CAL or Core License)
SQL Server 2014 Developer [2]	SQL Server 2014 Enterprise (Server+CAL or Core License) SQL Server 2014 Business Intelligence SQL Server 2014 Web SQL Server 2014 Standard
SQL Server 2014 Web	SQL Server 2014 Enterprise (Server+CAL or Core License) SQL Server 2014 Business Intelligence SQL Server 2014 Standard
SQL Server 2014 Express [1]	SQL Server 2014 Enterprise (Server+CAL or Core License) SQL Server 2014 Business Intelligence SQL Server 2014 Developer SQL Server 2014 Standard SQL Server 2014 Web

Additionally you can also perform an edition upgrade between SQL Server 2014 Enterprise (Server+CAL license) and SQL Server 2014 Enterprise (Core License):

EDITION UPGRADE FROM	EDITION UPGRADE TO
SQL Server 2014 Enterprise (Server+CAL License) [2]	SQL Server 2014 Enterprise (Core License)
SQL Server 2014 Enterprise (Core License)	SQL Server 2014 Enterprise (Server+CAL License)

[1] Also applies to SQL Server 2014 Express with Tools and SQL Server 2014 Express with Advanced Services.

[2] Changing the edition of a SQL Server 2014 failover cluster is limited. The following scenarios are not supported for SQL Server 2014 failover clusters:

- SQL Server 2014 Enterprise to SQL Server 2014 Developer, Standard, or Enterprise Evaluation.

- SQL Server 2014 Developer to SQL Server 2014 Standard or Enterprise Evaluation.
- SQL Server 2014 Standard to SQL Server 2014 Enterprise Evaluation.
- SQL Server 2014 Enterprise Evaluation to SQL Server 2014 Standard.

See Also

Features Supported by the Editions of SQL Server 2014
Hardware and Software Requirements for Installing SQL Server 2014
Upgrade to SQL Server 2014
Use Upgrade Advisor to Prepare for Upgrades

Are you upgrading from SQL Server 2005?

5/8/2020 • 4 minutes to read • Edit Online

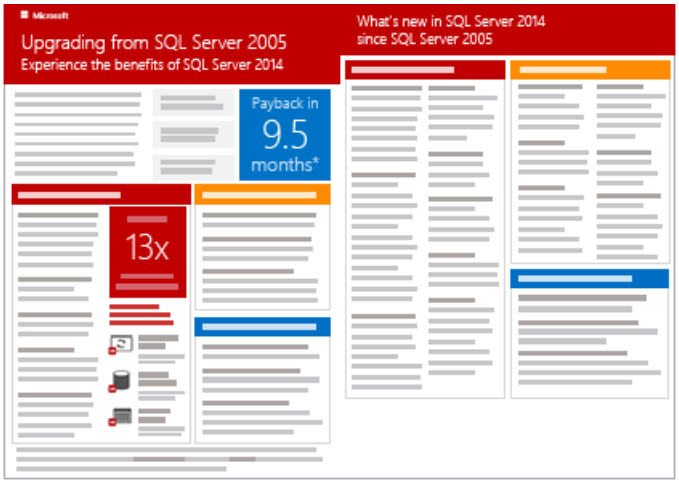
The end of extended support for SQL Server 2005 is one reason to upgrade now to a newer version of SQL Server and to Azure SQL Database. Upgrading enables you to maintain security and compliance, achieve breakthrough performance, and optimize your data platform infrastructure.

For more info, guidance, and tools to plan and automate your upgrade or migration, see SQL Server 2005 end of support.

Why upgrade?

> **IMPORTANT**
>
> Extended support for SQL Server 2005 ends on April 12, 2016. If you're still running SQL Server 2005 after April 12, 2016, you'll no longer receive security updates.

To get the data sheet in PDF format about upgrading from SQL Server 2005, click here (not on the thumbnail image below).



Choose your upgrade option

If you're upgrading relational databases from SQL Server 2005, here are your options for relational storage on the Microsoft platform.

To see a more comprehensive analysis of these options, click here.

RELATIONAL STORAGE OPTION	BENEFITS	OTHER FACTORS TO CONSIDER
SQL Server on premises Consider this option for database applications of any kind, from transactional systems to data warehouses. For more info, see SQL Server 2014.	You have the most control over features and scalability because you manage both hardware and software. If you're upgrading from SQL Server 2005, this is the most similar environment.	You have to make the biggest up-front investment and provide the most ongoing management, because you have to buy, maintain, and manage your own hardware and software.

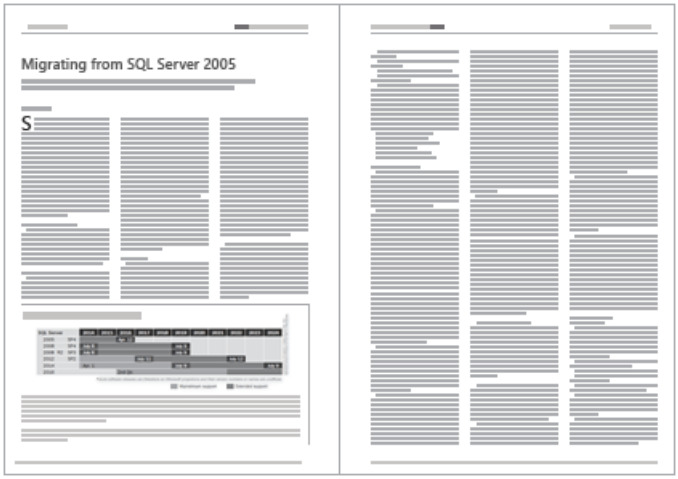
RELATIONAL STORAGE OPTION	BENEFITS	OTHER FACTORS TO CONSIDER
SQL Server hosted on Azure virtual machines Consider this option if you want the following things. -Benefits of migrating to a hosted environment. -Control over the operating environment. -Familiar feature set of SQL Server. For more info, see SQL Server on Azure Virtual Machines overview. For info about migrating, see Migrate a database to SQL Server on an Azure VM.	You can deploy quickly from a library of virtual machine images. You get the full SQL Server feature set. You save the cost of hardware and of server software. You pay only for hourly usage.	You have to configure and manage both the SQL Server and the operating system software.
Azure SQL Database hosted database service Consider this option if you want a lower-cost solution with less maintenance. This option is particularly well suited for apps that don't require the same capacity at all times, or that have to provide external access. For more info, see SQL Database. For info about migrating, see Migrating a SQL Server database to Azure SQL Database.	You can deploy quickly and scale up easily. You pay only for hourly usage. The cost of the service includes not only storage, but high availability and automated backups.	Azure SQL Database lacks some SQL Server features that are not applicable in a hosted cloud environment. For more info, see Azure SQL Database Transact-SQL information. Azure SQL Database also has a maximum database size of 500 GB, compared to 524 PB for SQL Server.

You may also want to consider a non-relational or NoSQL solution for certain data and applications.

NON-RELATIONAL SOLUTION	BENEFITS
Azure DocumentDB Consider this option for modern, scalable, mobile and web applications that use JSON data and require a combination of robust querying and transactional data processing. For more info, see DocumentDB.	Your documents are indexed and you can use familiar SQL syntax to query them. The database is schema-free. You can add properties to documents without having to rebuild indexes. You get JSON and JavaScript support right inside the database engine. You get native support for geospatial data and integration with other Azure Services including Azure Search, HDInsight, and Data Factory. You get low latency, high performance storage with reserved throughput levels.

NON-RELATIONAL SOLUTION	BENEFITS
Azure table storage Consider this option to store petabytes of semi-structured data in a cost-effective solution. For more info, see Table Storage.	You can evolve your apps and your table schema without taking the data offline. You can scale up without sharding your dataset. You get geo-redundant storage that replicates data across multiple regions.

To download the report "Migrating from SQL Server 2005" by Directions on Microsoft, which contains more details about the upgrade options, click here (not on the thumbnail image below).



Plan your upgrade

- Read about how to plan your upgrade in the following series of blog posts from the SQL Server team.
  - Planning an efficient upgrade from SQL Server 2005: Step 1 of 3
  - Planning an efficient upgrade from SQL Server 2005: Step 2 of 3
  - Planning an efficient upgrade from SQL Server 2005: Step 3 of 3
- Review the requirements and considerations under Planning a SQL Server Installation, including the Hardware and Software Requirements for Installing SQL Server 2014.
- Read about how to upgrade.
  - Review the available upgrade methods and learn how to plan and test in the topic Upgrade Database Engine.

    > **IMPORTANT**
    > You can't upgrade a SQL Server 2005 server to a SQL Server 2014 server in place. You have to install SQL Server 2014, then migrate your SQL Server 2005 databases to the new installation.

  - To get the more detailed "Technical Upgrade Guide" in PDF format, click here.
- For more info, guidance, and tools to plan and automate your upgrade or migration, see SQL Server 2005 end of support.

Get SQL Server 2014

To download an evaluation copy of SQL Server 2014, click here.

See Also

SQL Server 2014
SQL Server 2005 end of support
Upgrade from SQL Server 2005 to SQL Server 2016

Use Upgrade Advisor to Prepare for Upgrades

11/5/2019 • 2 minutes to read • Edit Online

SQL Server Upgrade Advisor helps you prepare for upgrades to SQL Server 2014. 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.

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.

Upgrade Advisor Analysis

Upgrade Advisor analyzes the following SQL Server components:

- Database Engine
- Analysis Services
- Reporting Services
- Integration Services

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.

> **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.

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 Supported Version and Edition Upgrades. If you are scanning instances of Reporting Services, you must install Upgrade

Advisor on the report server.

Upgrade Advisor is available in a feature pack.

Prerequisites for installing and running Upgrade Advisor are as follows:

- Windows Server 2008 SP2, Windows 7 SP1, and Windows Server 2008 R2 SP1.
- Windows Installer beginning with version 4.5. You can install Windows Installer from the Windows Installer Web site.
- Microsoft .NET Framework 4. .NET Framework 4 is available on the SQL Server 2014 product media, and from the .NET Framework 4 download page.
    - To install the .NET Framework 4 from the SQL Server 2014 media, locate the root of the disk drive. Then, double-click the \redist folder, double-click the DotNetFrameworks folder, and run dotNetFx40_Full_x86_x64.exe (for 32-bit operating systems or for 64-bit operating systems).

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 disk.

After you install Upgrade Advisor, you can open it from the **Start** menu:

- Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2014**, and then click **SQL Server 2014 Upgrade Advisor**.

For more information, see the Upgrade Advisor documentation included in the Upgrade Advisor download and the SQL Server 2014 Release Notes.

See Also

Work with Multiple Versions and Instances of SQL Server
Supported Version and Edition Upgrades
Backward Compatibility

Use the Distributed Replay Utility to Prepare for Upgrades

5/22/2019 • 2 minutes to read • Edit Online

The MicrosoftSQL Server Distributed Replay Utility can use multiple computers to replay trace data, simulating a mission-critical workload. By performing a replay on a test server before and after a SQL Server upgrade, you can measure performance differences and look for any incompatibilities your application may have with the upgrade. For more information, see SQL Server Distributed Replay.

See Also

Administration Tool Command-line Options (Distributed Replay Utility)

Upgrade Analysis Services

8/8/2019 • 2 minutes to read • Edit Online

Use SQL Server Setup to upgrade SQL Server. For detailed information on upgrading SQL Server in SharePoint mode, see Upgrade PowerPivot for SharePoint. For more information about upgrading an existing SQL Server instance, see Upgrade to SQL Server 2014 Using the Installation Wizard (Setup).

Known Upgrade Issues

Before upgrading to SQL Server 2014 Analysis Services, review the following:

- SQL Server 2014 Release Notes.
- To learn which Analysis Services features and functionality have been discontinued, deprecated, or changed see Analysis Services Backward Compatibility.

Pre-Upgrade Checklist

Before upgrading, review the following information:

- Supported Version and Edition Upgrades
- Hardware and Software Requirements for Installing SQL Server 2014
- Check Parameters for the System Configuration Checker
- Security Considerations for a SQL Server Installation
- Backup and Restore of Analysis Services Databases
- Use Upgrade Advisor to Prepare for Upgrades

Upgrading Analysis Services

You can choose from several approaches to upgrade server and data:

- An **in-place upgrade** replaces the existing program files with SQL Server 2014 program files. Databases remain in the same location. Program folders are updated to reflect the new name.
- A **side-by-side upgrade** is a new installation of SQL Server 2014 on the same computer that has an existing Analysis Services instance. You can move databases over to the new instance on the same computer, and then uninstall the old version if you no longer use it.
- You can also install Analysis Services on new hardware and then migrate existing databases to that server.

In-place Upgrade

You can upgrade an existing instance of Analysis Services to SQL Server 2014 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, run Setup and specify the name of the existing instance as the name of the new instance.

Upgrading Databases

Databases that were created in previous versions of Analysis Services run on the upgraded server under an older database compatibility level setting. Databases created in the following versions, have a database compatibility level of 105. You can change the compatibility level if you want to use features that require a newer database compatibility level. Otherwise, you can run the databases on the upgraded server using the original settings. For more information, see Set the Compatibility Level of a Multidimensional Database (Analysis Services).

- SQL Server 2012
- SQL Server 2008
- SQL Server 2008 R2
- SQL Server 2005

See Also

Features Supported by the Editions of SQL Server 2014
Planning a SQL Server Installation
Understanding Microsoft OLAP Architecture
Upgrade PowerPivot for SharePoint
Install Analysis Services in Multidimensional and Data Mining Mode
PowerPivot for SharePoint 2010 Installation

Upgrade Database Engine

11/5/2018 • 7 minutes to read • Edit Online

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.

Known Upgrade issues

Before upgrading the Database Engine, review SQL Server Database Engine Backward Compatibility. For information about supported upgrade scenarios and upgrade known issues, see Supported Version and Edition Upgrades. For backward compatibility content for other SQL Server components, see Backward Compatibility.

> **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.

> **NOTE**
> When you upgrade to SQL Server 2014 from a prior version of SQL Server Enterprise edition, choose between Enterprise Edition: Core-based Licensing and Enterprise Edition. These Enterprise editions differ only with respect to the licensing modes. For more information, see Compute Capacity Limits by Edition of SQL Server.

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 ServerIntegration Services Import and Export Wizard, and bulk export/bulk import methods.

Before upgrading the Database Engine, review the following:

- Review Hardware and Software Requirements for Installing SQL Server 2014.
- Review Check Parameters for the System Configuration Checker.
- Review Security Considerations for a SQL Server Installation.
- Review Supported Version and Edition Upgrades.

- Review Use Upgrade Advisor to Prepare for Upgrades.
- Review Use the Distributed Replay Utility to Prepare for Upgrades.
- Review SQL Server Database Engine Backward Compatibility.
- Review Migrate Query Plans.

Review the following issues and make changes before you upgrade SQL Server:

- When upgrading instances of SQL Server where SQL Server Agent is enlisted in MSX/TSX relationships, upgrade target servers before you upgrade master servers. If you upgrade master servers before target servers, SQL Server Agent will not be able to connect to master instances of SQL Server.
- When upgrading from a 64-bit edition of SQL Server to a 64-bit edition of SQL Server 2014, 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 Hardware and Software Requirements for Installing SQL Server 2014.
- 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.
- Make sure that Replication is current and then stop Replication.
- 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 use Database Mirroring, see Minimize Downtime for Mirrored Databases When Upgrading Server Instances.

Upgrading the Database Engine

You can overwrite an installation of SQL Server 2005 or later 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.

> **WARNING**
> When running the SQL Server 2014 setup program, the SQL Server instance is stopped and restarted as part of running the pre-upgrade checks.

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.

Database Compatibility Level After Upgrade

The compatibility levels of the `tempdb`, `model`, `msdb` and **Resource** databases are set to 120 after upgrade. The `master` system database retains the compatibility level it had before upgrade.

If the compatibility level of a user database was 100 or higher before the upgrade, it remains the same after upgrade. If the compatibility level was 90 before upgrade, in the upgraded database, the compatibility level is set to 100, which is the lowest supported compatibility level in SQL Server 2014.

> **NOTE**
> New user databases will inherit the compatibility level of the `model` database.

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. For more information, see Copy Databases with Backup and Restore or Database Detach and Attach (SQL Server).

> **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 Use the Copy Database Wizard.

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 Register Servers.
- Re-populate full-text catalogs to ensure semantic consistency in query results.

  SQL Server 2014 installs new word breakers for use by Full-Text and Semantic Search. The word breakers are used both at indexing time and at query time. If you do not rebuild the full-text catalogs, your search results may be inconsistent. If you issue a full-text query that looks for a phrase that is broken differently by the word breaker in a previous version of SQL Server and the current word breaker, a document or row containing the phrase might not be retrieved. This is because the indexed phrases were broken using different logic than the query is using. The solution is to repopulate (rebuild) the full-text catalogs with the new word breakers so that index time and query time behavior are identical.

  For more information, see sp_fulltext_catalog (Transact-SQL).

- 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 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 2014. We recommend the following procedures after you

upgrade to SQL Server 2014.

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.

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;
```

To resume full-text index population, run the following statement:

```
EXEC sp_fulltext_service 'pause_indexing', 0;
```

See Also

Supported Version and Edition Upgrades
Work with Multiple Versions and Instances of SQL Server
Backward Compatibility
Upgrade Replicated Databases

Migrate Query Plans

12/13/2018 • 2 minutes to read • Edit Online

In most cases, upgrading a database to the most recent version of SQL Server 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:

1. 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 the newer version of SQL Server.

   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.

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. XML-formatted 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.

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;
GO
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, DEFAULT) AS qp
    WHERE st.text LIKE N'SELECT City, StateProvinceID, PostalCode FROM Person.Address ORDER BY PostalCode
DESC;%';
GO
```

