Microsoft® SQL Server® 2012 Database Engine
Common Criteria Evaluation (EAL4+)

Guidance Addendum

SQL Server 2012 Team

Author: Roger French

Version: 1.0
Date: 2013-01-11

Abstract

This document is the Guidance Addendum for the Common Criteria certification of the database engine of Microsoft® SQL Server® 2012.

Keywords

CC, SQL, Common Criteria, Guidance Addendum

This page intentionally left blank

Table of Contents

Page

1 INTRODUCTION 7

2 SCOPE OF THE EVALUATION 7

2.1 ASSUMPTIONS OF THE OPERATIONAL ENVIRONMENT 7
2.1.1 Trained administrator 8
2.1.2 General purpose computing capabilities 8
2.1.3 Physical Protection 9

3 INSTALLATION AND START-UP GUIDE 9

3.1 PREREQUISITES 9
3.1.1 Hardware Prerequisites 9
3.2 SOFTWARE PREREQUISITES 10
3.2.1 TOE Delivery 10
3.3 SQL SERVER 2012 INSTALLATION 11
3.3.1 Checking the integrity of the TOE parts 11
3.3.2 Installing the product 12
3.3.3 Installing the Documentation Contents 26
3.3.4 Checking the version of the product 31
3.3.5 Enabling the certified version 32
3.3.6 Installing the logon triggers 32

4 SQL SERVER BOOKS ONLINE 35

5 GUIDANCE ADDENDUM 36

5.1 SQL SERVER STARTUP FLAGS 36
5.2 ADMINISTRATION INTERFACE 38
5.3 USER INTERFACE 39
5.4 SECURITY FUNCTIONS RELEVANT FOR ADMINISTRATION AND USE OF THE TOE 39
5.4.1 Security Management 39
5.4.2 Access Control 47
5.4.3 Identification & authentication 48
5.4.4 Security Audit 48
5.4.5 Session Handling 48

6 SQL SERVER AUDIT 49

6.1 SERVER AUDIT 49
6.1.1 Examples of Use 49
6.1.2 Audit Record Contents 50
6.2 SERVER AUDIT SPECIFICATION 51
6.2.1 Examples of Use 54
6.3 DATABASE AUDIT SPECIFICATION 55
6.4 SECURITY RELEVANT EVENTS 55

7 RECOMMENDATIONS AND REQUIREMENTS FOR SECURE ADMINISTRATION, CONFIGURATION AND USAGE 56

7.1 RECOMMENDATIONS/REQUIREMENTS ABOUT SECURITY AUDIT 56
7.2 RECOMMENDATIONS AND FURTHER INFORMATION ABOUT ACCESS CONTROL 57
7.3 RECOMMENDATIONS/REQUIREMENTS ABOUT IDENTIFICATION AND AUTHENTICATION (SECURE PASSWORDS) 58
7.4 OTHER RECOMMENDATIONS AND REQUIREMENTS 59

8 APPENDIX 61

8.1 STORED PROCEDURES 61
8.1.1 sp_MSgetversion 61
8.1.2 xp_dirtree 61
8.1.3 xp_fileexist 61
8.1.4 xp_fixeddrives 62
8.1.5 xp_getnetname 62
8.1.6 xp_qv 62
8.1.7 xp_instance_regread 63
8.1.8 xp_regread 63
8.1.9 sp_enable_sql_debug 63
8.2 REFERENCES 64

List of Tables

Page

Table 1: Assumptions on the operational environment 8
Table 2: Hash values for deliverables 11
Table 3: Entry Points into Books Online 35
Table 4: Startup Options for "sqlservr.exe" 37
Table 5: Commands to add and delete logins 40
Table 6: Commands to add and delete users 41
Table 7: Commands to add and delete users from database and server groups 42
Table 8: Commands to create and destroy database groups 42
Table 9: Commands to create, start and stop audit 43
Table 10: Commands to include and exclude auditable event 44
Table 11: Commands to grant, revoke and deny permissions 47
Table 12: Audit Record 51
Table 13: Audit Events 54