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.Address ORDER BY PostalCode DESC;',
@type = N'SQL',
@module_or_batch = NULL,
@params = NULL,
@hints = N'OPTION(USE PLAN N''<ShowPlanXML
xmlns=''''https://schemas.microsoft.com/sqlserver/2004/07/showplan''''
    Version=''''0.5'''' Build=''''9.00.1116''''>
    <BatchSequence><Batch><Statements><StmtSimple>
    ...
    </StmtSimple></Statements></Batch>
    </BatchSequence></ShowPlanXML>'')';
GO
```

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.

See Also

sp_create_plan_guide (Transact-SQL)
Query Hints (Transact-SQL)
Plan Guides

Upgrade Data Quality Services

12/13/2018 • 3 minutes to read • Edit Online

This topic provides information on how to upgrade your existing installation of Data Quality Services (DQS) to SQL Server 2014 CTP2. As part of upgrading your Data Quality Server in DQS to SQL Server 2014, you must also upgrade the DQS databases schema.

IMPORTANT

- You must back up your DQS databases before upgrading DQS to prevent any accidental data loss during the schema upgrade. For information about backing up DQS databases, see Backing Up and Restoring DQS Databases.
- You can connect to the SQL Server 2014 version of Data Quality Server by using the current or an earlier version of Data Quality Client or the DQS Cleansing Transformation in Integration Services to perform your data quality tasks.
- You can continue using the SQL Server 2012 SP1 version of Master Data Services Add-In for Excel after upgrading Data Quality Services and Master Data Services to SQL Server 2014 CTP2. However, any earlier version of the Master Data Services Add-In for Excel will not work after upgrading to SQL Server 2014 CTP2. You can download the SQL Server 2012 SP1 version of Master Data Services Add-In for Excel from here.

Prerequisites

- You must be logged on as a member of the Administrators group on the Data Quality Server computer.
- Your Windows user account must be a member of the sysadmin fixed server role in the SQL Server instance where Data Quality Server is installed.

Upgrading DQS

To upgrade DQS:

1. Back up your DQS databases before you start the upgrade process. For information about backing up DQS databases, see Backing Up and Restoring DQS Databases.
2. Upgrade the SQL Server instance where DQS is installed to SQL Server 2014.
   a. Run the SQL Server 2014 Setup wizard.
   b. In the left pane, click **Installation**.
   c. In the right pane, click **Upgrade from SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 or SQL Server 2012**.
   d. Complete the Setup wizard.

   NOTE

   This will upgrade your SQL Server instance to SQL Server 2014 and also install the latest Data Quality Client, if Data Quality Client was previously installed on this computer. If you have Data Quality Client installed on other computers, you must run the sub steps in Step 2 on those computers to install the current version of Data Quality Client. The Setup wizard installs the current version of Data Quality Client alongside the existing version of Data Quality Client. After you upgrade the DQS databases schema, you can connect to the SQL Server 2014 version of Data Quality Server by using the current or an earlier version of Data Quality Client.

3. Upgrade the DQS databases schema.

   a. Start Command Prompt as an administrator.

   b. At the command prompt, change your directory to the location where DQSInstaller.exe is available. For the default instance of SQL Server, the DQSInstaller.exe file is available at C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn:

   ```
   cd C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn
   ```

   c. At the command prompt, type the following command, and press ENTER:

   ```
   dqsinstaller.exe -upgrade
   ```

   d. The installer prompts you for backing up the DQS databases before proceeding. If you have already backed up your DQS databases, type `Y` or `Yes`, and then press ENTER to continue with the upgrade.

   e. A completion message is displayed after successful upgrade of the DQS databases schema.

Verifying the DQS Databases Schema Upgrade

To verify that DQS databases schema have successfully upgraded, you can check the current version in the DQS_MAIN and DQS_PROJECTS databases by querying the A_DB_VERSION table in each database. To do so:

1. Start SQL Server Management Studio, and connect to the SQL Server instance that contains the upgraded DQS databases schema.

2. Run the following query:

   ```
   SELECT * FROM DQS_MAIN.dbo.A_DB_VERSION WHERE STATUS=2;
   SELECT * FROM DQS_PROJECTS.dbo.A_DB_VERSION WHERE STATUS=2;
   ```

3. The output will display an entry for each upgrade along with the date of the upgrade. The maximum VERSION_ID and ASSEMBLY_VERSION on the latest date is the current version. A value of 2 in the STATUS column indicates success. If an error has occurred, the error will be listed in the ERROR column. A sample output:

ID	UPGRADE_DATE	VERSION_ID	ASSEMBLY_VERSION	USER_NAME	STATUS	ERROR
1000	2013-08-11 05:26:39.567	1200	11.0.3000.0	<DOMAIN\UserName>	2	
1001	2013-09-19 15:09:37.750	1600	12.0.xxxx.0	<DOMAIN\UserName>	2	

See Also

Install Data Quality Services
Remove Data Quality Server Objects
Upgrade to SQL Server 2014

Upgrade Integration Services

12/13/2018 • 9 minutes to read • Edit Online

If SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) is currently installed on your computer, you can upgrade to SQL Server 2014 Integration Services (SSIS).

When you upgrade to SQL Server 2014 Integration Services (SSIS) on a machine that has one of these earlier versions of Integration Services installed, SQL Server 2014 Integration Services (SSIS) is installed side-by-side with the earlier version.

With this side-by-side install, multiple versions of dtexec utility are installed. To ensure that you run the correct version of the utility, at the command prompt run the utility by entering the full path (<drive>:\Program Files\Microsoft SQL Server\<version>\DTS\Binn). For more information about dtexec, see dtexec Utility.

> **NOTE**
> In previous versions of SQL Server, by default when you installed SQL Server all users in the Users group had access to the Integration Services service. When you install SQL Server 2014, users do not have access to the Integration Services service. The service is secure by default. After SQL Server 2014 is installed, the SQL Server administrator must run the DCOM Configuration tool (Dcomcnfg.exe) to grant specific users access to the Integration Services service. For more information, see Grant Permissions to Integration Services Service.

Before Upgrading Integration Services

We recommended that you run Upgrade Advisor before you upgrade to SQL Server 2014. Upgrade Advisor reports issues that you might encounter if you migrate existing Integration Services packages to the new package format that SQL Server 2014 uses. For more information, see Use Upgrade Advisor to Prepare for Upgrades.

> **NOTE**
> Support for migrating or running Data Transformation Services (DTS) packages has been discontinued in the current release of SQL ServerIntegration Services. The following DTS functionality has been discontinued.
>
> - DTS runtime
> - DTS API
> - Package Migration Wizard for migrating DTS packages to the next version of Integration Services
> - Support for DTS package maintenance in SQL Server Management Studio
> - Execute DTS 2000 Package task
> - Upgrade Advisor scan of DTS packages.
>
> For information about other discontinued features, see Discontinued Integration Services Functionality in SQL Server 2014.

Upgrading Integration Services

You can upgrade by using one of the following methods:

- Run SQL Server 2014 Setup and select the option to **Upgrade from SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2**, or **SQL Server 2012**.
- 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 Install SQL Server 2014 from the Command

Prompt.

You cannot use upgrade to perform the following actions:

- Reconfigure an existing installation of Integration Services.
- Move from a 32-bit to a 64-bit version of SQL Server or from a 64-bit version to a 32-bit version.
- Move from one localized version of SQL Server to another localized version.

When you upgrade, you can upgrade both Integration Services and the Database Engine, or just upgrade the Database Engine, or just upgrade Integration Services. If you upgrade only the Database Engine, SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) remains functional, but you do not have the functionality of SQL Server 2014 Integration Services (SSIS). If you upgrade only Integration Services, SQL Server 2014 Integration Services (SSIS) is fully functional, but can only store packages in the file system, unless an instance of the SQL Server 2014 Database Engine is available on another computer.

Upgrading Both Integration Services and the Database Engine to SQL Server 2014

This section describes the effects of performing an upgrade that has the following criteria:

- You upgrade both Integration Services and an instance of the Database Engine to SQL Server 2014.
- Both Integration Services and the instance of the Database Engine are on the same computer.

What the Upgrade Process Does

The upgrade process does the following tasks:

- Installs the SQL Server 2014 Integration Services (SSIS) files, service, and tools (Management Studio and SQL Server Data Tools). When there are multiple instances of SQL Server 2005 or SQL Server 2008 on the same computer, the first time you upgrade any of the instances to SQL Server 2014, the SQL Server 2014 Integration Services (SSIS) files, service, and tools are installed.
- Upgrades the instance of the SQL Server 2005 or the SQL Server 2008Database Engine to the SQL Server 2014 version.
- Moves data from the SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) system tables to the SQL Server 2014 Integration Services (SSIS) system tables, as follows:
  - Moves packages without change from the msdb.dbo.sysdtspackages90 system table to the msdb.dbo.sysssispackages system table.

    > **NOTE**
    > Although the data moves to a different system table, the upgrade process does not migrate packages to the new format.

  - Moves folder metadata from the msdb.sysdtsfolders90 system table to the msdb.sysssisfolders system table.
  - Moves log data from the msdb.sysdtslog90 system table to the msdb.sysssislog system table.
- Removes the msdb.sysdts*90 system tables and the stored procedures that are used to access them after moving the data to the new msdb.sysssis* tables. However, upgrade replaces the sysdtslog90 table with a view that is also named sysdtslog90. This new sysdtslog90 view exposes the new msdb.sysssislog system table. This ensures that reports based on the log table continue to run without interruption.

- To control access to packages, creates three new fixed database-level roles: db_ssisadmin, db_ssisltduser, and db_ssisoperator. The SQL Server 2005Integration Services roles of db_dtsadmin, db_dtsltduser, and db_dtsoperator are not removed, but are made members of the corresponding new roles.
- If the SSIS package store (that is, the file system location managed by the Integration Services service) is the default location under **\SQL Server\90**, **\SQL Server\100**, or **\SQL Server\110** moves those packages to the new default location under **\SQL Server\120**.
- Updates the Integration Services service configuration file to point to the upgraded instance of the Database Engine.

What the Upgrade Process Does Not Do

The upgrade process does not do the following tasks:

- **Does not** remove the SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) service.
- Does not migrate existing Integration Services packages to the new package format that SQL Server 2014 uses. For information about how to migrate packages, see Upgrade Integration Services Packages.
- Does not move packages from file system locations, other than the default location, that have been added to the service configuration file. If you have previously edited the service configuration file to add more file system folders, packages that are stored in those folders will not be moved to a new location.
- In SQL Server Agent job steps that call the **dtexec** utility (dtexec.exe) directly, does not update the file system path for the **dtexec** utility. You have to edit these job steps manually to update the file system path to specify the SQL Server 2014 location for the **dtexec** utility.

What You Can Do After Upgrading

After the upgrade process finishes, you can do the following tasks:

- Run SQL Server Agent jobs that run packages.
- Use Management Studio to manage Integration Services packages that are stored in an instance of SQL Server 2008 or SQL Server 2014. You need to modify the service configuration file to add the instance of SQL Server 2008 to the list of locations managed by the service.

  > **NOTE**
  > Early versions of Management Studio cannot connect to SQL Server 2014 Integration Services (SSIS) Service.

- Identify the version of packages in the msdb.dbo.sysssispackages system table by checking the value in the packageformat column. The table has a packageformat column that identifies the version of each package. A value of 2 in the packageformat column indicates a SQL Server 2005 Integration Services (SSIS) package; a value of 3 indicates a SQL Server 2008 Integration Services (SSIS) package. Until you migrate packages to the new package format, the value in the packageformat column does not change.
- You cannot use the SQL Server 2005 or SQL Server 2008 tools to design, run, or manage Integration Services packages. The SQL Server 2005 and SQL Server 2008 tools include the respective versions of SQL Server Data Tools (SSDT), the SQL Server Import and Export Wizard, and the Package Execution Utility (dtexecui.exe). The upgrade process does not remove the SQL Server 2005 or SQL Server 2008tools. However, you will not able to use these tools to continue to work with SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) packages on a server that has been upgraded.
- By default, in an upgrade installation, Integration Services is configured to log events that are related to the running of packages to the Application event log. This setting might generate too many event log entries when you use the Data Collector feature of SQL Server 2014. The events that are logged include EventID

12288, "Package started," and EventID 12289, "Package finished successfully." To stop logging these two 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\120\SSIS node, and change the DWORD value of the LogPackageExecutionToEventLog setting from 1 to 0.

Upgrading only the Database Engine to SQL Server 2014

This section describes the effects of performing an upgrade that has the following criteria:

- You upgrade only an instance of the Database Engine. That is, the instance of the Database Engine is now an instance of SQL Server 2014, but the instance of Integration Services and the client tools are from SQL Server 2005 or SQL Server 2008.
- The instance of the Database Engine is on one computer, and Integration Services and the client tools are on another computer.

What You Can Do After Upgrading

The system tables that store packages in the upgraded instance of the Database Engine are not the same as those used in SQL Server 2005 or SQL Server 2008. Therefore, the SQL Server 2005 or SQL Server 2008 versions of Management Studio and SQL Server Data Tools cannot discover the packages in the system tables on the upgraded instance of the Database Engine. Because these packages cannot be discovered, there are limitations on what you can do with those packages:

- You cannot use the SQL Server 2005 or SQL Server 2008 tools, Management Studio and SQL Server Data Tools, on other computers to load or manage packages from the upgraded instance of the Database Engine.

  > **NOTE**
  > Although the packages in the upgraded instance of the Database Engine have not yet been migrated to the new package format, they are not discoverable by the SQL Server 2005 or SQL Server 2008 tools. Therefore, the packages cannot be used by the SQL Server 2005 or SQL Server 2008 tools.

- You cannot use SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) on other computers to run packages that are stored in msdb on the upgraded instance of the Database Engine.
- You cannot use SQL Server Agent jobs on SQL Server 2005 or SQL Server 2008 computers to run SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) packages that are stored in the upgraded instance of the Database Engine.

External Resources

Blog entry, Making your Existing Custom SSIS Extensions and Applications Work in Denali, on blogs.msdn.com.

Upgrade Integration Services Packages

12/13/2018 • 7 minutes to read • Edit Online

When you upgrade an instance of SQL Server 2005 or SQL Server 2008 to the current release of SQL Server, your existing SQL Server 2008 Integration Services (SSIS) packages are not automatically upgraded to the package format that the current release SQL ServerIntegration Services uses. You will have to select an upgrade method and manually upgrade your 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 Migrate Scripts to VSTA.

For information on upgrading packages when you convert a project to the project deployment model, see Deploy Projects to Integration Services Server.

SQL Server 2000 Data Transformation Services Packages

Support for migrating or running Data Transformation Services (DTS) packages has been discontinued in the current release of SQL ServerIntegration Services. The following DTS functionality has been discontinued.

- DTS runtime
- DTS API
- Package Migration Wizard for migrating DTS packages to the next version of Integration Services
- Support for DTS package maintenance in SQL Server Management Studio
- Execute DTS 2000 Package task
- Upgrade Advisor scan of DTS packages.

The following options are available for migrating DTS packages.

- Migrate the packages to SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS), and then upgrade the packages to SQL Server 2012 Integration Services (SSIS).

  For information about migrating DTS packages to SQL Server 2005 Integration Services (SSIS) and SQL Server 2008 Integration Services (SSIS), see Migrating Data Transformation Services Packages (2005) and Migrating Data Transformation Services Packages (2008).

- Recreate the DTS packages by using SQL Server 2012 Integration Services (SSIS).

  For information about the new features in SQL Server 2012 Integration Services (SSIS), see What's New (Integration Services). For an overview of the structure of Integration Services packages, see Integration Services (SSIS) Packages.

Selecting an Upgrade Method

You can use various methods to upgrade SQL Server 2005 and SQL Server 2008 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.

> **NOTE**
>
> When you run a SQL Server 2005 or SQL Server 2008 package using the **dtexec** utility (dtexec.exe) that is installed with the current release of SQL Server, the temporary package upgrade increases the execution time. The rate of increase in package execution time varies depending on the size of the package. To avoid an increase in the execution time, it is recommended that you upgrade the package before running it.

UPGRADE METHOD	TYPE OF UPGRADE
Use the **dtexec** utility (dtexec.exe) that is installed with the current release of SQL Server to run a SQL Server 2005 or SQL Server 2008 package. For more information, see dtexec Utility.	The package upgrade is temporary. For a SQL Server 2005 package, the script migration is temporary. The changes cannot be saved.
Open a SQL Server 2005 or SQL Server 2008 package file in SQL Server Data Tools (SSDT).	The package upgrade is permanent if you save the package; otherwise, it is temporary if you do not save the package. For a SQL Server 2005 package, the script migration is permanent if you save the package; otherwise, it is temporary if you do not save the package.
Add a SQL Server 2005 or SQL Server 2008 package to an existing project in SQL Server Data Tools (SSDT).	The package upgrade is permanent. For a SQL Server 2005 package, the script migration is permanent.
Open a SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) project file in Visual Studio, and then use the SSIS Package Upgrade Wizard to upgrade multiple packages in the project. For more information, see Upgrade Integration Services Packages Using the SSIS Package Upgrade Wizard and SSIS Package Upgrade Wizard F1 Help.	The package upgrade is permanent. For a SQL Server 2005 package, the script migration is permanent.
Use the Upgrade method to upgrade one or more Integration Services packages.	The package upgrade is permanent. For a SQL Server 2005 package, the script migration is permanent.

Custom Applications and Custom Components

SQL Server 2005 Integration Services (SSIS) custom components will not work with the current release of SQL ServerIntegration Services.

You can use the current release of SQL ServerIntegration Services tools to run and manage packages that include SQL Server 2008 and SQL Server 2008 R2SSIS custom components. We added four binding redirection rules to the following files to help redirect the runtime assemblies from version 10.0.0.0 (SQL Server 2008 R2) to version 11.0.0.0 (SQL Server 2014).

- DTExec.exe.config
- dtshost.exe.config
- DTSWizard.exe.config
- DTUtil.exe.config
- DTExecUI.exe.config

To use SQL Server Data Tools to design packages that include SQL Server 2008 and SQL Server 2008 R2 custom

components, you need to modify the devenv.exe.config file that is located at *<drive>*:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE.

To use these packages with customer applications that are built with the runtime for SQL Server 2014, include redirection rules in the configuration section of the *.exe.config file for the executable. The rules redirect the runtime assemblies to version 11.0.0.0 (SQL Server 2014). For more information about assembly version redirection, see <assemblyBinding> Element for <runtime>.

Locating the Assemblies

In SQL Server 2014, the Integration Services assemblies were upgraded to .NET 4.0. There is a separate global assembly cache for .NET 4, located in *<drive>*:\Windows\Microsoft.NET\assembly. You can find all of the Integration Services assemblies under this path, usually in the GAC_MSIL folder.

As in previous versions of SQL Server, the core Integration Services extensibility .dll files are also located at *<drive>*:\Program Files\Microsoft SQL Server\100\SDK\Assemblies.

Understanding SQL Server Package Upgrade Results

During the package upgrade process, most components and features in SQL Server 2005 and SQL Server 2008 packages convert seamlessly to their counterparts in the current release of SQL Server. However, there are several components 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.

> **NOTE**
> To identify which packages have the issues listed in this table, run Upgrade Advisor. For more information, see Use Upgrade Advisor to Prepare for Upgrades.

COMPONENT OR FEATURE	UPGRADE RESULTS
Connection strings	For SQL Server 2005 and SQL Server 2008 packages, the names of certain providers have changed and require different values in the connection strings. To update the connection strings, use one of the following procedures: -Use the SSIS Package Upgrade Wizard to upgrade the package, and select the **Update connection strings to use new provider names** option. -In SQL Server Data Tools (SSDT), on the General page of the Options dialog box, select the **Update connection strings to use new provider names** option. For more information about this option, see General Page. -In SQL Server Data Tools (SSDT), open the package and manually change the text of the ConnectionString property. Note: You cannot use the previous procedures to update a connection string when the connection string is stored in either a configuration file or a data source file, or when an expression sets the `ConnectionString` property. To update the connection string in these cases, you must manually update the file or the expression. For more information about data sources, see Data Sources.

COMPONENT OR FEATURE	UPGRADE RESULTS
Lookup transformation	For SQL Server 2005 packages, the upgrade process automatically upgrades the Lookup transformation to the current release of SQL ServerIntegration Services. However, the current release of this component has some additional capabilities of which you might want to take advantage. For more information, see Lookup Transformation.
Script task and Script component	For SQL Server 2005 packages, the 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 Migrate Scripts to VSTA.

Scripts that Depend on ADODB.dll

Script Task and Script Component scripts that explicitly reference ADODB.dll may not upgrade or run on machines without SQL Server Management Studio or SQL Server Data Tools (SSDT) installed. In order to upgrade these Script Task or Script Component scripts, it is recommended that you remove the dependency on ADODB.dll. Ado.Net is the recommended alternative for managed code such as VB and C# scripts.

External Resources

- Technical article, 5 Tips for a Smooth SSIS Upgrade to SQL Server 2012, on msdn.microsoft.com.
- Blog entry, Making your Existing Custom SSIS Extensions and Applications Work in Denali, on blogs.msdn.com.
- Webcast, Upgrading SSIS Packages to SQL Server 2012, on channel9.msdn.com.

Upgrade Integration Services Packages Using the SSIS Package Upgrade Wizard

3/22/2019 • 3 minutes to read • Edit Online

You can upgrade packages that were created in earlier versions of Integration Services to the Integration Services format that SQL Server 2014 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.

> **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 Upgrade Integration Services Packages.

The remainder of this topic describes how to run the wizard and back up the original packages.

Running the SSIS Package Upgrade Wizard

You can run the SSIS Package Upgrade Wizard from SQL Server Data Tools (SSDT), from SQL Server Management Studio, or at the command prompt.

To run the wizard from SQL Server Data Tools

1. In SQL Server Data Tools (SSDT), 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.

   > **NOTE**
   > When you open an Integration Services project that contains SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) packages, Integration Services automatically opens the SSIS Package Upgrade Wizard.

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**.

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\120\DTS\Binn** folder.

Backing 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 SQL Server Data Tools (SSDT), the wizard automatically stores both the original packages and upgraded packages in the same folder in the file system.
- 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.

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.

   > **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** tosave the upgraded packages to the same location as the original packages.

   > **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.

To back up the original packages in SQL Server Data Tools

1. Save the original packages to a location on the file system.
2. On the **Select Package Management Options** page of the wizard, select the **Backup original packages** option.

   > **WARNING**
   > The **Backup original packages** option is not displayed when you open a SQL Server 2005 Integration Services (SSIS) or SQL Server 2008 Integration Services (SSIS) project in SQL Server Data Tools (SSDT), which automatically launches the wizard.

3. In SQL Server Data Tools (SSDT), run the SSIS Package Upgrade Wizard.

Migrate Scripts to VSTA

5/22/2019 • 2 minutes to read • Edit Online

When you upgrade SQL Server 2005 packages to SQL Server 2014, 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 2014 uses. In SQL Server 2005, the scripting environment for Integration Services is Microsoft Visual Studio for Applications (VSA).

If the scripts in either the Script tasks or Script components reference interfaces, you might have to modify those references before you upgrade the package. Otherwise, the package will not be upgraded or the scripts will not be validated, depending on the upgrade method that you use. To modify these references, replace references to IDTS*xxx*90 interfaces with references to the corresponding IDTS*xxx*100 interfaces.

For more information about how to migrate scripts and upgrade packages, see Upgrade Integration Services Packages.

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# 2010. However, you can create a new VSTA script that uses Microsoft Visual C# 2010. For more information, see Coding and Debugging the Script Task and Coding and Debugging the Script Component.

See Also