List of Figures

Page

Figure 1: Successful verification of integrity 12
Figure 2: Installing SQL Server 2012 (I) 13
Figure 3: Installing SQL Server 2012 (II) 13
Figure 4: Installing SQL Server 2012 (III) 14
Figure 5: Installing SQL Server 2012 (IV) 15
Figure 6: Installing SQL Server 2012 16
Figure 7: Installing SQL Server 2012 (V) 17
Figure 8: Installing SQL Server 2012 (VI) 17
Figure 9: Installing SQL Server 2012 (VII) 18
Figure 10: Installing SQL Server 2012 (VIII) 19
Figure 11: Installing SQL Server 2012 (IX) 19
Figure 12: Installing SQL Server 2012 (X) 20
Figure 13: Installing SQL Server 2012 (XI) 21
Figure 14: Installing SQL Server 2012 (XII) 22
Figure 15: Installing SQL Server 2012 (XIII) 23
Figure 16: Installing SQL Server 2012 (XIV) 24

Figure 17: Installing SQL Server 2012 (XV) 25
Figure 18: Installing SQL Server 2012 (XVI) 26
Figure 19: Installation of content from the disk 27
Figure 20: Setting the path for the content 28
Figure 21: Selection of the content to install 28
Figure 22: Updating the Local Library was successful 29
Figure 23: After the restart the Documentations contents are updated 30
Figure 24: Choosing online or local help 30
Figure 25: Setting the preferred help experience 31
Figure 26: Extract of permission hierarchy 57

1 Introduction

This document has been created as part of the Common Criteria (CC) Evaluation of Database Engine of Microsoft SQL Server 2012. It covers the specific aspects that shall be considered when operating SQL Server 2012 in its certified version and extends the general guidance of the product given in Books Online. The document follows the following structure:

Chapter 2 of this document gives more details about the scope of the certification for SQL Server 2012 and the assumptions, which have been made about the environment of the TOE.

Chapter 3 of this document describes the steps for the installation process of the database engine of SQL Server 2012 in its certified version.

Chapter 4 introduces the concept of the SQL Server Books Online and provides the administrator and users with entry points for important aspects.

Chapter 5 contains the important aspects of the guidance, which are specific to the certified version of SQL Server 2012.

Chapter 6 introduces the concept and the important aspects of the audit mechanism of SQL Server 2012

Finally **chapter 7** gives requirements and recommendations for the secure operation of the TOE.

2 Scope of the evaluation

The Target of Evaluation (TOE), which has been addressed during this evaluation and certification process according to Common Criteria is one instance of the Database Engine of Microsoft SQL Server 2012 Enterprise Edition x64 (English) and its related guidance documentation. Please see [ST] for version information.

This database engine is the core component of the SQL Server Platform.

The TOE has been defined to be one instance of the database engine as it realizes the complete set of security functions as described in [ST, chapter 6.1].

Additional information about the certification process and related documents can be obtained via [WEB].

The following chapter describes the assumptions, which have been made about the environment of the TOE during evaluation, and which therefore have to be addressed during the start-up and operation of the TOE. It further explains how these assumptions can be addressed.

2.1 Assumptions of the Operational Environment

According to [ST] the following assumptions apply to the environment of use of the TOE.

Assumption	Description
A.NO_EVIL	Administrators are non-hostile, appropriately trained, and follow all administrator guidance.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing or storage repository capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

Table 1: Assumptions on the operational environment

The following chapters provide more details about the requirements which result out of the several assumptions for the secure administration of the TOE.

2.1.1 Trained administrator

To address the assumption A.NO_EVIL, authorized administrators shall read and follow all guidance documentation.

Further this assumption requires appropriate training for the administrators.

It is assumed that the ‘sa’ (sa stands for system administrator and represents the administrative user that has the highest level of permissions) has a commensurate level of knowledge to a MCDBA. Therefore, it is recommended that the 'sa' receive formal DBA training on the level of a MCDBA (or equivalent).

It is the responsibility of ‘sa’ to ensure that all other authorized administrators have sufficient knowledge and skills for the scope of their administrative permissions.