Extending Packages with Scripting

Upgrade Lookup Transformations

5/22/2019 • 2 minutes to read • Edit Online

When you upgrade from SQL Server 2005 to SQL Server 2014 consider modifying packages to take advantage of the new features in the Lookup Transformation. The transformation supports the caching types and data output options available in SQL Server 2005 Integration Services (SSIS). For more information about additional the caching and data outputs, see Lookup Transformation.

In SQL Server 2005 Integration Services (SSIS), the available caching types are full caching, partial caching, and no caching. In SQL Server 2014 Integration Services (SSIS), you can configure a Lookup transformation to use one of these caching types. For more information about how to implement partial caching or no caching, see Implement a Lookup in No Cache or Partial Cache Mode. For information about how to implement full caching, see Implement a Lookup Transformation in Full Cache Mode Using the Cache Connection Manager and Implement a Lookup Transformation in Full Cache Mode Using the OLE DB Connection Manager.

In SQL Server 2005 Integration Services (SSIS), the Lookup transformation had an input, an output, and an error output. Rows in the input that had matching entries in the reference dataset were handled by the output. Rows in the input that did not have matching entries were treated as errors and could be redirected to the error output. In SQL Server 2014 Integration Services (SSIS), the Lookup transformation has two outputs: a match output and a no match output.

By default, when you run a Lookup transformation that was created in SQL Server 2005, SQL Server 2014 Integration Services (SSIS) treats the rows without matching entries as errors and enables you to redirect the rows to an error output. You have the option of configuring the Lookup transformation to treat the rows as non-errors and redirect the rows to the no match output.

For more information, see Lookup Transformation Editor (General Page).

See Also

Lookup Transformation

Upgrade Master Data Services

12/13/2018 • 13 minutes to read • Edit Online

There are four scenarios for upgrading to Microsoft SQL Server 2014 CTP2. Choose the scenario that fits your situation.

- Upgrade without Database Engine Upgrade
- Upgrade with Database Engine Upgrade
- Upgrade in Two-Computer Scenario
- Upgrade with Restoring a Database from Backup

IMPORTANT

- Upgrading from the SQL Server 2014 CTP1 release to the CTP2 release is not supported.
- Back up your database before performing any upgrade.
- The upgrade process recreates stored procedures and upgrades tables used by Master Data Services. Any customizations you have made to either of these components may be lost.
- Model deployment packages can be used only in the edition of SQL Server they were created in. You cannot deploy model deployment packages created in SQL Server 2008 R2/SQL Server 2012 to SQL Server 2014.
- You can continue using the SQL Server 2012 SP1 version of Master Data Services Add-In for Excel after upgrading Master Data Services and Data Quality Services to SQL Server 2014 CTP2. However, any earlier version of the Master Data Services Add-In for Excel will not work after upgrading to SQL Server 2014 CTP2. You can download the SQL Server 2012 SP1 version of Master Data Services Add-In for Excel from here.

Upgrade without Database Engine Upgrade

This scenario can be considered a side-by-side installation, because both SQL Server 2008 R2/SQL Server 2012 and SQL Server 2014 are installed in parallel, on either the same computer or separate computers.

In this scenario you continue to use SQL Server 2008 R2 or SQL Server 2012 to host your MDS database. However, you must upgrade the schema of the MDS database, and then create a SQL Server 2014 web application to access the MDS database. The MDS database can no longer be accessed by the SQL Server 2008 R2 or SQL Server 2012 web application.

If you choose to install SQL Server 2014 and an earlier version of SQL Server (SQL Server 2008 R2/SQL Server 2012) on the same computer, you can do so because the files are installed in a different location.

- In SQL Server 2014, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\120\Master Data Services.
- In SQL Server 2012, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\110\Master Data Services.
- In SQL Server 2008 R2, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\Master Data Services.

To perform this task, complete the following steps.

1. Install Master Data Services and any other features you want.
    a. Open the SQL Server 2014 Setup wizard.

b. In the left pane, click **Installation**.

c. In the right pane, click **New SQL Server stand-alone installation or add features to an existing installation**.

d. On the **Feature Selection** page, select **Master Data Services** and any other features you want to install.

e. Complete the wizard.

2. When the installation is complete, upgrade the MDS database schema.

a. Open the SQL Server 2014 version of Master Data Services Configuration Manager.

> **IMPORTANT**
> To upgrade the MDS database schema, you must be logged in as the Administrator Account that was specified when the MDS database was created. In the MDS database, in mdm.tblUser, this user has the **ID** value of **1**. For information on changing this user, see Change the System Administrator Account (Master Data Services).

b. In the left pane, click **Database Configuration**.

c. In the right pane, click **Select Database** and specify the information for your SQL Server 2008 R2 or SQL Server 2012 database instance.

d. Click **Upgrade Database** to start the **Upgrade Database Wizard**. For more information, see Upgrade Database Wizard (Master Data Services Configuration Manager).

3. When the upgrade is complete, create a SQL Server 2014 web application.

a. Open the SQL Server 2014 version of Master Data Services Configuration Manager.

b. In the left pane, click **Web Configuration**.

c. In the right pane, from the **Website** list, select one of the following options:

- **Default Web Site**, then click **Create Application**.
- **Create new site**. A new web application is automatically created when the website is created.

> **IMPORTANT**
> Your existing MDS web application from an earlier version of SQL Server (SQL Server 2008 R2 or SQL Server 2012) is available for selection in the SQL Server 2014 version of Master Data Services Configuration Manager. You must not select the existing web application, and instead must create a SQL Server 2014 web application for MDS. Otherwise, you will receive an error when you try to associate the web application with the upgraded MDS database stating that the requested page cannot be accessed because the related configuration data for the page is invalid.
>
> If you want to use the same name (alias) for MDS web application as your existing (SQL Server 2008 R2 or SQL Server 2012) web application, you must first delete the web application and the associated application pool from IIS, and then create a web application with the same name using SQL Server 2014 version of Master Data Services Configuration Manager. For information about removing web application and application pools from IIS, see Remove an Application (IIS) and Remove an Application Pool (IIS).

4. Now associate the new web application with the upgraded MDS database.

a. In the **Associate Application with Database** section, click **Select**.

b. Select the MDS database.

c. Click **Apply**.

Upgrade with Database Engine Upgrade

In this scenario you will upgrade both the database engine and Master Data Services application from SQL Server 2008 R2 or SQL Server 2012 to SQL Server 2014.

To perform this task, complete the following steps.

1. **For SQL Server 2008 R2 only**: Open **Control Panel** > **Programs and Features** and uninstall Microsoft SQL Server 2008 R2Master Data Services.
2. Upgrade the database engine to SQL Server 2014.
   a. Open the SQL Server 2014 Setup wizard.
   b. In the left pane, click **Installation**.
   c. In the right pane, click **Upgrade from SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 or SQL Server 2012**.
   d. Complete the wizard.
3. **For SQL Server 2008 R2 only**: When the upgrade is complete, add the **Master Data Services** feature.
   a. Open the SQL Server 2014 Setup wizard.
   b. In the left pane, click **Installation**.
   c. In the right pane, click **New SQL Server stand-alone installation or add features to an existing installation**.
   d. On the **Installation Type** page of the wizard, select the **Add features to an existing instance** option, and choose the instance where MDS database is installed.
   e. On the **Feature Selection** page, under **Shared Features**, select **Master Data Services**.
   f. Complete the wizard.
4. Upgrade the MDS database schema.
   a. Open the SQL Server 2014 version of Master Data Services Configuration Manager.

   > **IMPORTANT**
   > To upgrade the MDS database schema, you must be logged in as the Administrator Account that was specified when the MDS database was created. In the MDS database, in mdm.tblUser, this user has the **ID** value of **1**. For information on changing this user, see Change the System Administrator Account (Master Data Services).

   b. In the left pane, click **Database Configuration**.
   c. In the right pane, click **Select Database** and specify the information for your database instance.
   d. Click **Upgrade Database** to start the **Upgrade Database Wizard**. For more information, see Upgrade Database Wizard (Master Data Services Configuration Manager).
   e. Click **Apply**.

5. When the upgrade is complete, create a SQL Server 2014 web application.
    a. Open the SQL Server 2014 version of Master Data Services Configuration Manager.
    b. In the left pane, click **Web Configuration**.
    c. In the right pane, from the **Website** list, select one of the following options:
        - **Default Web Site**, then click **Create Application**.
        - **Create new site**. A new web application is automatically created when the website is created.

> **IMPORTANT**
> Your existing MDS web application from an earlier version of SQL Server (SQL Server 2008 R2 or SQL Server 2012) is available for selection in the SQL Server 2014 version of Master Data Services Configuration Manager. You must not select the existing web application, and instead must create a SQL Server 2014 web application for MDS. Otherwise, you will receive an error when you try to associate the web application with the upgraded MDS database stating that the requested page cannot be accessed because the related configuration data for the page is invalid.
>
> If you want to use the same name (alias) for MDS web application as your existing (SQL Server 2008 R2 or SQL Server 2012) web application, you must first delete the web application and the associated application pool from IIS, and then create a web application with the same name using SQL Server 2014 version of Master Data Services Configuration Manager. For information about removing web application and application pools from IIS, see Remove an Application (IIS) and Remove an Application Pool (IIS).

6. Now associate the new web application with the upgraded MDS database.
    a. In the **Associate Application with Database** section, click **Select**.
    b. Select the MDS database.
    c. Click **Apply**.

Upgrade in Two-Computer Scenario

This scenario involves upgrading a system in which SQL Server is installed on two computers: one with SQL Server 2014, and the other with SQL Server 2008 R2 or SQL Server 2012.

If SQL Server 2008 R2 or SQL Server 2012 is installed, you continue to use SQL Server 2008 R2 or SQL Server 2012 respectively to host your MDS database on one computer. However, you must upgrade the schema of the MDS database, and then use the SQL Server 2014 web application to access the MDS database. The MDS database can no longer be accessed by the SQL Server 2008 R2 or SQL Server 2012 web application.

- In SQL Server 2014, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\120\Master Data Services.
- In SQL Server 2012, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\110\Master Data Services.
- In SQL Server 2008 R2, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\Master Data Services.

To perform this task, complete the following steps.

1. Install Master Data Services and any other features you want.
    a. Open the SQL Server 2014 Setup wizard.

b. In the left pane, click **Installation**.

c. In the right pane, click **New SQL Server stand-alone installation or add features to an existing installation**.

d. On the **Feature Selection** page, select **Master Data Services** and any other features you want to install.

e. Complete the wizard.

2. When the installation is complete, upgrade the MDS database schema.

a. Open the SQL Server 2014 version of Master Data Services Configuration Manager.

> **IMPORTANT**
> To upgrade the MDS database schema, you must be logged in as the Administrator Account that was specified when the MDS database was created. In the MDS database, in mdm.tblUser, this user has the **ID** value of **1**. For information on changing this user, see Change the System Administrator Account (Master Data Services).

b. In the left pane, click **Database Configuration**.

c. In the right pane, click **Select Database** and specify the information for your SQL Server 2008 R2 or SQL Server 2012 database instance on the other computer, if SQL Server 2008 R2 or SQL Server 2012 is installed on the other computer.

d. Click **Upgrade Database** to start the **Upgrade Database Wizard**. For more information, see Upgrade Database Wizard (Master Data Services Configuration Manager).

3. When the upgrade is complete, create a SQL Server 2014 web application.

a. Open the SQL Server 2014 version of Master Data Services Configuration Manager.

b. In the left pane, click **Web Configuration**.

c. In the right pane, from the **Website** list, select one of the following options:

- **Default Web Site**, then click **Create Application**.
- **Create new site**. A new web application is automatically created when the website is created.

> **IMPORTANT**
> Your existing MDS web application from an earlier version of SQL Server (SQL Server 2008 R2 or SQL Server 2012) is available for selection in the SQL Server 2014 version of Master Data Services Configuration Manager. You must not select the existing web application, and instead must create a SQL Server 2014 web application for MDS. Otherwise, you will receive an error when you try to associate the web application with the upgraded MDS database stating that the requested page cannot be accessed because the related configuration data for the page is invalid.
>
> If you want to use the same name (alias) for MDS web application as your existing (SQL Server 2008 R2 or SQL Server 2012) web application, you must first delete the web application and the associated application pool from IIS, and then create a web application with the same name using SQL Server 2014 version of Master Data Services Configuration Manager. For information about removing web application and application pools from IIS, see Remove an Application (IIS) and Remove an Application Pool (IIS).

4. Now associate the web application with the upgraded MDS database.

a. In the **Associate Application with Database** section, click **Select**.

b. Select the MDS database.

c. Click **Apply**.

Upgrade with Restoring a Database from Backup

In this scenario, SQL Server 2014 is installed along with SQL Server 2008 R2 or SQL Server 2012 on the same computer or two different computers. Also, a database was backed up on a version earlier than the SQL Server 2014 CTP2 release, prior to upgrade, and the database has to be restored.

- In SQL Server 2014, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\120\Master Data Services.
- In SQL Server 2012, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\110\Master Data Services.
- In SQL Server 2008 R2, by default, the files are installed at *drive*:\Program Files\Microsoft SQL Server\Master Data Services.

To perform this task, complete the following steps.

1. Install Master Data Services and any other features you want.

   a. Open the SQL Server 2014 Setup wizard.

   b. In the left pane, click **Installation**.

   c. In the right pane, click **New SQL Server stand-alone installation or add features to an existing installation**.

   d. On the **Feature Selection** page, select **Master Data Services** and any other features you want to install.

   e. Complete the wizard.

2. Restore the database that was backed up.

3. When the installation is complete, upgrade the MDS database schema.

   a. Open the SQL Server 2014 version of Master Data Services Configuration Manager.

   > **IMPORTANT**
   > To upgrade the MDS database schema, you must be logged in as the Administrator Account that was specified when the MDS database was created. In the MDS database, in mdm.tblUser, this user has the **ID** value of **1**. For information on changing this user, see Change the System Administrator Account (Master Data Services).

   b. In the left pane, click **Database Configuration**.

   c. In the right pane, click **Select Database** and specify the information for your SQL Server 2008 R2 or SQL Server 2012 database instance.

   d. Click **Upgrade Database** to start the **Upgrade Database Wizard**. For more information, see Upgrade Database Wizard (Master Data Services Configuration Manager).

4. When the upgrade is complete, create a SQL Server 2014 web application.

a. Open the SQL Server 2014 version of Master Data Services Configuration Manager.

b. In the left pane, click **Web Configuration**.

c. In the right pane, from the **Website** list, select one of the following options:

- **Default Web Site**, then click **Create Application**.
- **Create new site**. A new web application is automatically created when the website is created.

> **IMPORTANT**
>
> Your existing MDS web application from an earlier version of SQL Server (SQL Server 2008 R2 or SQL Server 2012) is available for selection in the SQL Server 2014 version of Master Data Services Configuration Manager. You must not select the existing web application, and instead must create a SQL Server 2014 web application for MDS. Otherwise, you will receive an error when you try to associate the web application with the upgraded MDS database stating that the requested page cannot be accessed because the related configuration data for the page is invalid.
>
> If you want to use the same name (alias) for MDS web application as your existing (SQL Server 2008 R2 or SQL Server 2012) web application, you must first delete the web application and the associated application pool from IIS, and then create a web application with the same name using SQL Server 2014 version of Master Data Services Configuration Manager. For information about removing web application and application pools from IIS, see Remove an Application (IIS) and Remove an Application Pool (IIS).

5. Now associate the new web application with the upgraded MDS database.

a. In the **Associate Application with Database** section, click **Select**.

b. Select the MDS database.

c. Click **Apply**.

Troubleshooting

Issue: When you open the SQL Server 2008 R2 or SQL Server 2012Master Data Manager web application, a "client version is not compatible with the database version" error message is displayed.

Solution: This issue occurs when a SQL Server 2008 R2 or SQL Server 2012 Master Data Manager web application tries to access a database that has been upgraded to SQL Server 2014 Master Data Services. You must use a SQL Server 2014 web application instead.

This issue may also occur if you did not stop and restart the **MDS Application Pool** in IIS when upgrading the MDS database schema. Restart the **MDS Application Pool** to correct the issue.

See Also

Install Master Data Services

Upgrade PowerPivot for SharePoint

4/14/2020 • 19 minutes to read • Edit Online

This topic summarizes the steps required to upgrade a deployment of PowerPivot for SharePoint to Microsoft SQL Server 2014 PowerPivot for SharePoint. The specific steps depend on the version of SharePoint your environment is currently running and include the PowerPivot for SharePoint Add-in (**spPowerPivot.msi**).

Applies to: SharePoint 2010 | SharePoint 2013

For release notes, see SQL Server 2014 Release Notes.

Background

- If you are upgrading a multi-server SharePoint 2010 farm that has two or more PowerPivot for SharePoint instances, you must fully upgrade each server **before** continuing to the next server. A full upgrade includes running SQL Server Setup to upgrade PowerPivot for SharePoint program files, followed by SharePoint upgrade actions that configure the upgraded services. Server availability will be limited until you run upgrade actions in the appropriate PowerPivot Configuration Tool or Windows PowerShell.
- All instances of PowerPivot System Service and Analysis Services in a SharePoint 2010 farm must be the same version. For information on how to verify the version, see the section Verify the Versions of PowerPivot Components and Services in this topic.
- The PowerPivot configuration tools are one of the SQL Server shared features and all shared features upgrade at the same time. If during an upgrade process you select other SQL Server instances or features that require a shared feature upgrade, then the PowerPivot configuration tool will also be upgraded. You may have issues if the PowerPivot configuration tool is upgraded but your PowerPivot instance is not. For more information about SQL Server shared features, see Upgrade to SQL Server 2014 Using the Installation Wizard (Setup).
- The PowerPivot for SharePoint Add-in (**spPowerPivot.msi**) installs side by side with previous versions. For example the SQL Server 2014 add-in installs to the folder `c:\Program Files\Microsoft SQL Server\120\Tools\PowerPivotTools`.

Prerequisites

Permissions

- You must be a farm administrator to upgrade a PowerPivot for SharePoint installation. You must be a local administrator to run SQL Server Setup.
- You must have **db_owner** permissions on the farm configuration database.

SQL Server:

- If the existing PowerPivot installation is SQL Server 2008 R2, the SQL Server 2008 R2 Service Pack 2 (SP2) is required for an upgrade to SQL Server 2014 PowerPivot for SharePoint.
- If the existing PowerPivot installation is SQL Server 2012, the SQL Server 2012 Service Pack 1 (SP1) is required for an upgrade to SQL Server 2014 PowerPivot for SharePoint.

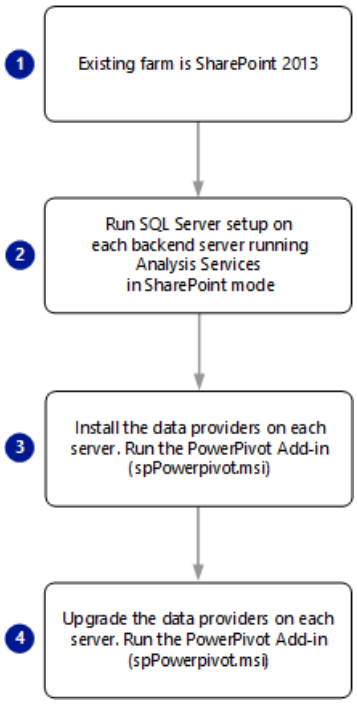
SharePoint 2010:

- If the existing installation is running SharePoint 2010, install the SharePoint 2010 Service Pack 2 before

upgrading to SQL Server 2014PowerPivot for SharePoint. For more information, see Service Pack 2 for Microsoft SharePoint 2010. Use the PowerShell command `(Get-SPfarm).BuildVersion.ToString()` to verify the version. To reference the build version to release date, see SharePoint 2010 Build Numbers.

Upgrade an Existing SharePoint 2013 Farm

To upgrade PowerPivot for SharePoint deployed in SharePoint 2013, do the following:



1. Run SQL Server 2014 Setup on backend server(s) that run Analysis Services in SharePoint mode. If the server hosts multiple instances of Analysis Services, upgrade at least the **POWERPIVOT** instance. The following list is a summary of setup wizard steps related to a PowerPivot upgrade:

   a. In the SQL Server Setup Wizard, click **Installation**.

   b. Click **Upgrade from SQL Server.....**

   c. On the **Select Instance** page, select the **POWERPIVOT** instance name and then click **Next**.

   d. For more information, see Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)

2. Restart the server.

3. Run the PowerPivot for SharePoint add-in (**spPowerPivot.msi**) on each server in the SharePoint 2013 farm to install the data providers. The exception is servers where you ran the SQL Server setup wizard, which also upgrades data providers. For more information, see Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013).

4. **Run the PowerPivot for SharePoint 2013 Configuration** tool on one of the SharePoint application servers to configure the SharePoint farm with the updated solution files that the add-in installed. You cannot use Central SharePoint Administration for this step. For more information, see the following:

   a. From the Windows Start page, type **PowerPivot** and in the search results, click **PowerPivot for SharePoint 2013 Configuration**. Note the search may return both versions of the configuration tool.

Or

On the **Start** menu, point to **All Programs**, click **Microsoft SQL Server 2014**, click **Configuration Tools**, and then click **PowerPivot for SharePoint 2013 Configuration Too**. Note that this tool is listed only when PowerPivot for SharePoint is installed on the local server.

b. At startup, the configuration tool checks the upgrade status of the PowerPivot farm solution and PowerPivot web application solutions. If older versions of these solutions are detected, you will see the message "**Newer versions of the PowerPivot solution files have been detected. Please select the upgrade option to upgrade your farm**." Click **OK** to close the system validation message.

c. Click **Upgrade Features, Services, Applications and Solutions**, and then click **OK**.

d. Review the actions in the left pane task list and exclude any that you do not want the tool to perform. All actions are included by default. To remove an action, select it in the left task list, and then on the **Parameters** page, clear the **Include this action in the task list** checkbox.

e. Optionally, review detailed information in the **Script** or **Output** tab.

The Output tab is a summary of the actions that will be performed by the tool. This information is saved in log files at `C:\Program Files\Microsoft SQL Server\120\Tools\PowerPivotTools\SPAddinConfiguration\Log`.

The Script tab shows the PowerShell cmdlets or references the PowerShell script files that the tool will run.

f. Click **Validate** to check whether each action is valid. If **Validate** is not available, it means that all of the actions are valid for your system. If **Validate** is available, you might have modified an input value (for example, the Excel service application name), or the tool might have determined that a particular action cannot be performed. If an action cannot be performed, you must exclude it or fix the underlying conditions that cause the action to be flagged as invalid.

> **IMPORTANT**
> The first action, **Upgrade Farm Solution**, must always be processed first. It registers the PowerShell cmdlets that are used to configure the server. If you get an error on this action, do not continue. Instead, use the information provided by the error to diagnose and resolve the problem before processing additional actions in the task list.

g. Click **Run** to perform all of the actions that are valid for this task. **Run** is available only after the validation check is passed. When you click **Run**, the following warning appears, reminding you that actions are processed in batch mode: "**All of the configuration settings that are flagged as valid in the tool will be applied to the SharePoint farm. Do you want to continue?**".

h. Click **Yes** to continue.

i. Upgrading solutions and features in the farm can take several minutes to complete. During this time, connection requests for PowerPivot data **will fail** with errors similar to "**Unable to refresh data**" or "**An error has occurred trying to perform the requested action. Please try again**." After upgrade is finished, the server will become available and these errors will no longer occur.

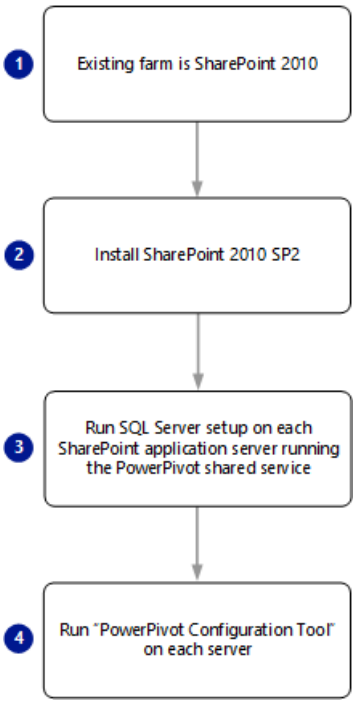
For more information, see the following:

- PowerPivot Configuration Tools
- Configure or Repair PowerPivot for SharePoint 2013 (PowerPivot Configuration Tool)
- PowerPivot Configuration using Windows PowerShell
- PowerShell Reference for PowerPivot for SharePoint

5. Verify that upgrade succeeded by performing the post-upgrade steps and by checking the version of PowerPivot servers in the farm. For more information, see Post-upgrade verification tasks in this topic and the following section.

Upgrade an Existing SharePoint 2010 Farm

To upgrade PowerPivot for SharePoint deployed in SharePoint 2010, do the following:



1. Download Service Pack 2 for Microsoft SharePoint 2010 and apply it on all servers in the farm. Verify that SharePoint SP2 installation succeeded. In Central Administration, on the Upgrade and Migration page, open the Check product and patch installation status page to view status messages related to SP2.
2. Verify that the SharePoint 2010 Administration Windows service is running.

```
Get-Service | Where {$_.displayname -like "*SharePoint*"}
```

3. Verify the **SharePoint** services **SQL Server Analysis Services** and **SQL Server PowerPivot System Service** are started in SharePoint Central Administration or use the following PowerShell command:.

```
Get-SPServiceInstance | where {$_.typename -like "*sql*"}
```

4. Verify the **Windows** service **SQL Server Analysis Services (PowerPivot)** is running.

```
Get-Service | where {$_.displayname -like "*powerpivot*"}
```

5. **Run SQL Server 2014 Setup** on the first SharePoint application server that runs the **SQL Server**

Analysis Services (PowerPivot) Windows service to upgrade the POWERPIVOT instance. On the Installation page of the SQL Server Setup Wizard, choose the upgrade option. For more information, see Upgrade to SQL Server 2014 Using the Installation Wizard (Setup).

6. **Restart the server** before running the configuration tool. This step ensures that any updates or prerequisites installed by SQL Server Setup are fully configured on the system.

7. **Run the PowerPivot Configuration Tool** on the first SharePoint application server that runs SQL Server Analysis Services (PowerPivot) service to upgrade the solutions and Web services in SharePoint. You cannot use Central Administration for this step.

   a. On the **Start** menu, point to **All Programs**, click **Microsoft SQL Server 2014**, click **Configuration Tools**, and then click **PowerPivot Configuration Tool**. Note that this tool is listed only when PowerPivot for SharePoint is installed on the local server.

   b. At startup, the configuration tool checks the upgrade status of the PowerPivot farm solution and PowerPivot web application solutions. If older versions of these solutions are detected, you will see the message "Newer versions of the PowerPivot solution files have been detected. Please select the upgrade option to upgrade your farm." Click **OK** to close the message.

   c. Click **Upgrade Features, Services, Applications and Solutions**, and then click **OK** to continue.

   d. The following warning appears: "Workbooks in the PowerPivot Management Dashboard are about to be upgraded to the latest version. Any customizations you made to the existing workbooks will be lost. Do you want to continue?"

      This warning refers to workbooks in the PowerPivot Management Dashboard that report on data refresh activity. If you customized these workbooks, any changes you made to those workbooks will be lost when existing files are replaced with newer versions.

      Click **Yes** to overwrite the workbooks with newer versions. Otherwise, click **No** to return to the home page. Save the workbooks to a different location so that you have a copy, and then return to this step when you are ready to continue.

      For more information about customizing workbooks used in the dashboard, see Customizing the PowerPivot Management Dashboard.

   e. Review the actions in the task list and exclude any that you do not want the tool to perform. All actions are included by default. To remove an action, select it in the task list, and then clear the **Include this action in the task list** checkbox on the Parameters page.

   f. Optionally, review detailed information in the **Output** tab or **Script** tab.

      The Output tab is a summary of the actions that will be performed by the tool. This information is saved in log files at `c:\Program Files\Microsoft SQL Server\120\Tools\PowerPivotTools\ConfigurationTool\Log`.

      The Script tab shows the PowerShell cmdlets or references the PowerShell script files that the tool will run.

   g. Click **Validate** to check whether each action is valid. If **Validate** is not available, it means that all of the actions are valid for your system. If **Validate** is available, you might have modified an input value (for example, the Excel service application name), or the tool might have determined that a particular action cannot be performed. If an action cannot be performed, you must exclude it or fix the underlying conditions that cause the action to be flagged as invalid.

> **IMPORTANT**
> The first action, **Upgrade Farm Solution**, must always be processed first. It registers the PowerShell cmdlets that are used to configure the server. If you get an error on this action, do not continue. Instead, use the information provided by the error to diagnose and resolve the problem before processing additional actions in the task list.

h. Click **Run** to perform all of the actions that are valid for this task. **Run** is available only after the validation check is passed. When you click **Run**, the following warning appears, reminding you that actions are processed in batch mode: "All of the configuration settings that are flagged as valid in the tool will be applied to the SharePoint farm. Do you want to continue?"

i. Click **Yes** to continue.

j. Upgrading solutions and features in the farm can take several minutes to complete. During this time, connection requests for PowerPivot data will fail with errors like "Unable to refresh data" or "An error has occurred trying to perform the requested action. Please try again." After upgrade is finished, the server will become available and these errors will no longer occur.

8. **Repeat the process** for each SQL Server Analysis Services (PowerPivot) service in the farm: 1) Run SQL Server setup 2) Run the PowerPivot Configuration tool.

9. Verify that upgrade succeeded by performing the post-upgrade steps and by checking the version of PowerPivot servers in the farm. For more information, see Post-upgrade verification tasks in this topic and the following section.

10. **Troubleshooting errors**

    You can view error information in the Parameters pane for each action.

    For problems related to solution deployment or retraction, verify the SharePoint 2010 Administrator service is started. This service runs the timer jobs that trigger configuration changes in a farm. If the service is not running, solution deployment or retraction will fail. Persistent errors indicate that an existing deployment or retraction job is already in the queue and blocking further action from the configuration tool.

    a. Start the SharePoint 2010 Management Shell as an administrator and then run the following command to view jobs in the queue:

    ```
    stsadm -o enumdeployments
    ```

    b. Review existing deployments for the following information: **Type** is Retraction or Deployment, **File** is powerpivotwebapp.wsp or powerpivotfarm.wsp.

    c. For deployments or retractions related to PowerPivot solutions, copy the GUID value for **JobId** and then paste it into the following command (use the Mark, Copy, and Paste commands on the Shell's Edit menu to copy the GUID):

    ```
    stsadm -o canceldeployment -id "<GUID>"
    ```

    d. Retry the task in the configuration tool by clicking **Validate** followed by **Run**.

    For all other errors, check the ULS logs. For more information, see Configure and View SharePoint Log Files and Diagnostic Logging (PowerPivot for SharePoint).

Workbooks

Upgrading a server does not necessarily upgrade the PowerPivot workbooks that run on it, but older workbooks created in the previous version of PowerPivot for Excel will continue to work as before, using the features available in that release. Workbooks remain functional because an upgraded server has the version of the Analysis Services OLE DB provider that was part of the previous installation.

Data Refresh

Upgrade will impact data refresh operations. Scheduled data refresh on the server is available only for workbooks that match the server version. If you are hosting workbooks from the previous version, data refresh might no longer work for those workbooks. To re-enable data refresh, you must upgrade the workbooks. You can upgrade each workbook manually in PowerPivot for Excel, or enable the auto-upgrade for data refresh feature in SharePoint 2010. Auto-upgrade will upgrade a workbook to the current version prior to running data refresh, allowing data refresh operations to remain on schedule.

Verify the Versions of PowerPivot Components and Services

All instances of PowerPivot System Service and Analysis Services must be the same version. To verify that all server components are at the same version, check version information for the following:

Verify the version of PowerPivot Solutions and the PowerPivot System Service

Run the following PowerShell command:

```
Get-PowerPivotSystemService
```

Verify the **CurrentSolutionVersion**. SQL Server 2014 is version 12.0.<major build>.<minor build>

Verify the version of the Analysis Services Windows Service

If you upgraded only some of your PowerPivot for SharePoint servers in a SharePoint 2010 farm, the instance of Analysis Services on un-upgraded servers will be older than the version expected in the farm. You will need to upgrade all of your servers to the same version in order for them to be usable. Use one of the following methods to verify the version of the SQL Server Analysis Services (PowerPivot) Windows service on each computer.

Windows File Explorer:

1. Navigate to the **Bin** folder for the PowerPivot instance. For example `C:\Program Files\Microsoft SQL Server\MSAS12.POWERPIVOT\OLAP\bin`.
2. Right-click `msmdsrv.exe`, and select **Properties**.
3. Click **Details**.
4. SQL Server 2014 file version should be 12.00.<major build>.<minor build>.
5. Verify that this number is identical to the PowerPivot solution and system service version.

Service start information:

When the PowerPivot service starts, it writes version information into the Windows event log.

1. Run Windows `eventvwr`
2. Create a filter for source `MSOLAP$POWERPIVOT`.
3. Look for an information level event similar to the following

   Service started. Microsoft SQL Server Analysis Services 64 Bit Evaluation (x64) RTM **12.0.2000.8**.

Use PowerShell to verify file version.

You can use PowerShell to verify the product version. PowerShell is a good option if you want to script or automate version verification.