2.1.2 General purpose computing capabilities

The administrator of the TOE shall not install and/or use any general computing software on the machine where the TOE has been or will be installed other than those services necessary for the operation, administration and support of the DBMS. Only applications or services indispensable for the operation or installation of the TOE shall be run. The installation of the TOE has to be performed on a virgin OS. Beside the installation of the TOE itself a SQL-client may be installed on the machine to be used for administration. Also the SQL Server Management Studio and Books Online which ship together with the TOE can be used for administration (see also [AGD, ms-xhelp://?method=page&id=f289e978-14ca-46ef-9e61-e1fe5fd593be&product=SQLServer&productversion=110]).

However it should be noted that Management Studio has not been within the scope of evaluation. Specifically all functionality of the Graphical User Interface has not been evaluated. Thus – within the context of the evaluation Management Studio should just be seen as any other T-SQL client and the administrator shall ensure that the version of the client he is using is up to date and does not introduce any potential vulnerabilities.

Further other parts of the SQL Server 2012 Platform may be installed as long as they are needed to support the administration and operation of the TOE.

For example the SQL Server Profiler may be used to review the audit logs (see also [AGD, ms-xhelp:///?method=page&id=3AD5F33D-559E-41A4-BDE6-BB98792F7F1A&product=SQLServer&productversion=110])

The administrator shall further consider that also other users with accounts on the local machine the SQL Server is running on may have the possibility to install and use general computing software. This scenario however, is also covered by the assumption described in this chapter. Therefore the administrator shall also ensure that other users do not have the possibility to install and/or use general computing capabilities or untrusted software.

One way to enforce this would e.g. be to have no accounts for non-administrative users on the machine the SQL Server database engine is running on.

2.1.3 Physical Protection

It shall be ensured by the administrator that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information. Physical protection of the wire can provide an adequate level of security for the information transmitted between the clients and the TOE as all connections from clients to the TOE are unencrypted per default. It has to be mentioned that the maximum level of protection the TOE can provide for the user data which is stored in it depends on the physical security of the machine where the TOE is installed. With physical access to this machine an attacker could easily gain complete access to the user data which is stored in the database.

3 Installation and Start-up Guide

This chapter provides instructions for a secure setup, installation, and configuration of the TOE. In addition, this chapter describes the prerequisites for the installation process.

3.1 Prerequisites

3.1.1 Hardware Prerequisites

According to [ST] a machine that meets at least following criteria has to be available:

- AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support at 1.4 GHz or faster (Please note that IA64 CPUs are not supported for the certified version of the database engine of SQL Server 2012)
- 1 gigabyte (GB) of RAM or more

- Approximately 1500 MB of available hard-disk space for the recommended installation
- DVD-ROM drive
- SVGA (1,024x768) or higher-resolution video adapter and monitor
- Microsoft Mouse or compatible pointing device
- keyboard

Please note that additional disc space will be required for the recommended audit processes (Up to 10 GB in its default configuration).

3.2 Software Prerequisites

Before the installation of the TOE can start the following Operating System and additional prerequisites have to be installed on the machine:

- Windows Server 2008 R2 Enterprise Edition x64 (English) or Windows Server 2012 Standard Edition (English)[1]
- Microsoft .NET Framework 3.5 Service Pack 1[2] and Microsoft .NET Framework 4.

Further it is recommended to consider installing critical updates for those products before proceeding with the installation. However, it should be noted that any configuration of SQL Server that bases on a different configuration of the software prerequisites has not been considered during evaluation. In this context it should be noted that the installers for the .NET Framework and the Windows Installer do automatically receive updates if the machine is connected to the internet. In order to ensure that the exact version is installed the administrator shall therefore consider to disconnect the machine from the internet before installation.

3.2.1 TOE Delivery

The TOE is available as a download only and can be retrieved through Microsoft's Volume Licensing Service Center (VLSC). To download the TOE login to https://www.microsoft.com/licensing/servicecenter first. Then choose "SQL Server Enterprise Core 2012" in the download area. Download the ISO file and either burn it to a DVD or mount it. The installation description in the next chapter assumes that the ISO file is available on driver letter "D:".