```
(Get-ChildItem "C:\Program Files\Microsoft SQL
Server\MSAS12.POWERPIVOT2000\OLAP\bin\msmdsrv.exe").VersionInfo
```

The above PowerShell command returns information similar to the following:

ProductVersion FileVersion FileName

12.0.2000.8 2014.0120.200 C:\Program Files\Microsoft SQL Server\MSAS12.POWERPIVOT2000\OLAP\bin\msmdsrv.exe

Verify the MSOLAP Data Provider Version on SharePoint

Use the following instructions to check which versions of the Analysis Services OLE DB providers are trusted by Excel Services. You must be a farm or service application administrator to check the trusted data provider settings of Excel Services.

1. In Central Administration, in Application Management, click **Manage service applications**.
2. Click the name of the Excel Services service application, for example **ExcelServiceApp1**.
3. Click **Trusted Data Providers**. You should see MSOLAP.5 (Microsoft OLE DB Provider for OLAP Services 11.0). If you upgraded your PowerPivot for SharePoint installation, you will also see MSOLAP.4 from the previous version.
4. For more information, see Add MSOLAP.5 as a Trusted Data Provider in Excel Services.

MSOLAP.4 is described as the Microsoft OLE DB Provider for OLAP Services 10.0. This version might be the default version from SQL Server 2008 that is installed with Excel Services, or it might be the SQL Server 2008 R2 version. The default version that SharePoint installs does not support PowerPivot data access. You must have the SQL Server 2008 R2 version or later to connect to PowerPivot workbooks on SharePoint. To verify you have the SQL Server 2008 R2 version, use the instructions in the previous section that explain how to verify the version by viewing file properties.

Verify the ADOMD.NET Data Provider Version

Use the following instructions to check which version of ADOMD.NET is installed. You must be a farm or service application administrator to check the trusted data provider settings of Excel Services.

1. On your SharePoint application server, browse to `c:\Windows\Assembly`.
2. Sort by assembly name and find the **Microsoft.Analysis Services.Adomd.Client**.
3. Verify you have version 12.0.<build number>.

Upgrading Multiple PowerPivot for SharePoint Servers in a SharePoint Farm

In a multi-server topology that includes more than one PowerPivot for SharePoint server, all server instances and components must be the same version. The server that runs the highest version of the software sets the level for all servers in the farm. If you upgrade just some of the servers, the ones that are running older versions of the software will become unavailable until they are also upgraded.

After you upgrade the first server, additional servers that are not yet upgraded **will become unavailable**. Availability is restored after all servers run at the same level.

SQL Server Setup upgrades the PowerPivot solution files in place on the physical computer, but to upgrade the solutions in use by the farm, you must use the PowerPivot Configuration Tool described in a previous section of this topic..

Applying a QFE to a PowerPivot instance in the farm

Patching a PowerPivot for SharePoint server updates existing program files with a newer version that includes a fix for a specific problem. When applying a QFE to a multi-server topology, there is no primary server that you must begin with. You can start with any server as long as you apply the same QFE to the other PowerPivot servers in the farm.

When you apply the QFE, you must also perform a configuration step that updates the server version information in the farm configuration database. The version of the patched server becomes the new expected version for the farm. Until the QFE is applied and configured across all machines, the PowerPivot for SharePoint instances that do not have the QFE will be unavailable to handle requests for PowerPivot data.

To ensure the QFE is applied and configured correctly, follow these instructions:

1. Install the patch using the instructions that are provided with the QFE.
2. Start the PowerPivot Configuration Tool.
3. Click **Upgrade Features, Services, Applications and Solutions**, and then click **OK**.
4. Review the actions that are included in the upgrade task, and then click **Validate**.
5. Click **Run** to apply the actions.
6. Repeat for additional PowerPivot for SharePoint instances in the farm.

> **IMPORTANT**
> In a multi-server deployment, be sure to both patch and configure each instance before continuing to the next machine. The PowerPivot Configuration Tool must complete the upgrade task for the current instance before you move on to the next instance.

To check version information for services in the farm, use the **Check product and patch installation status** page in the Upgrade and Patch Management section in Central Administration.

Post-upgrade verification tasks

After upgrade is complete, use the following steps to verify the server is operational.

TASK	LINK
Verify the service is running on all computers that run PowerPivot for SharePoint.	Start or Stop a PowerPivot for SharePoint Server
Verify feature activation at the site collection level.	Activate PowerPivot Feature Integration for Site Collections in Central Administration
Verify individual PowerPivot workbooks are loading properly by opening a workbook and clicking on filters and slicers to initiate a query.	Check for the presence of cached files on the hard drive. A cached file confirms that the data file was loaded on that physical server. Look for cached files in the c:\Program Files\Microsoft SQL Server\MSAS12.POWERPIVOT\OLAP\Backup folder.

TASK	LINK
Test data refresh on selected workbooks that are configured for data refresh.	The easiest way to test data refresh is to modify a data refresh schedule, choosing the **Also refresh as soon as possible** checkbox so that data refresh runs immediately. This step will determine whether data refresh is successful for the current workbook. Repeat these steps for other frequently used workbooks to ensure that data refresh is functional. For more information about scheduling data refresh, see Schedule a Data Refresh (PowerPivot for SharePoint).
Over time, monitor the data refresh reports in PowerPivot Management Dashboard to confirm there are no data refresh errors.	PowerPivot Management Dashboard and Usage Data

For more information about how to configure PowerPivot settings and features, see PowerPivot Server Administration and Configuration in Central Administration.

For step-by-step instructions that guide you through all of the post-installation configuration tasks, see Initial Configuration (PowerPivot for SharePoint).

See Also

Features Supported by the Editions of SQL Server 2014
PowerPivot for SharePoint 2010 Installation

Upgrade Replicated Databases

1/8/2019 • 4 minutes to read • Edit Online

SQL Server 2014 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:

- 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 2008 Publisher running can have SQL Server 2014 Subscribers; and a SQL Server 2014 Publisher can have SQL Server 2008 Subscribers.
  - A Subscriber to a merge publication can be any version less than or equal to the Publisher version.

> **NOTE**
> This topic is available in the Setup Help documentation and in SQL Server 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 2014, 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 2014.
7. Restart SQL Server Agent and the Log Reader Agent if they do not start automatically after the upgrade.

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.

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:

- Create and Apply the Initial Snapshot
- Start and Stop a Replication Agent (SQL Server Management Studio)
- Create and Apply the Initial Snapshot
- Replication Agent Executables Concepts

For more information about running the Merge Agent, see the following topics:

- Synchronize a Pull Subscription
- Synchronize a Push Subscription

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.

Upgrading to Standard, Workgroup, or Express Editions

Before upgrading from one edition of SQL Server 2014 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 on Replication in Features Supported by the Editions of SQL Server 2014.

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 Configure 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

Replication Administration FAQ
Replication Backward Compatibility
Supported Version and Edition Upgrades
Upgrade to SQL Server 2014

Upgrade and Migrate Reporting Services

2/28/2020 • 15 minutes to read • Edit Online

This topic is an overview of the upgrade and migration options for SQL Server 2014Reporting Services. There are two general approaches to upgrading a Reporting Services deployment:

- **Upgrade:** You upgrade the Reporting Services components on the servers and instances where they are currently installed. This is commonly called an "in place" upgrade. In-place upgrade is not supported from one mode of Reporting Services server to another. For example, you cannot upgrade a Native Mode report server to a SharePoint mode report server. You can migrate your report items from one mode to another. For more information, see the 'Native to SharePoint Migration' section later in this document and the related topic Sample Reporting Services rs.exe Script to Migrate Content between Report Servers.
- **Migrate**: You install and configure a new SharePoint environment, copy your report items and resources to the new environment, and configure the new environment to use existing content. A lower level form of migration is to copy the Reporting Services databases, configuration files, and if you are using SharePoint mode, the SharePoint content databases.

Applies to: Reporting Services Native mode | Reporting Services SharePoint mode

In this topic:

- Known Upgrade Issues and Best Practices
- Side By Side Installations
- In-place upgrade
- Pre-Upgrade Checklist
- Native Mode Upgrade and Migration Scenarios
- Upgrade a Reporting Services Native Mode Scale-out Deployment
- SharePoint Mode Upgrade and Migration Scenarios
- Considerations for a Migration
- Additional Resources

Known Upgrade Issues and Best Practices

For a detailed list of the supported editions and versions you can upgrade, see Supported Version and Edition Upgrades.

TIP

For the latest information regarding issues with SQL Server 2014, see the following:

- SQL Server 2014 Release Notes.
- SQL Server 2014 Reporting Services Tips, Tricks, and Troubleshooting.
- Use SQL Server Upgrade Advisor. For more information, see Reporting Services Upgrade Issues (Upgrade Advisor) and How to: Install Upgrade Advisor.

In this topic:

Side By Side Installations

SQL Server 2014 Reporting Services (SSRS) Native mode can be installed side-by-side with a SQL Server 2012 Native mode deployment.

There is no support for side-by-side deployments of SQL Server 2014 Reporting Services (SSRS) SharePoint mode and any previous versions of Reporting Services SharePoint mode components.

In this topic:

In-place upgrade

Upgrade is completed by SQL Server Setup. SQL Server Setup can be used to upgrade any or all SQL Server components, including Reporting Services. Setup detects the existing instances and prompts you to upgrade. SQL Server Setup provides upgrade options that you can specify as a command-line argument or in the Setup wizard.

When you run SQL Server Setup, you can select the option to upgrade from one of the following versions or you can install a new instance of SQL Server 2014 Reporting Services (SSRS) that runs side-by-side existing installations:

- SQL Server 2012
- SQL Server 2008 R2
- SQL Server 2008
- SQL Server 2005

For more information on SQL Server, see the following:

Upgrade to SQL Server 2014
Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)
Install SQL Server 2014 from the Command Prompt

In this topic:

Pre-Upgrade Checklist

Before upgrading to SQL Server 2014, review the following:

- Review requirements to determine whether your hardware and software can support SQL Server 2014

Reporting Services (SSRS). For more information, see Hardware and Software Requirements for Installing SQL Server 2014.

- Use System Configuration Checker (SCC) to scan the report server computer for any conditions that might prevent a successful installation of SQL Server 2014. For more information, see Check Parameters for the System Configuration Checker.
- Review security best practices and guidance for SQL Server. For more information, see Security Considerations for a SQL Server Installation.
- Run SQL Server Upgrade Advisor on the report server computer to determine any issues that might prevent you from successfully upgrading. For more information, see Use Upgrade Advisor to Prepare for Upgrades.
- Back up your symmetric key. For more information, see Back Up and Restore Reporting Services Encryption Keys.
- Back up your report server databases and configuration files. For more information, see Backup and Restore Operations for Reporting Services.
- Back up any customizations to existing Reporting Services virtual directories in IIS.
- Remove invalid SSL certificates. This includes certificates that are expired and you do not plan to update prior to upgrading Reporting Services. Invalid certificates will cause upgrade to fail and an error message similar to the following will be written to the Reporting Services Log file: **Microsoft.ReportingServices.WmiProvider.WMIProviderException: A Secure Sockets Layer (SSL) certificate is not configured on the Web site.**.

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.

In this topic:

Overview of Migration Scenarios

If you are upgrading from a supported version of Reporting Services to SQL Server, you can usually run the SQL Server Setup Wizard to upgrade the report server program files, database, and all application data.

However, **migrating** a report server installation manually is required if you encounter any of the following conditions:

- Upgrade Advisor detected one more upgrade blockers. For more information, see Use Upgrade Advisor to Prepare for Upgrades.
- You want to change the type of report server used in your deployment. For example, you cannot upgrade or convert a native mode report server to SharePoint mode. For more information, see Native to SharePoint Migration (SSRS).
- You want to minimize the amount of time the report server is taken offline during the upgrade process. Your current installation remains online while you copy content data to a new report server instance and test the installation without changing the state of your existing report server installation.
- You want to migrate a SharePoint 2010 deployment of Reporting Services to SharePoint 2013. SharePoint 2013 does not support in-place upgrade from SharePoint 2010. For more information, see Migrate a Reporting Services Installation (SharePoint Mode).

In this topic:

Native Mode Upgrade and Migration Scenarios

Upgrade: In-place upgrade for native mode is the same process for each of the supported versions that are listed earlier in this topic. Run the SQL Server installation wizard or a command line installation. Following installation the report server database will automatically upgrade to the new report server database schema. For more information, see the In-place upgrade section in this topic.

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 2014 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 2014 program files. The program folders for a Reporting Services installation include MSRS12.<*instance name*>.

5. Setup adds the SQL Server 2014 report server program files, configuration tools, and command line utilities that are part of the report server feature.

    a. Program files from the previous version are removed.

    b. Report server configuration tools and utilities that are upgraded to the new version include the Native Mode Reporting Services Configuration tool, command line utilities such as RS.exe, and Report Builder.

    c. Other client tools such as 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 2014 versions. If you installed samples, the earlier version will remain. Setup does not support upgrade for the SQL Server samples.

    d. SQL Server Data Tools (SSDT) is a separate download. For more information, see Microsoft SQL Server 2014 Data Tools - Business Intelligence for Microsoft Visual Studio 2012.

6. Setup reuses the service entry in Service Control Manager for the SQL Server 2014 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 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. 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.

Migration: Migrating a previous version of a native mode installation to SQL Server 2014 is the same steps for

all of the supported s versions that are listed earlier in this topic. For more information, see Migrate a Reporting Services Installation (Native Mode)

In this topic:

Upgrade a Reporting Services Native Mode Scale-out Deployment

The following is a summary of how to upgrade a Reporting Services Native mode deployment that is scaled-out to more than one report server. This process requires downtime of the Reporting Services deployment:

1. Backup the report server databases and encryption keys. For more information, see Backup and Restore Operations for Reporting Services and Add and Remove Encryption Keys for Scale-Out Deployment (SSRS Configuration Manager).
2. Use the Reporting Services Configuration Manager and remove all of the report servers from the scaled-out deployment. For more information, see Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager).
3. Upgrade one of the report servers to SQL Server 2014.
4. Use the Reporting Services Configuration Manager to add the report servers back to the scale-out deployment. For more information, see Configure a Native Mode Report Server Scale-Out Deployment (SSRS Configuration Manager).

   For each server, repeat the upgrade and Scale-out steps.

SharePoint Mode Upgrade and Migration Scenarios

The following sections describe the issues and basic steps needed to upgrade or migrate from specified versions of Reporting Services SharePoint mode to SQL Server 2014 Reporting Services SharePoint mode.

There are two installation components to upgrade a Reporting Services SharePoint Mode deployment.

- Reporting Services SharePoint Shared Service.

  TIP
  Use the Reporting Services SharePoint cmdlet `Get-SPRSServiceApplicationServers` to determine servers in the SharePoint farm that are currently running the Reporting Services SharePoint Shared Service and therefore require an upgrade.

- Reporting Services Add-in for SharePoint products. For more information, see Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013).

For detailed steps on Migrating a SharePoint mode installation, see Migrate a Reporting Services Installation (SharePoint Mode).

IMPORTANT
Some of the following scenarios require down time of the SharePoint environment due to the different technologies that need to be upgraded. If your situation does not allow for down time, you will need to complete a migration instead of an in-place upgrade.

SQL Server 2012 to SQL Server 2014

Starting environment: SQL Server 2012 or SQL Server 2012 SP1., SharePoint 2010.

Ending environment: SQL Server 2014, SharePoint 2010 or SharePoint 2013.

- **SharePoint 2010:** In-place upgrade of Reporting Services is supported but the upgrade scenario does require down time of the SharePoint environment.

  If you also want the ending environment to run SharePoint 2013, you need to complete a database-attach upgrade of the SharePoint 2010 to SharePoint 2013.

- **SharePoint 2013:** SharePoint 2013 does not support in-place upgrade from SharePoint 2010. However the procedure of **database-attach upgrade** is supported. The behavior is different from upgrading to SharePoint 2010, where a customer could choose between the two basic upgrade approaches, in-place upgrade and database-attach upgrade.

  If you have a Reporting Services installation integrated with SharePoint 2010, you cannot upgrade in-place the SharePoint server. However you can migrate content databases and service application databases from the SharePoint 2010 farm to a SharePoint 2013 farm.

SQL Server 2008 R2 to SQL Server 2014

Starting environment: SQL Server 2008 R2, SharePoint 2010.

Ending environment: SQL Server 2014, SharePoint 2010.

- In-place upgrade is supported and there is no down time for your SharePoint environment.
- Install the SQL Server 2014 version of the Reporting Services add-in for SharePoint on each web front-end in the farm. You can install the add-in by using the SQL Server 2014 installation wizard or by downloading the add-in.
- Run SQL Server 2014 installation to upgrade SharePoint mode for each 'report server'.The SQL Server installation wizard will install the Reporting Services Service and create a new Service application.

  If you also want the ending environment to run SharePoint 2013, you need to complete a database-attach upgrade of the SharePoint 2010 to SharePoint 2013.

In this topic:

SQL Server 2008 SP2 to SQL Server 2014

Starting environment: SQL Server 2008 SP2, SharePoint 2007.

Ending environment: SQL Server 2014, SharePoint 2010.

- This in-place upgrade scenario does require down time of the SharePoint environment because both the SharePoint and SQL Server technologies need to be upgraded. You may want to consider completing a migration rather than an in-place upgrade.
- Upgrade SQL Server 2008 to Service Pack 2 (SP2) first, if that has not already been completed.
- Upgrade SharePoint to 2010. When you run the SharePoint 2010 prerequisite installer it will upgrade the Reporting Services add-in for SharePoint 2010 products.
- Install the SQL Server 2014 version of the Reporting Services add-in for SharePoint on all SharePoint web front-ends. The SharePoint pre-requisite installer installed the SQL Server 2008 R2 version of the add-in but you need the SQL Server 2014 version to work with a SQL Server 2014 report server.
- **WARNING**

  Following the SharePoint upgrade your reporting services environment will be in a non-working state until SQL Server is upgraded.

- Upgrade Reporting Services to SQL Server 2014. When you run the SQL server installation wizard, you will a dialog regarding "**SQL Server Reporting Services SharePoint Mode Authentication**" dialog. The Reporting Services service will be installed and the credentials from the authentication page will be used for creating a new SharePoint application pool.

In this topic:

SQL Server 2005 SP2 to SQL Server 2014

Starting environment: SQL Server 2005 SP2, SharePoint 2007.

Ending environment: SQL Server 2014, SharePoint 2010.

- This in-place upgrade scenario does require down time of the SharePoint environment because both the SharePoint and SQL Server technologies need to be upgraded. You may want to consider completing a migration rather than an in-place upgrade.
- Upgrade SQL server 2005 to Service Pack 2 (SP2) first, if that has not already been completed.
- Upgrade SharePoint to SharePoint 2010. When you run the SharePoint 2010 prerequisite installer it will upgrade the Reporting Services add-in for SharePoint 2010 products.
- **WARNING**

  Following the SharePoint upgrade your reporting services environment will be in a non-working state until SQL Server is upgraded.

- Install the SQL Server 2014 version of the Reporting Services add-in for SharePoint on all SharePoint web front-ends. The SharePoint pre-requisite installer installed the SQL Server 2008 R2 version of the add-in but you need the SQL Server 2014 version to work with a SQL Server 2014 report server.
- Upgrade Reporting Services to SQL Server 2014. When you run the SQL server installation wizard, you will a dialog regarding "SQL Server Reporting Services SharePoint Mode Authentication" dialog. The Reporting Services service will be installed and the credentials from the authentication page will be used for creating a new SharePoint application pool.

In this topic:

Considerations for a Migration

When moving application data, you should be aware of the following concerns and restrictions:

- Protection of encryption key includes a hash that incorporates machine identity.
- Report server database names are fixed and cannot be renamed on new computer.

Encryption Key Considerations

Always back up the encryption keys before moving a report server database to a new computer.

Moving a report server installation to another computer will invalidate the hash that protects the encryption keys used to help secure sensitive data stored in the report server database. Each report server instance that uses the database has its copy of the encryption key, which is encrypted with the identity of the service account as it is defined on the current computer. If you change computers, the service will no longer have access to its key, even if you use the same account name on the new computer.

To re-establish reversible encryption on the new report server computer, you must restore the key that you previously backed up. The complete key set that is stored in the report server database consists of a symmetric key value, plus service identity information used to restrict access to the key so that it can be used only by the

report server instance that stored it. During key restoration, the report server replaces existing copies of the key with new versions. The new version includes machine and service identity values as defined on the current computer. For more information, see the following topics:

- SharePoint mode: See the "Key Management" section of Manage a Reporting Services SharePoint Service Application
- Native Mode: See Back Up and Restore Reporting Services Encryption Keys

In this topic:

Fixed Database Name

You cannot rename the report server database. The identity of the database is recorded in report server stored procedures when the database is created. Renaming either the report server primary or temporary databases will cause errors to occur when the procedures run, invalidating your report server installation.

If the database name from the existing installation is not suited for the new installation, you should consider creating a new database that has the name that you prefer, and then load existing application data using the techniques in the following list:

- Write a Visual Basic script that calls Report Server Web service SOAP methods to copy data between databases. You can use the RS.exe utility to run the script. For more information about this approach, see Scripting and PowerShell with Reporting Services.
- Write code that calls the WMI provider to copy data between databases. For more information about this approach, see Access the Reporting Services WMI Provider.
- If you have just a few items, you can republish reports, report models, and shared data sources from Report Designer, Model Designer, and Report Builder to the new report server. You must re-create role assignments, subscriptions, shared schedules, report snapshot schedules, custom properties that you set on reports or other items, model item security, and properties that you set on the report server. You will lose report history and report execution log data.

In this topic:

Additional Resources

> **NOTE**
> For more information on SharePoint database-attach upgrade, see the following:

- Overview of the upgrade process to SharePoint 2013 (https://go.microsoft.com/fwlink/p/?LinkId=256688).
- Clean up preparations before an upgrade to SharePoint 2013 (https://go.microsoft.com/fwlink/p/?LinkId=256689).
- Upgrade databases from SharePoint 2010 to SharePoint 2013 (https://go.microsoft.com/fwlink/p/?LinkId=256690).

In this topic:

See Also

Upgrade Reports Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)

Migrate a Reporting Services Installation (Native Mode)

4/18/2019 • 13 minutes to read • Edit Online

This topic provides step-by-step instructions for migrating one of the following supported versions of a Reporting Services native mode deployment to a new SQL Server 2014 Reporting Services instance:

- SQL Server 2012
- SQL Server 2008 R2
- SQL Server 2008
- SQL Server 2005 (Requires more steps, see You cannot use SQL Server 2005 to host Report Server 2014 Databases.

Applies to: Reporting Services Native mode.

For information on migrating a Reporting Services SharePoint mode deployment, see Migrate a Reporting Services Installation (SharePoint Mode).

Migration is defined as moving application data files to a new SQL Server 2014 instance. The following are common reasons you must migrate your installation:

- You have a large-scale deployment or uptime requirements.
- You are changing the hardware or topology of your installation.
- You encounter an issue that blocks upgrade.

Native Mode Migration Overview

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 2014. If you are using the same hardware, you can install SQL Server 2014 side-by-side with your existing installation if it was one of the supported versions.

  > **TIP**
  > A side-by-side installation may require that you install SQL Server 2014 as a named instance.

- Move the report server database and other application files from your existing installation to your new SQL Server 2014 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.
- Remove unused applications and tools after you have confirmed that the new instance is fully operational.

There are restrictions on the editions of SQL Server that host the report server database. Review the following topic if you are reusing a report server database that was created in a previous installation.

- Create a Report Server Database (SSRS Configuration Manager)

Fixed Database Name

You cannot rename the report server database. The identity of the database is recorded in report server stored procedures when the database is created. Renaming either the report server primary or temporary databases causes errors when the procedures run, invalidating your report server installation.

If the database name from the existing installation is not suited for the new installation, you should consider creating a new database that has the name, and then load existing application data using the techniques in the following list:

- Write a Visual Basic script that calls Report Server Web service SOAP methods to copy data between databases. You can use the RS.exe utility to run the script. For more information about this approach, see Scripting and PowerShell with Reporting Services.
- Write code that calls the WMI provider to copy data between databases. For more information about this approach, see Access the Reporting Services WMI Provider.
- If you have just a few items, you can republish reports, report models, and shared data sources from Report Designer, Model Designer, and Report Builder to the new report server. You must re-create role assignments, subscriptions, shared schedules, report snapshot schedules, custom properties that you set on reports or other items, model item security, and properties that you set on the report server. You will lose report history and report execution log data.

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 2014 installation.

In addition, you should be aware of several important changes in SQL Server 2014 that will affect how you migrate your installation:

- Starting with SQL Server 2008, 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 and later versions. 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 and later versions. 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 and later versions. 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 and later versions.
- 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 2014 Reporting Services, see the Upgrade Advisor documentation and What's New (Reporting Services).

Backup 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 Manager.
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 (SSRS Native Mode).
3. Back up the report server configuration files. Files to back up include:
    a. Rsreportserver.config
    b. Rswebapplication.config
    c. Rssvrpolicy.config
    d. Rsmgrpolicy.config
    e. Reportingservicesservice.exe.config
    f. Web.config for both the Report Server and Report Manager ASP.NET applications.
    g. Machine.config for ASP.NET if you modified it for report server operations.

Install SQL Server 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:

- Install SQL Server 2014 from the Installation Wizard (Setup)
- Install SQL Server 2014 from the Command Prompt

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 section Move Custom Assemblies or Extensions.

To move the report server database, do the following:

1. Choose the Database Engine instance to use. SQL Server 2014 Reporting Services requires that you use one of the following versions to host the report server database:
   - SQL Server 2014
   - SQL Server 2012
   - SQL Server 2008 R2
   - SQL Server 2008
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 Create the RSExecRole.
4. Follow the instructions in Moving the Report Server Databases to Another Computer (SSRS Native Mode).

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.

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 section 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 2005 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 and later versions.
   - Other custom assemblies should not require recompilation.
2. Move the assemblies to the new report server and Report Manager \bin folders. In SQL Server 2014, the report server binaries are located in the following location for the default SQL Server 2014 Reporting Services instance:

```
\Program files\Microsoft SQL Server\MSRS12.MSSQLSERVER\Reporting Services\ReportServer\bin
```

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:

   a. Deploying a Custom Assembly

   b. How to: Deploy a Custom Report Item

   c. Deploying a Data Processing Extension

   d. Deploying a Delivery Extension

   e. Deploying a Rendering Extension

   f. Implementing a Security Extension

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 2014 database version.

> **IMPORTANT**
> 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.

> **NOTE**
> If the report server you migrated was configured as the shared database for a scale-out deployment, you need to delete any of the old encryption keys from the **Keys** table in the **ReportServer** database, before configuring the report server service. If the keys are not removed, the migrated report server will try to initialize in scale-out deployment mode. For more information, see Add and Remove Encryption Keys for Scale-Out Deployment (SSRS Configuration Manager) and Configure and Manage Encryption Keys (SSRS Configuration Manager).
>
> The scale-out keys cannot be deleted by using the Reporting Services Configuration Manager. The old keys must be deleted from the **Keys** table in the **ReportServer** database using SQL Server Management Studio. Delete all rows in the Keys table. This will clear the table and prepare it for restoring the Symmetric key only, as documented in the following steps.
>
> Prior to deleting the keys it is recommended you first back up the Symmetric Encryption key. You can use the Reporting Services Configuration Manager to back up the key. Open the Configuration Manager open, click the **Encryption Keys** tab and then click the **Backup** button. You can also script WMI commands to back up the encryption key. For more information on WMI, see BackupEncryptionKey Method (WMI MSReportServer_ConfigurationSetting).

1. Start the Reporting Services Configuration Manager and connect to the Reporting Services instance you just installed. For more information, see Reporting Services Configuration Manager (Native Mode).

2. Configure URLs for the report server and Report Manager. For more information, see Configure a URL (SSRS Configuration Manager).

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 2014

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 Create a Native Mode Report Server Database (SSRS Configuration Manager).

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 Back Up and Restore Reporting Services Encryption Keys.

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 Configure a Firewall for Report Server Access.

6. If you want to administer your native mode report server locally, you need to configure the operating system to allow local administration with Report Manager. For more information, see Configure a Native Mode Report Server for Local Administration (SSRS).

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 2014.

CUSTOMIZATION	INFORMATION
Report Server E-mail delivery with custom settings	Configure a Report Server for E-Mail Delivery (SSRS Configuration Manager) and E-mail Settings - Configuration Manager (SSRS Native Mode).
Device information settings	Customize Rendering Extension Parameters in RSReportServer.Config
Report Manager on a remote instance	Configure Report Manager (Native Mode)

Windows Service Group and Security ACLs

In SQL Server 2014 Reporting Services (SSRS), 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 2014 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 2014 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 Verify a Reporting Services Installation.

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 2014 report object model when processing and rendering reports, but it does not replace SQL Server 2008 or SQL Server 2008 R2 constructs with new report definition language elements. To learn more about how existing reports run on a SQL Server 2014 report server, see Upgrade Reports.

Remove Unused Programs and Files

Once you have successfully migrated your report server to a SQL Server 2014 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.

See Also

Migrate a Reporting Services Installation (SharePoint Mode)
Report Server Database (SSRS Native Mode)
Upgrade and Migrate Reporting Services
Reporting Services Backward Compatibility
Reporting Services Configuration Manager (Native Mode)

Migrate a Reporting Services Installation (SharePoint Mode)

2/28/2020 • 7 minutes to read • Edit Online

This topic is an overview of the steps needed to migrate a Reporting Services SharePoint mode deployment from one SharePoint environment to another. The specific steps can be different depending on the version you are migrating from. For more information on Upgrade and Migration scenarios for SharePoint mode, see Upgrade and Migrate Reporting Services. If you only want to copy the report items from one server to another, see Sample Reporting Services rs.exe Script to Migrate Content between Report Servers.

For information on migrating a Reporting Services native mode deployment, see Migrate a Reporting Services Installation (Native Mode).

Applies to: SharePoint 2010 and SharePoint 2013

A common reason you complete a migration is when you want to upgrade your SharePoint 2010 deployment to SharePoint 2013. SharePoint 2013 does not support in-place upgrade from SharePoint 2010 and you must complete the procedure of **database-attach upgrade** or a content only migration.

For more information on upgrading SharePoint 2013, see the following:

- Overview of the upgrade process to SharePoint 2013.
- Upgrade databases from SharePoint 2010 to SharePoint 2013.
- Move content databases in SharePoint 2013.

Migrate from Reporting Services SharePoint mode versions prior to SQL Server 2012

The Reporting Services SharePoint mode architecture changed in SQL Server 2012, including the service application database schema. If you want to migrate to SQL Server 2014 Reporting Services (SSRS) SharePoint mode from a version prior to SQL Server 2012, first create the new SharePoint environment by installing SharePoint and SQL Server 2014 Reporting Services SharePoint mode. For more information, see Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013).

Once the new SharePoint environment is running, you can choose between a content only migration or a full migration at the database level that includes content databases.

Content Only Migration

Reporting Services Content only migration: If you want to copy the Reporting Services content to a new farm, then you need to use tools such as **rs.exe** to copy the content to the new SharePoint installation. For more information on content only migrations, see the following:

- **Reporting Services RSS scripts:** The Scripts can migrate content and resources between Native mode and SharePoint mode report servers. For more information, see Sample Reporting Services rs.exe Script to Migrate Content between Report Servers and Reporting Services RS.exe script that migrates content from one report server to another.
- **Reporting Services Migration Tool:** The tool can copy your report items from a native mode server to a

SharePoint mode server. For more information, see Reporting Services Migration Tool.

Full Migration

Full Migration: If you are migrating SharePoint content databases along with the Reporting Services catalog databases to a new farm you can follow a series of backup and restore options summarized in this topic. In some cases you will need to use a different tool for the restore phase than you used for the backup phase. For example you can use Reporting Services Configuration Manager to backup encryption keys from a previous version of Reporting Services but you need to use SharePoint Central administration or PowerShell to restore the encryption keys to a SQL Server 2014 Reporting Services SharePoint mode installation.

Databases you will see in the completed migration

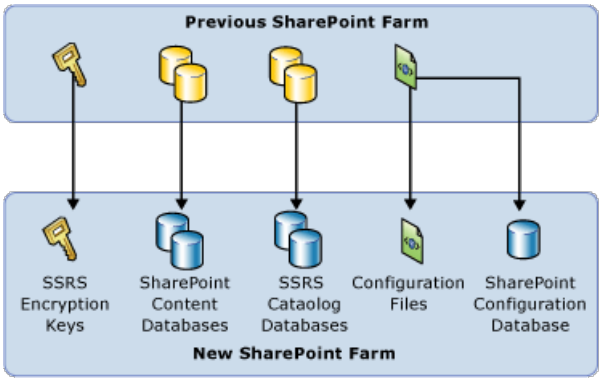
The following table describes the SQL Server Databases related to Reporting Services you will have after you successfully migrate your Reporting Services SharePoint installation:

DATABASE	EXAMPLE NAME	
Catalog database	ReportingService_[service application GUID] **(*)**	User migrates.
Temp database	ReportingService_[service application GUID]TempDB **(*)**	User migrates.
Alerting database	ReportingService_[service application GUID]_Alerting	Created when a Reporting Services service application is created.

(*) The example names shown in the table follow the naming convention SSRS uses when you create a new SSRS service application. If you are migrating from a different server, your catalog and tempDBs will have the names from the original installation.

Backup operations

This section describes the types of information you need to migrate and the tools or process you use to complete the backup.



	OBJECTS	METHOD	NOTES
1	Reporting Services encryption keys.	**Rskeymgmt.exe** or Reporting Services Configuration Manager. See Back Up and Restore Reporting Services Encryption Keys.	The noted tools can used for the backup but for the restore operation you will use the Reporting Services service application management pages or PowerShell.

	OBJECTS	METHOD	NOTES
2	SharePoint content databases.		Backup the database and detach it. See the section "Database attach upgrade " in Determine upgrade approach (SharePoint Server 2010) (https://technet.microsoft.com/library/cc263447.aspx).
3	SQL Server database that is the Reporting Servicescatalog database.	SQL Server database backup and restore or SQL Server database detach and attach.	
4	Reporting Services configuration files.	Simple file copy.	You only need to copy rsreportserver.config if you have made customizations to the file. Example of the default location of the files: C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\15\WebServices\Reporting Rsreportserver.config Rssvrpolicy.config Web.config for the Report Server ASP.NET application. Machine.config for ASP.NET.

Restore Operations

This section describes the types of information you need to migrate and the tools or process you use to complete the restore. The tools you use for restoring may be different than the tools you used for the backup.

Before you complete the restore steps, you need to install and configuring the new SharePoint Farm and Reporting Services SharePoint mode. For more information on a basic installation of Reporting Services SharePoint mode, see Reporting Services SharePoint Mode Installation (SharePoint 2010 and SharePoint 2013).

	OBJECTS	METHOD	NOTES

	OBJECTS	METHOD	NOTES
1	Restore SharePoint Content databases to the new farm.	SharePoint "Database attach upgrade" Method.	Basic Steps: 1) Restore the database on the new server. 2) Attach the content database to a web application by indicating the URL. 3) Get-SPWebapplication lists all web applications and the URLs. See the section "Database attach upgrade " in Determine upgrade approach (SharePoint Server 2010) (https://technet.microsoft.com/library/cc263447.aspx)and Attach databases and upgrade to SharePoint Server 2010 (https://technet.microsoft.com/library/cc263299.aspx).
2	Restore the SQL database that is the Reporting Services catalog database (ReportServer).	SQL Database backup and restore. **or** SQL Server database attached and detach.	The first time the database is used, Reporting Services will update the database schema as needed so it will work with the SQL Server 2014 environment.
3	Create a new Reporting Services service application.	SharePoint Central administration.	When you create the new service application, configure it to use the report server database you copied over. For more information on using SharePoint Central Administration, see the "Step 3: Create a Reporting Services Service Application" section in Install Reporting Services SharePoint Mode for SharePoint 2013. For examples using PowerShell, see the section "To create a Reporting Services Service Application using PowerShell" in Reporting Services SharePoint Service and Service Applications.

	OBJECTS	METHOD	NOTES
4	Restore Reporting Services configuration files.	Simple file copy.	Example of the default location of the files: C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\15\WebServices\Reporting.
5	Restore the Reporting Services encryption keys.	Restore the key back up file using the Reporting Services Service Application "SystemSettings" page. **or** PowerShell.	See the section "Key Management" in the topic Manage a Reporting Services SharePoint Service Application.

Additional Configuration

Depending on how your previous SharePoint environment was configured, you may need to complete one or more of the following:

1. Configure NTLM authentication for a Reporting Services service application. For more information, see Configure E-mail for a Reporting Services Service Application (SharePoint 2010 and SharePoint 2013)

Migrate from a SQL Server 2012 Deployment

In a multi-server farm, users will likely have the Content and Catalog databases on a different machine, in which case you really just need to add a new server with the Reporting Services service installed, to the SharePoint farm and then remove the old server. There should be no need to copy databases.

Backup Operations

1. Backup the Reporting Services encryption keys.
2. Backup the Reporting Services Service application in SharePoint central Administration (or use PowerShell). This will also backup the service application databases in SharePoint. See topic Backup and Restore Reporting Services SharePoint Service Applications
3. If you have an Unattended Execution account (UEA) and windows authentication, make a note of the credentials so you can use them for the restore process.
4. For more information, see Back up service applications in SharePoint 2013.

Restore Operations

1. Restore the Reporting Services service application using SharePoint Central Administration. You could also use PowerShell.
2. Restore Reporting Services encryption keys.

   See the section "Key Management" in the topic Manage a Reporting Services SharePoint Service Application
3. Configure the UEA and windows credentials on the service application.
4. For more information, see Restore service applications in SharePoint 2013.

Additional Resources

- Get started with upgrades to SharePoint 2013 (https://technet.microsoft.com/library/ee833948.aspx).
- Overview of the upgrade process to SharePoint 2013 (https://technet.microsoft.com/library/cc262483.aspx).

See Also

Upgrade and Migrate Reporting Services Migrate a Reporting Services Installation (Native Mode)

Native to SharePoint Migration (SSRS)

12/16/2019 • 2 minutes to read • Edit Online

You cannot upgrade or convert from one Reporting Services server mode to another. For example, you cannot upgrade or convert a Native mode report server to SharePoint mode. You cannot copy the report server databases between modes because they use different database schemas. You can migrate the content from one report server to another. The tools you use depend on the type of report server mode that is configured for the source and destination servers.

Applies to: Reporting Services Native mode | Reporting Services SharePoint mode

Reporting Services Migration tool

The tool supports content migration from a native mode Deployment to a SharePoint mode deployment. The tool does not support migration from SharePoint mode to SharePoint mode or from SharePoint mode to Native mode.

For more information, see Reporting Services Migration Tool (https://www.microsoft.com/download/details.aspx?id=29560).

Use Script to migrate content

If the migration tool does not meet your needs, you can manually migrate the report server data. The following is a summary of the steps to complete to migrate report items from a one Reporting Services deployment ot another. The approach supports either Native or SharePoint mode as the source or destination servers.

1. Backup and restore encryption keys. This is the key that is used to encrypt data. The encryption key is also used to encrypt passwords such as the passwords stored for data source connections. However, passwords cannot be migrated and you will need to enter them again in the destination environment.
2. **Reporting Services RSS scripts:** Write a Visual Basic script that calls Report Server Web service SOAP methods to copy data between databases. Use the **RS.exe** utility to run the script. Rs.exe is installed with Reporting Services.
   - Sample Reporting Services rs.exe Script to Migrate Content between Report Servers. The topics explains how to use the sample script you can download from CodePlex.
   - The sample rss script on CodePlex, Reporting Services RS.exe script that migrates content from one report server to another
   - Scripting and PowerShell with Reporting Services

The following table summarizes the Reporting Services objects you can migrate with scripts:

OBJECT	CAN BE SCRIPTED	COMMENTS
Reports	Yes	Following migration, to re-enter passwords for datasources.
Datasources	Yes	Following migration, Re-link reports to datasources.
Models	Yes	

OBJECT	CAN BE SCRIPTED	COMMENTS
Datasets	Yes	
Report Parts		Following migration, verify or update the path to the report parts.
Schedules	Yes	See the ListSchedules method Subscription and Delivery Methods
Subscriptions	yes	See the List Subscriptions method Subscription and Delivery Methods and the ChangeSubscriptionOwner method ChangeSubscriptionOwner
Snapshots		

Upgrade a Report Server Database

5/22/2019 • 4 minutes to read • Edit Online

The report server database provides 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.

Native Mode: In Reporting Services Native mode, the report server database is actually comprised of two database that have default names of "ReportServer and ReportServerTempDB".

SharePoint mode: In SQL Server 2012Reporting Services SharePoint mode the report server database is actually a collection of databases that is created for each instance of the Reporting Services service application.

Ways to Upgrade a Native Mode Report Server Database

The following list identifies the conditions under which a report server database is upgraded:

- SQL Server Setup upgrades a single instance of a report server. The report server 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 Manager 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 Manager no longer provides a separate Upgrade button or upgrade script. Those features are obsolete starting with 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.

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 Native mode report server database in the Reporting Services Configuration Manager that is an older version. 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 Upgrade Reports.

In addition to the report server database, a report server also uses a temporary database. The temporary database is upgraded automatically when you upgrade the report server database.

Permissions required to upgrade a Report Server Database

If you are upgrading a Reporting Services installation that includes a report server database, you may see an error message if the database upgrade is performed with insufficient permissions. 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 **sysadmin** permission to 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 Manager, 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 a Native Mode database With Scripts

You can use WMI scripts to upgrade a report server database. For more information, see GenerateDatabaseUpgradeScript Method (WMI MSReportServer_ConfigurationSetting)

See Also

Reporting Services Configuration Manager (Native Mode)
Create a Report Server Database (SSRS Configuration Manager)
Change Database Wizard (SSRS Native Mode)
Upgrade and Migrate Reporting Services
Migrate a Reporting Services Installation (Native Mode)

Upgrade Reports

5/22/2019 • 12 minutes to read • Edit Online

Report definition (.rdl) files are automatically upgraded in the following ways:

- When you open a report in Report Designer in SQL Server Data Tools (SSDT), the report definition is upgraded to the currently supported RDL schema. When you specify a SQL Server 2008 or SQL Server 2008 R2 report server in the project properties, the report definition is saved in a schema that is compatible with the target server.
- When you upgrade a Reporting Services installation to a SQL Server 2014 Reporting Services (SSRS) installation, existing reports and snapshots that have been published to a report server are compiled and 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. The report definition remains in the original schema.

Reports are not upgraded when you upload a report definition file directly to the report server or SharePoint site. Upgrading a report definition in SQL Server Data Tools is the only way to upgrade the .rdl file.

After a report is upgraded locally or on the report server, 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 Reporting Services Backward Compatibility.

For more information about new features for SQL Server 2014 Reporting Services (SSRS), see What's New (Reporting Services).

In this topic:

- Versions Supported by Upgrade
- Report Definition (.rdl) Files and Report Designer
- Published Reports and Report Snapshots
- Backward-Compatibility Mode
- Upgrading a Report with Subreports
- Upgrading a Report with Custom Report Items
- Convert CRI Dialog Box

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 2005
- SQL Server 2005 with Service Pack 1
- SQL Server 2005 with Service Pack 2
- SQL Server 2008

- SQL Server 2008 R2
- SQL Server 2014

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 SQL Server Data Tools (SSDT), if the report was created for a previous namespace, Report Designer automatically creates a backup file and upgrades the report to the current namespace. This is the only way you can upgrade a report definition file.

Deployment properties that you set can affect which schema the report definition file is saved in. For more information, see Deployment and Version Support in SQL Server Data Tools (SSRS).

You can upload an .rdl file created in an earlier version of Reporting Services to a SQL Server 2014 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.

> **NOTE**
> You cannot publish or upload a report that has the SQL Server 2014 report definition namespace to a SQL Server 2005 report server.

To identify the current RDL schema for a report, for a report server, or for Report Designer, see Find the Report Definition Schema Version (SSRS).

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 2014 report server in its original schema. If a report cannot be upgraded, the report runs in backward-compatibility mode.

Backward-Compatibility Mode

A report that is successfully upgraded is processed by the SQL Server 2014 Reporting Services (SSRS) 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 2014 Reporting Services (SSRS) report processor supports new features. If a report cannot be upgraded, you can still view the rendered report but new features are not available. To take advantage of the new features, a report must be successfully upgraded.

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 2014 Reporting Services (SSRS) 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 2014 Reporting Services (SSRS) 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 2014 Reporting Services (SSRS) 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.

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 2014 Reporting Services (SSRS) 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 2014 Reporting Services (SSRS) report server. The report is automatically upgraded on first use.
- A SQL Server 2005 Reporting Services report is opened in Report Designer in SQL Server Data Tools (SSDT). 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 Opening a Report in Report Designer later in this topic.

For information about identifying the current RDL namespace for a report server, SQL Server Data Tools, or a report, see Find the Report Definition Schema Version (SSRS).

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 2014 Reporting Services (SSRS) 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 SQL Server Data Tools.

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

CRI TYPE	REPORT SERVER UPGRADE ACTION
Third-party CRIs	Upgrade not performed. 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 chart data regions that are compatible with SQL Server 2014 Reporting Services (SSRS). Processed by the SQL Server 2014 Reporting Services (SSRS) report processor.
Dundas 2005 Gauge CRI with no unsupported features	Upgraded to the latest RDL schema. All Dundas 2005 Gauge CRIs are converted to gauge data regions that are compatible with SQL Server 2014 Reporting Services (SSRS) Processed by the SQL Server 2014 Reporting Services (SSRS) 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 2014 SQL Server Data Tools (SSDT), 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:

```
<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.

Convert CRI Dialog Box

This report contains custom report items (CRIs) with unsupported features. CRIs are extensions to the Report Definition Language (RDL) that support custom objects that display data in a report. CRIs include design-time and run-time components that are supplied by third-party software vendors.

> **NOTE**
> Choosing to support custom report items on a report server is a decision made by the system administrator. To view CRIs in a report, the CRI components must be installed on the report authoring client to preview a report and on the report server to view a published or uploaded report. For more information, see Custom Report Items and documentation from the third-party software vendor.

Some CRIs can be converted to report items in the new report definition format. For the list of CRIs that can be converted, see Upgrading Reports. Use the following list to decide whether to convert the CRIs in this report:

- **Yes** Choose **Yes** to convert all the CRIs in the report, where possible. Unsupported features in the CRIs cannot be upgraded and are removed from the report definition file. For the list of unsupported features, see Upgrading Reports. When you view the report, you might see differences in the way the CRI displays in the report.
- **No** Choose **No** when you do not want to convert the CRIs in the report. These CRIs cannot be displayed by the report processor in their current version. If your system administrator is planning to install a new version of the CRI from the third-party software vendor that is compatible with the new report definition

format, you should choose **No**. Until new versions are available, the CRIs display in the report as an empty text box with a red X.

In either case, the report is upgraded to the new report definition format and a backup copy of the original report is saved as *<Report Name>* - Backup.rdl. If you save the report in your report authoring tool, you are saving the upgraded report in the new report definition format. If you publish the report, the report is first saved on your computer, and then published to the report server. You are publishing the upgraded version of the report to the report server.

If you do not save the report, the original report remains unchanged. However, you cannot edit this report in the SQL Server 2014 version of SQL Server Data Tools or a report authoring environment that uses a newer report definition format. You can continue to run the original version of the report by uploading it to a SQL Server 2014 Reporting Services (SSRS) report server by using Report Manager. For more information, see Upload a File or Report (Report Manager).

For reports that you upload instead of publish to a report server, the report processor determines whether the report can be upgraded on first use. Reports that cannot be upgraded are processed in backward-compatibility mode, and continue to display as they did in the earlier version of Reporting Services.

See Also

Upgrade and Migrate Reporting Services
Breaking Changes in SQL Server Reporting Services in SQL Server 2014
Behavior Changes to SQL Server Reporting Services in SQL Server 2014
Discontinued Functionality to SQL Server Reporting Services in SQL Server 2014
Custom Report Items
Upgrade a Report Server Database

Backup and Restore Operations for Reporting Services

5/22/2019 • 3 minutes to read • Edit Online

This topic provides an overview of all data files used in a Reporting Services installation and describes when and how you should back up the files. Developing a backup and restore plan for the report server database files is the most important part of a recovery strategy. However, a more comprehensive recovery strategy would include backups of the encryption keys, custom assemblies or extensions, configuration files, and source files for reports and models.

Applies to: Reporting Services Native Mode | Reporting Services SharePoint Mode

Backup and restore operations are often used to move all or part of a Reporting Services installation:

- If you are moving just the report server databases, you can use backup and restore or attach and detach to relocate the databases on a different SQL Server instance. For more information, see Moving the Report Server Databases to Another Computer (SSRS Native Mode).
- Moving a Reporting Services installation to a new computer is called a migration. When you migrate an installation, you run Setup to install a new report server instance and then copy instance data to the new computer. For more information about migrating a Reporting Services installation, see the following topics:
  - Upgrade and Migrate Reporting Services
  - Migrate a Reporting Services Installation (SharePoint Mode)
  - Migrate a Reporting Services Installation (Native Mode)

Backing Up the Report Server Databases

Because a report server is a stateless server, all application data is stored in the **reportserver** and **reportservertempdb** databases that run on a SQL Server Database Engine instance. You can backup the **reportserver** and **reportservertempdb** databases using one of the supported methods for backing up SQL Server databases. Recommendations that are specific to the report server databases include the following:

- Use the full recovery model to backup the **reportserver** database.
- Use the simple recovery model to backup the **reportservertempdb** database.
- You can use different backup schedules for each database. The only reason to backup the **reportservertempdb** is to avoid having to recreate it if there is a hardware failure. In the event of hardware failure, it is not necessary to recover the data in **reportservertempdb**, but you do need the table structure. If you lose **reportservertempdb**, the only way to get it back is to recreate the report server database. If you recreate the **reportservertempdb**, it is important that it have the same name as the primary report server database.

For more information about backup and recovery of SQL Server relational databases, see Back Up and Restore of SQL Server Databases.

IMPORTANT

If your SQL Server 2014 report server is in SharePoint mode, there are additional databases to be concerned with, including SharePoint configuration databases and the Reporting Services alerting database. In SharePoint mode, three databases are created for each Reporting Services service application. The **reportserver**, **reportservertempdb**, and **dataalerting** databases. For more information see Backup and Restore Reporting Services SharePoint Service Applications

Backing Up the Encryption Keys

You should backup the encryption keys when you configure a Reporting Services installation for the first time. You should also backup the keys any time you change the identity of the service accounts or rename the computer. For more information, see Back Up and Restore Reporting Services Encryption Keys. For SharePoint mode report servers, see the "Key Management" section of Manage a Reporting Services SharePoint Service Application.

Backing Up the Configuration Files

Reporting Services uses configuration files to store application settings. You should backup the files when you first configure the server and after you deploy any custom extensions. Files to back up include:

- Rsreportserver.config
- Rssvrpolicy.config
- Rsmgrpolicy.config
- Reportingservicesservice.exe.config
- Web.config for both the Report Server and Report Manager ASP.NET applications
- Machine.config for ASP.NET

Backing Up Data Files

Backup the files that you create and maintain in Report Designer and Model Designer. These include report definition (.rdl) files, report model (.smdl) files, shared data source (.rds) files, data view (.dv) files, data source (.ds) files, report server project (.rptproj) files, and report solution (.sln) files.

Remember to backup any script files (.rss) that you created for administration or deployment tasks.

Verify that you have a backup copy of any custom extensions and custom assemblies you are using.

See Also

Report Server Database (SSRS Native Mode)
Reporting Services Configuration Files
rskeymgmt Utility (SSRS)
Copy Databases with Backup and Restore
Administer a Report Server Database (SSRS Native Mode)
Configure and Manage Encryption Keys (SSRS Configuration Manager)

Upgrade SQL Server Management Tools

11/5/2018 • 2 minutes to read • Edit Online

SQL Server 2014 supports upgrade from SQL Server 2005 and later. 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.

> **IMPORTANT**
> 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.

Known Upgrade Issues

Consider the following issues before you upgrade to SQL Server 2014:

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 Automated Administration Across an Enterprise.
- 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 2014.
- You can add components to an existing installation of SQL Server at the time that you upgrade to SQL Server 2014. For more information, see Upgrade to SQL Server 2014 Using the Installation Wizard (Setup).
- SQL Server Client Tools, such as SQL Server Management Studio, SQL Server Profiler, the Database Engine Tuning Advisor, sqlcmd, and osql are not upgraded to SQL Server 2014. Instead, Client Tools run side-by-side with tools from previous SQL Server versions. SQL Server 2014 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 2014.
- Data for jobs and alerts will be preserved during upgrade to SQL Server 2014.
- 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 2014. By default, Database Mail will be off after upgrade. Any schema updates should be reconciled with an update script after upgrade.

See Also

Supported Version and Edition Upgrades
Backward Compatibility
Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)

Upgrade How-to Topics

5/22/2019 • 2 minutes to read • Edit Online

The topics in this section describe how to upgrade a SQL Server stand-alone instance.

TOPIC	DESCRIPTION
Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)	The procedural topic to upgrade a stand-alone instance of SQL Server to SQL Server 2014.
Upgrade a SQL Server Failover Cluster Instance (Setup)	The procedural topic to upgrade an instance of a SQL Server failover cluster to SQL Server 2014.
Install SQL Server 2014 from the Command Prompt	Provides sample syntax for installing SQL Server 2014 at a command prompt.

See Also

Minimize Downtime for Mirrored Databases When Upgrading Server Instances
Supported Version and Edition Upgrades
Backward Compatibility

Upgrade to SQL Server 2014 Using the Installation Wizard (Setup)

11/28/2018 • 6 minutes to read • Edit Online

The SQL Server Installation Wizard provides a single feature tree for upgrade of SQL Server components. You can also install SQL Server 2014 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 2014.

For more information, see these topics:

- Supported Version and Edition Upgrades
- Work with Multiple Versions and Instances of SQL Server
- Upgrade a SQL Server Failover Cluster Instance (Setup)
- Install SQL Server 2014 from the Command Prompt
- Use the Copy Database Wizard

> **NOTE**
> Upgrade of an earlier version of SQL Server to SQL Server 2014 is not supported on a computer that is running Windows Server 2008 R2 Server Core SP1. For more information on Server Core installations, see Install SQL Server 2014 on Server Core.

Prerequisites

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, and is a local administrator.

Before upgrading the Database Engine, review the following topics:

- Upgrade to SQL Server 2014
- Hardware and Software Requirements for Installing SQL Server 2014
- Check Parameters for the System Configuration Checker
- Security Considerations for a SQL Server Installation
- SQL Server Database Engine Backward Compatibility

> **WARNING**
> Be aware that you cannot change the features to be upgraded, and you cannot add features during the upgrade operation. For more information about how to add features to an upgraded instance of SQL Server 2014 after the upgrade operation is complete, see Add Features to an Instance of SQL Server 2014 (Setup).

Procedure

To upgrade to SQL Server 2014

1. Insert the SQL Server installation media, and from the root folder, double-click Setup.exe. To install from

a network share, move to the root folder on the share, and then double-click Setup.exe.

2. The Installation Wizard starts the SQL Server Installation Center. To upgrade an existing instance of SQL Server, click **Installation** in the left-hand navigation area, and then click **Upgrade from SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, or SQL Server 2012**.

3. On the Product Key page, click an option 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 2014 and Supported Version and Edition Upgrades.

4. On the License Terms page, review the license agreement and, if you agree, select the **I accept the license terms** check box, and then click **Next**. To help improve SQL Server, you can also enable the feature usage option and send reports to Microsoft.

5. In the Global Rules window, the setup procedure will automatically advance to the Product Updates window if there are no rule errors.

6. The Microsoft Update page will appear next if the Microsoft Update check box in Control Panel\All Control Panel Items\Windows Update\Change settings is not checked. Putting a check in the Microsoft Update page will change the computer settings to include the latest updates when you scan for Windows Update.

7. On the Product Updates page, the latest available SQL Server product updates are displayed. If you don't want to include the updates, clear the **Include SQL Server product updates** check box. If no product updates are discovered, SQL Server Setup does not display this page and auto advances to the **Install Setup Files** page.

8. On the Install Setup Files page, Setup provides the progress of downloading, extracting, and installing the Setup files. If an update for SQL Server Setup is found, and is specified to be included, that update will also be installed.

9. In the Upgrade Rules window, the setup procedure will automatically advance to the Select instance window if there are no rule errors.

10. On the Select Instance page, specify the instance of SQL Server to upgrade. To upgrade Management tools and shared features, select **Upgrade shared features only**.

11. On the Select Features page, the features to upgrade will be preselected. A description for each component group appears in the right pane after you select the feature name.

    The prerequisites for the selected features are displayed on the right-hand pane. SQL Server Setup will install the prerequisite that are not already installed during the installation step described later in this procedure.

    > **NOTE**
    > If you have opted to upgrade the shared features by selecting **<Upgrade shared features only>** on the **Select Instance** page, all the shared features are preselected on the Feature Selection page. All the shared components are upgraded at the same time.

12. On the Instance Configuration page, specify the Instance ID for the instance of SQL Server.

    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, provide a value for the **Instance ID** textbox.

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

Installed instances - 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 2014.

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

14. On the Server Configuration - Service Accounts page, the default service accounts are displayed for SQL Server services. The actual services that are configured on this page depend on the features that you are upgrading.

    Authentication and login information 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 have to have to complete their tasks. For more information, see Configure Windows Service Accounts and Permissions.

    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**.

15. On the Full-Text Search Upgrade Options page, specify the upgrade options for the databases being upgraded. For more information, see Full-Text Search Upgrade Options.

16. The Feature Rules window will automatically advance if all rules pass.

17. The Ready to Upgrade page displays a tree view of installation options that were specified during Setup. To continue, click **Install**. SQL Server Setup will first install the required prerequisites for the selected features followed by the feature installation.

18. During installation, the progress page provides status so that you can monitor installation progress as Setup continues.

19. 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**.

20. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you have finished with Setup. For more information about Setup log files, see View and Read SQL Server Setup Log Files.

Next Steps

After you upgrade to SQL Server, complete the following tasks:

- **Register your servers** - Upgrade removes registry settings for the previous instance of SQL Server. After you upgrade, 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.
- **Configure your new SQL Server installation** - To reduce the attackable surface area of a system, SQL Server selectively installs and enables key services and features. For more information about surface area configuration, see the readme file for this release.

See Also

Upgrade to SQL Server 2014
Backward Compatibility

Upgrade to a Different Edition of SQL Server 2014 (Setup)

11/5/2018 • 3 minutes to read • Edit Online

SQL Server Setup supports edition upgrade among various editions of SQL Server 2014. For information about supported edition upgrade paths, see Supported Version and Edition Upgrades. Before you initiate the edition upgrade of an instance of SQL Server 2014, review the following topics:

- Features Supported by the Editions of SQL Server 2014
- Editions and Components of SQL Server 2014
- Compute Capacity Limits by Edition of SQL Server
- Hardware and Software Requirements for Installing SQL Server 2014

> **NOTE**
>
> **SQL Server in a clustered environment:** Running edition upgrade on one of the nodes of SQL Server cluster is sufficient. This node can be either active or passive, and the engine does not bring the resources offline during the Edition Upgrade. After the edition upgrade it is required to either restart the SQL Server instance or failover to a different node.

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 permissions on the remote share.

> **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.

Procedure

To upgrade to a different edition of SQL Server 2014

1. Insert the SQL Server installation media. From the root folder, double-click setup.exe or launch the SQL Server Installation Center from Configuration Tools. To install from a network share, locate the root folder on the share, and then double-click Setup.exe.
2. To upgrade an existing instance of SQL Server 2014 to a different edition, from the SQL Server Installation Center click **Maintenance**, and then select **Edition Upgrade**.
3. If Setup support files are required, SQL Server Setup installs them. If you are instructed to restart your computer, restart before you continue.
4. The System Configuration Checker runs a discovery operation on your computer. To continue, click **OK**.
5. 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 Editions and Components of SQL Server 2014 and Supported Version and Edition Upgrades.

6. 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**.
7. On the Select Instance page, specify the instance of SQL Server to upgrade.
8. The Edition Upgrade Rules page validates your computer configuration before the edition upgrade operation begins.
9. The Ready to Upgrade Edition page shows a tree view of installation options that were specified during Setup. To continue, click **Upgrade**.
10. During the edition upgrade process, the services need to be restarted to pick up the new setting. After edition upgrade, the Complete page provides a link to the summary log file for the edition upgrade. To close the wizard, click **Close**.
11. The Complete page provides a link to the summary log file for the installation and other important notes.
12. 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 View and Read SQL Server Setup Log Files.
13. 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.

In addition to the steps above, you may need to do the following if you upgraded from SQL Server Express:

- Users that were provisioned in SQL Server Express remain provisioned after the upgrade. Specifically, the BUILTIN\Users group remains provisioned. Disable, remove, or reprovision these accounts as needed. For more information, see Configure Windows Service Accounts and Permissions.
- Sizes and recovery mode for the tempdb and model system databases remain unchanged after the upgrade. Reconfigure these settings as needed. For more information, see Back Up and Restore of System Databases (SQL Server).
- Template databases remain on the computer after the upgrade.

See Also

Upgrade to SQL Server 2014
Backward Compatibility

Uninstall SQL Server 2014

12/3/2018 • 2 minutes to read • Edit Online

Follow the topics below to uninstall an existing instance of SQL Server 2014 completely, and prepare the system so that you can reinstall SQL Server.

In This Section

Uninstall an Existing Instance of SQL Server (Setup)
This topic describes how to manually uninstall a stand-alone instance of SQL Server.

Uninstall PowerPivot for SharePoint
This topic describes how to manually uninstall PowerPivot for SharePoint.

Uninstall Reporting Services
This topic describes how to uninstall Reporting Services servers for both SharePoint mode and Native mode servers.

Uninstall and Remove Master Data Services
This topic describes the process of uninstalling and removing Master Data Services from the local computer.

Remove Data Quality Server Objects
This topic describes how to manually remove the Data Quality Server objects after uninstalling either SQL Server or just the Data Quality Server component in Data Quality Services (DQS).

Related Sections

- Remove a SQL Server Failover Cluster Instance (Setup)
- Add or Remove Nodes in a SQL Server Failover Cluster (Setup)
- Drop a SQL Server 2014 Installation

See Also

Planning a SQL Server Installation
Install SQL Server 2014
Upgrade to SQL Server 2014

Uninstall an Existing Instance of SQL Server (Setup)

12/13/2018 • 3 minutes to read • Edit Online

This article describes how to 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.

> **IMPORTANT**
> To uninstall an instance of SQL Server, you must be a local administrator with permission to log on as a service.

> **NOTE**
> 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 Add or Remove Nodes in a SQL Server Failover Cluster (Setup)

Consider the following important information before you uninstall SQL Server:

- 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.
- If you have multiple instances of SQL Server, the SQL Server Browser uninstalls automatically when the last instance of SQL Server 2014 is uninstalled.

  If you want to uninstall all components of SQL Server 2014, you must uninstall the SQL Server Browser component manually from **Programs and Features** in **Control Panel**.

Before You Uninstall

1. **Back up your data.** Although this is not a required step, 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:

   - Master.mdf
   - Mastlog.ldf
   - Model.mdf
   - Modellog.ldf
   - Msdbdata.mdf
   - Msdblog.ldf
   - Mssqlsystemresource.mdf
   - Mssqlsustemresource.ldf
   - Tempdb.mdf

- Templog.ldf
- `ReportServer[$InstanceName]` (Thisis the Reporting Services default database.)
- ReportServer[$InstanceName]TempDB (This is the Reporting Services default temporary database.)

2. **Delete the local security groups.** Before you uninstall SQL Server, delete the local security groups for SQL Server components.
3. **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.
4. **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.

To Uninstall an Instance of SQL Server

1. To begin the uninstall process, go to **Control Panel** and then **Programs and Features**.
2. Right click **MicrosoftSQL Server 2014** and select **Uninstall**. Then click **Remove**. This starts the SQL Server Installation Wizard.

   Setup Support Rules runs to verify your computer configuration. To continue, click **Next**.
3. 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**.
4. On the Select Features page, specify the features to remove from the specified instance of SQL Server.

   Removal rules runs to verify that the operation can complete successfully.
5. On the **Ready to Remove** page, review the list of components and features that will be uninstalled. Click **Remove** to begin uninstalling
6. Immediately after you uninstall the last SQL Server instance, the other programs associated with SQL Server will still be visible in the list of programs in **Programs and Features**. However, if you close **Programs and Features**, the next time you open **Programs and Features**, it will refresh the list of programs, to show only the ones that are actually still installed.

If the Uninstallation Fails

1. If the uninstallation process does not complete successfully, attempt to fix the problem that caused the uninstallation to fail. The following articles can help you understand the cause of the failed uninstallation:
   - How to identify SQL Server 2008 setup issues in the setup log files
   - View and Read SQL Server Setup Log Files
2. If you are unable to fix the cause of the uninstallation failure, you can contact Microsoft Support. In some cases, such as unintentional deletion of important files, reinstalling the operating system may be required before reinstalling SQL Server on the computer.

See Also

View and Read SQL Server Setup Log Files

Remove Data Quality Server Objects

12/10/2018 • 2 minutes to read • Edit Online

Uninstalling Data Quality Server from an instance of SQL Server, or completely removing an instance of SQL Server that has Data Quality Server does not delete some Data Quality Server objects, including the DQS databases. This implies that you do not lose your DQS data if you uninstall Data Quality Server using the SQL Server setup. You must manually delete these Data Quality Server objects after the uninstall process is complete.

NOTE

- Before uninstalling Data Quality Server, consider backing up all your existing knowledge bases by exporting it to a .dqsb file, and use the file later to import all the knowledge bases back to a new Data Quality Server installation. Exporting and importing of all DQS knowledge bases can only be done by running DQSInstaller.exe with appropriate command line parameters from the command prompt. For more information, see Export and Import DQS Knowledge Bases Using DQSInstaller.exe.
- Before deleting the DQS databases, consider backing up the databases if you want to preserve it, and use it later for restoring the data. For information about doing so, see Manage DQS Databases.

Uninstall Data Quality Server from a SQL Server Instance

If you are just uninstalling Data Quality Server from an instance of SQL Server, you must manually delete the following Data Quality Server objects from the SQL Server instance after the Data Quality Server uninstall process is complete:

- DQS_MAIN, DQS_PROJECTS, and DQS_STAGING_DATA databases.
- ##MS_dqs_db_owner_login## and ##MS_dqs_service_login## logins.
- DQInitDQS_MAIN stored procedure from the master database.

You can delete these objects in SQL Server Management Studio by right-clicking the object, and clicking **Delete** in the shortcut menu.

IMPORTANT

If you just uninstall Data Quality Server from a SQL server instance using the `-uninstall` command line parameter from the command prompt, all the DQS objects are deleted as part of the uninstall process. You do not have to delete them manually after uninstalling Data Quality Server. To uninstall Data Quality Server from command prompt, type the following command at the command prompt, and press ENTER:

```
dqsinstaller.exe -uninstall
```

Uninstall SQL Server Instance Containing Data Quality Server

If you are completely uninstalling the SQL Server instance that has Data Quality Server, you must manually delete the DQS_MAIN, DQS_PROJECTS, and DQS_STAGING_DATA databases from your computer after the uninstall process is complete. For a default SQL Server installation, the DQS_MAIN, DQS_PROJECTS, and DQS_STAGING_DATA databases files are available at C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA.

See Also

Uninstall an Existing Instance of SQL Server (Setup)
Uninstall SQL Server 2014

Uninstall and Remove Master Data Services

12/13/2018 • 2 minutes to read • Edit Online

To uninstall the Master Data Services feature from an instance of SQL Server, follow the steps in Uninstall an Existing Instance of SQL Server (Setup) and specify Master Data Services as a feature to remove on the **Select Features** page. The uninstall process removes Master Data Services folders and files, and uninstalls Master Data Services Configuration Manager from the local computer.

To prevent data loss and avoid affecting other computers in the system, some items are not removed or changed by the uninstall process. Review the following table to determine whether to leave or remove items.

ITEM	DESCRIPTION
Folders and files	The uninstall process removes most folders and files from the installation path. The uninstall process does not remove the Master Data Services and MDSTempDir folders from the installation location. After the uninstall process is complete, you can manually delete these folders from the file system. For more information, see Folder and File Permissions (Master Data Services).
Master Data Services assemblies	The uninstall process removes Master Data Services assemblies from the Global Assembly Cache (GAC).
Database	The uninstall process does not affect the Master Data Services database. The database remains intact in the instance of Database Engine so that you do not lose any data, including master data, model objects, user and group permissions, business rules, and so on. If you do not need the database, and do not anticipate connecting it to another Master Data Services web site or application in the future, you might want to delete the database from the instance of Database Engine that hosts it. For more information, see Delete a Database.
Master Data Manager and Web.config	The uninstall process removes the WebApplication folder from the file system. The WebApplication folder contains the web application files and Web.config file for Master Data Manager. ** **Important** ** Before you uninstall, you might want to copy the Web.config file to another location to preserve any custom settings or other information in the file. After the uninstall process completes, the Web.config file is not recoverable.

ITEM	DESCRIPTION
Internet Information Services (IIS) items	The uninstall process does not affect any application pools, web sites, or web applications in IIS on the local computer. Because the uninstall process removes the WebApplication folder and Web.config file for Master Data Manager, any Master Data Manager web applications that require those files will no longer serve content. If users try to access the web application, they will receive HTTP Error 500.19-Internal Server Error: "The requested page cannot be accessed because the related configuration data for the page is invalid." If you no longer require the web site or application, and the application pool serving your site or application, you can use an IIS tool to delete them. For more information, see IIS 7 Operations Guide on Microsoft TechNet.
MDS_ServiceAccounts group	After the uninstall process is complete, the **MDS_ServiceAccounts** Windows group and any service accounts added to the group remain. If you no longer need the group and accounts, you can remove them.
Registry	The uninstall process removes all Master Data Services registry keys from the Windows registry.

See Also

Install Master Data Services

Uninstall PowerPivot for SharePoint

10/23/2019 • 9 minutes to read • Edit Online

Uninstalling a PowerPivot for SharePoint installation is a multi-step operation that includes preparing for uninstall, removing features and solutions from the farm, and removing program files and registry settings.

Applies to: SharePoint 2013 | SharePoint 2010

In this topic:

- Prerequisites
- Step 1: Pre-Uninstall Checklist
- Step 2: Remove Features and Solutions from SharePoint
- Step 3: Run SQL Server Setup to Remove Programs from the Local Computer
- Step 4: Uninstall the PowerPivot for SharePoint add-in
- Step 5: Verify Uninstall
- Step 6: Post-Uninstall Checklist

Prerequisites

- You must be a SharePoint farm administrator or service application administrator to uninstall features and solutions in the farm.
- You must be a SQL Server System Administrator and a member of the local Administrators group if you are also uninstalling the Database Engine.
- You must be an Analysis Services System Administrator and a member of the local Administrators group to uninstall Analysis Services and PowerPivot for SharePoint.

Step 1: Pre-Uninstall Checklist

PowerPivot data access will be disabled once the software that supports query and data processing is removed from the farm. As a first step, you should preemptively delete files and libraries that will no longer be operational. This lets you address any questions or concerns about 'missing data' before you uninstall the software.

1. Delete all PowerPivot workbooks, documents, and libraries that are associated with a PowerPivot for SharePoint installation. Neither the libraries nor the documents will function once the software is uninstalled.
    - Delete PowerPivot Gallery
    - Delete a PowerPivot Data Feed Library
2. Delete Excel workbooks or Reporting Services reports in other libraries that contain or reference PowerPivot data.
3. Remove any web part in a PerformancePoint dashboard that references PowerPivot data.
4. Review SharePoint permissions on existing sites and libraries to determine whether you need to change them. Some PowerPivot data access scenarios, particularly secondary data access via a URL connection

string to PowerPivot data in another workbook, require Read permissions, which is higher than the View permissions that are typically assigned to SharePoint users who only visit a site. If you no longer require Read permissions, you can reduce permissions accordingly.

5. Optionally, stop the services and wait several days before uninstalling the software. This step is not necessary for uninstall, but it gives you the option of temporarily resuming the service while you work out any data migration or technology substitution issues that you might have missed.

Step 2: Remove Features and Solutions from SharePoint

Use the PowerPivot Configuration Tool to remove PowerPivot services and applications from SharePoint.

- You must be a farm administrator, a server administrator on the Analysis Services instance, and **db_owner** on the farm's configuration database.
- Use the appropriate version of the configuration tool for the version of SharePoint. Do not use either tool with SQL Server 2008 R2 installations.
- Verify that the SharePoint Administration service is running.

1. **Run the Configuration tool:** Note the configuration tools are only listed only when PowerPivot for SharePoint is installed on the local server.On the **Start** menu, point to **All Programs**, click **Microsoft SQL Server 2014**, click **Configuration Tools**, and then click one of the following:
   - **PowerPivot for SharePoint 2013 Configuration**
   - **PowerPivot Configuration Tool**
2. Select **Remove Features, Services, Applications, and Solutions** and then click **OK**.
3. Optionally, expand the window to full size. You should see a button bar at the bottom of the window that includes **Validate**, **Run**, and **Exit** commands.
4. Review each action in the task list to understand what each one does.

   In **Remove PowerPivot Service Applications**, you are given the option of deleting application data associated with the service application. The application data is a SQL Server database that was created with the service application for the purpose of storing data refresh schedules, database instance information, usage data, and other data used by PowerPivot for SharePoint. It does not store user files, such as PowerPivot workbooks. Unless you have a specific reason to keep the application data (for example, if you have data retention policies related to data refresh or data access) you can delete the application database without removing any files that were created or saved by SharePoint users.

   To delete the database, select **Remove PowerPivot Service Applications** and then select the **Delete application data associated with this service application**.
5. Optionally, review detailed information in the **Output** tab or **Script** tab.

   The Output tab is a summary of the actions that will be performed by the tool. This information is saved in log files at:

   ```
   C:\Program Files\Microsoft SQL Server\120\Tools\PowerPivotTools\SPAddinConfiguration\Log
   ```

   The Script tab shows the PowerShell cmdlets or references the PowerShell script files that the tool will run.
6. Click **Validate** to check whether each action is valid. If **Validate** is not available, it means that all of the actions are valid for your system.
7. Click **Run** to perform all of the actions that are valid for this task. **Run** is available only after the validation check is passed. When you click **Run**, the following warning appears, reminding you that actions are

processed in batch mode: "All of the configuration settings that are flagged as valid in the tool will be applied to the SharePoint farm. Do you want to continue?"

8. Click **Yes** to continue.

Troubleshooting errors:

In the Configuration Tool, you can view error information in the Parameters pane for each action. For problems related to solution deployment or retraction, verify the SharePoint Administration service is started. This service runs the timer jobs that trigger configuration changes in a farm. If the service is not running, solution deployment or retraction will fail. Persistent errors indicate that an existing deployment or retraction job is already in the queue and blocking further action from the configuration tool. You can use the following PowerShell command to verify the service is running.

```
Get-Service | Where {$_.displayname -like "*sharepoint* administration*"}
```

To find and remove a deployment or retraction job that is already in the queue, do the following:

1. For all other errors, check the ULS logs. For more information, see Configure and View SharePoint Log Files and Diagnostic Logging (PowerPivot for SharePoint).

2. Start the SharePoint Management Shell as an administrator and then run the following command to view jobs in the queue:

   ```
   stsadm -o enumdeployments
   ```

3. Review existing deployments for the following information: **Type** is Retraction or Deployment, **File** is powerpivotwebapp.wsp or powerpivotfarm.wsp.

4. For deployments or retractions related to PowerPivot solutions, copy the GUID value for **JobId** and then paste it into the following command (use the Mark, Copy, and Paste commands on the Shell's Edit menu to copy the GUID):

   ```
   stsadm -o canceldeployment -id "<GUID>"
   ```

5. Retry the task in the configuration tool by clicking **Validate** followed by **Run**.

Alternatively, you can use PowerShell to remove features and solutions from the farm. For more information, see PowerShell Reference for PowerPivot for SharePoint.

Step 3: Run SQL Server Setup to Remove Programs from the Local Computer

Deleting program files requires that you run SQL Server Setup to uninstall the software. Uninstall removes both files and the registry entries that were created by Setup. You can use the Programs and Features page to uninstall the software. An installation of PowerPivot for SharePoint is part of a SQL Server installation.

You can uninstall part of an installation without impacting other SQL Server instances (or features in the same instance) that are already installed. For example, you can uninstall PowerPivot for SharePoint while leaving other components, such as Reporting Services or the Database Engine, installed.

1. Select **Microsoft SQL Server 2014 (64-bit)** from the program list.

2. Click **Uninstall/Change**.

3. Click **Remove**. This starts SQL Server Setup.

   From Setup, you can select the **PowerPivot** instance, and then select **Analysis Services** and **Analysis Services SharePoint Integration** to remove just that feature, leaving everything else in place.

Step 4: Uninstall the PowerPivot for SharePoint add-in

If your PowerPivot for SharePoint deployment has two or more servers and you installed the PowerPivot for SharePoint Add-in, then uninstall the PowerPivot for SharePoint add-in from each server where it was installed to completely uninstall all PowerPivot for SharePoint files. For more information, see Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013).

Step 5: Verify Uninstall

1. In Central Administration, in **Manage services on server**, connect to the server from which you uninstalled PowerPivot for SharePoint.
2. - If you uninstalled PowerPivot for SharePoint 2013, verify that **SQL Server PowerPivot System Service** no longer appear in the list.
   - If you uninstalled PowerPivot for SharePoint 2010, verify that **SQL Server Analysis Services** and **SQL Server PowerPivot System Service** no longer appear in the list.
3. After you uninstall the last PowerPivot for SharePoint server in the farm, do the following:

   a. In Application Management, in **Manage service applications**, verify that PowerPivot Service Application no longer appears in the list.

   b. In System Settings, in **Manage farm features**, verify that PowerPivot Integration Feature is no longer on the page. In **Manage farm solutions**, verify that the PowerPivot solutions no longer appear on the page.

   c. In Monitoring, in **Configure diagnostic logging** and in **Configure usage and health data collection**, verify that PowerPivot events and event categories no longer appear.

   d. In General Application Settings, verify that **PowerPivot Management Dashboard** is no longer on the page.

Step 6: Post-Uninstall Checklist

Use the following list to remove software and files that were not deleted during uninstall.

1. Delete all data files and subfolders from `C:\Program Files\Microsoft SQL Server\MSAS12.PowerPivot`, and then delete the folder. This step also deletes previously cached files in the DATA directory.
2. Delete all PowerPivot workbooks, documents, and libraries if you have not already done so.
   - Delete PowerPivot Gallery
   - Delete a PowerPivot Data Feed Library
3. In Secure Store Service, delete any target applications that contain stored credentials used by PowerPivot for SharePoint. Some, but not all, entries in Secure Store Service are deleted when you uninstall PowerPivot for SharePoint. Target applications created for the PowerPivot unattended data refresh account plus any target applications you created for data refresh still exist and must be deleted manually.

   In contrast, individual target applications that were auto-generated by the PowerPivot System Service are deleted automatically when PowerPivot is uninstalled.

4. In Control Panel, click **Programs**, and then click **Uninstall a program.** Uninstall any Analysis Services client libraries that are no longer used. Analysis Services ADOMD.NET and Microsoft SQL Server Analysis Management Objects are not removed when you uninstall PowerPivot for SharePoint. Because the libraries might be used by other programs that use Analysis Services data, SQL Server Setup does not uninstall them automatically. You must uninstall these client libraries individually if you no longer need them.

   Do not uninstall the SQL Server Reporting Services SharePoint 2010 Add-in unless you are following troubleshooting or installation instructions that specifically direct you to uninstall it. The Reporting Services Add-in is used by Access Services. It is installed by the SharePoint Products Preparation tool and should remain on the system to support functionality required by SharePoint.

   Do not uninstall the Analysis Services OLE DB provider. SharePoint installs the OLE DB provider as a prerequisite for Excel workbooks that connect to Analysis Services databases. PowerPivot for SharePoint installs a newer version, but this version is backwards compatible so you should leave it on the system to avoid data connection problems later.

See Also

Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013)
PowerPivot Configuration Tools

Uninstall Reporting Services

2/28/2020 • 2 minutes to read • Edit Online

Uninstalling Reporting Services does not remove the content you have created or configuration you have modified. However, if there is content you need after the uninstall is complete, it is recommended you make copies of content before you begin the uninstallation process.

Uninstall SharePoint Mode

When you uninstall Reporting Services SharePoint mode, the following are removed:

- Reporting Services service and service proxy.
- Files used for the Reporting Services installation.

The Reporting Services service applications are not removed. If you no longer want the service applications, delete them by using Windows PowerShell or SharePoint Central Administration.

The report items and related meta data are not removed. This information is contained in the content and configuration databases related to the Reporting Services service applications. The databases are not removed and you can manually migrate the databases to another installation of Reporting Services in SharePoint mode. If you no longer want the information, delete the databases. For more information, see Upgrade and Migrate Reporting Services.

The following are example names of the three Reporting Services databases that are not removed.

- **Report server database:** ReportingService_7f616e2d253040e8ab5653b3c09a065e
- **Report server temp database:** ReportingService_7f616e2d253040e8ab5653b3c09a065eTempDB
- **Report server alerting database:** ReportingService_7f616e2d253040e8ab5653b3c09a065e_Alerting

Uninstall the Add-in for SharePoint Products.

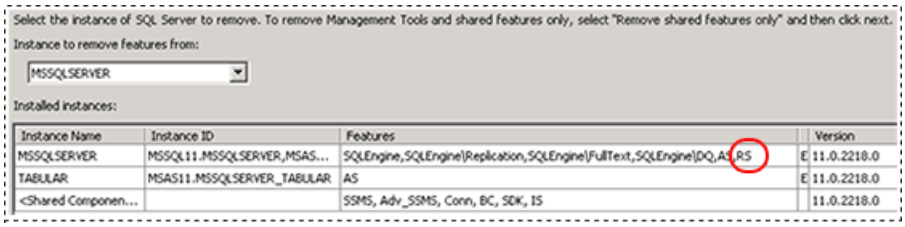
When you uninstall the add-in from a computer, you can choose to only uninstall the files or to also remove the Reporting Services feature from the farm. For information on uninstalling the Reporting Services add-in for SharePoint products, see Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013).

Uninstall Native Mode

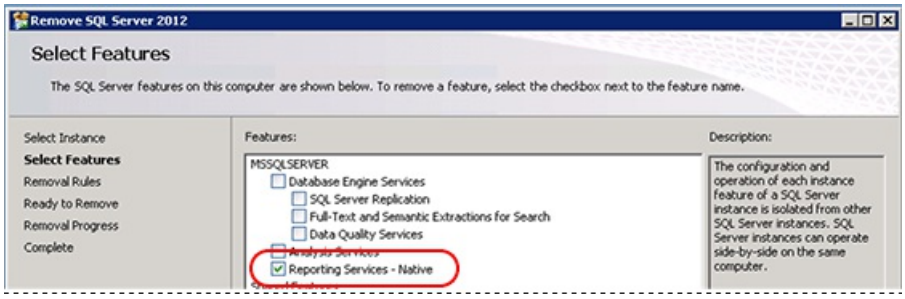
When you uninstall Reporting Services native mode, anything that was **created** or **modified** after the installation is left in place. For example database files, log files, Reporting Services configuration files, and content items such as reports and datasource files.

Reporting Services is an instance feature and therefore is not listed in Windows Control Panel, Programs and Features. To uninstall Reporting Services Native mode:

1. In Windows Control Panel, click **Programs and Features**.
2. In **Programs and Features** select **Microsoft SQL Server 2012**.
3. In the uninstall wizard, select the instance that includes the Reporting Services instance feature **RS**.



4. After you select the instance, select the Reporting Services feature.



5. Complete the wizard.

See Also

Uninstall an Existing Instance of SQL Server (Setup) Install or Uninstall the PowerPivot for SharePoint Add-in (SharePoint 2013) Install or Uninstall the Reporting Services Add-in for SharePoint (SharePoint 2010 and SharePoint 2013)