[1] Both including Core installation variant

[2] This is delivered with Windows Server 2008 R2 Enterprise Edition and can be enabled in the Server Manager via the option „Add Feature".

3.3 SQL Server 2012 Installation

3.3.1 Checking the integrity of the TOE parts

It is assumed that the administrator has already successfully verified the integrity of the SQL Server 2012 Guidance Addendum (this document) as described on [WEB]. This activity includes the secure download of the FCIV tool.

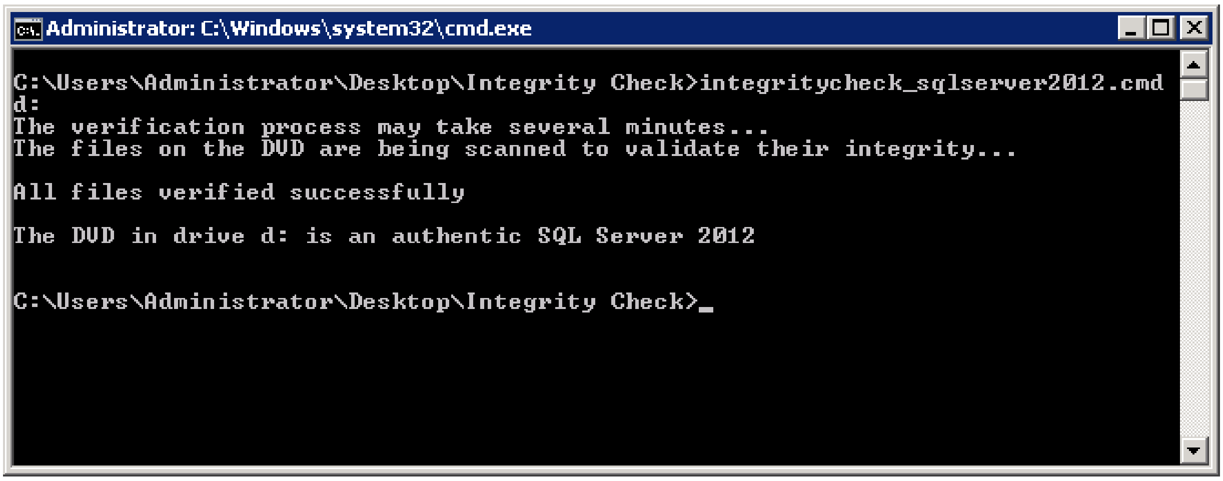
Before installing the product the administrator shall furthermore verify the integrity of the ISO image and all other downloads. This verification shall be done as follows:

1. Download the following files from [WEB] to the folder that contains the FCIV tool:
   - integritycheck_SQL2012.zip
   - permission_hierarchy.zip
   - SQLServer2012Documentation_December2012_EN.exe
   - Install_cc_triggers.sql
2. Download the following SP1 update file from http://www.microsoft.com/en-us/download/details.aspx?id=35575 to the folder that contains the FCIV tool:
   - SQLServer2012SP1-KB2674319-x64-ENU.exe
3. Open a command prompt and change to the directory where FCIV has been extracted. Run “fciv –sha1 <file>” for each downloaded file and compare the output hash with the following hashes:

File	SHA1 Hash
integritycheck_SQL2012.zip	208a209377d678944b8d03f749e092e8002d832f
permission_hierarchy.zip	578bf0aa2fb56e113118b6e00ed2aec75fe95a8f
SQLServer2012Documentation_December2012_EN.exe	4abfd4e620302ffafeb39091e4740a82381ec74c
Install_cc_triggers.sql	b52eafad4c436c5825cc0a17fc5d59d530dab67d
SQLServer2012SP1-KB2674319-x64-ENU.exe	58c45506605b17150983123ca1a3e020928d84b9

Table 2: Hash values for deliverables

4. Put the burnt DVD of SQL Server 2012 Enterprise into the local DVD drive and extract integritycheck_SQL2012.zip to the folder that contains the FCIV tool.
5. Open a command prompt and change to the directory to which the integritycheck_SQL2012.zip has been extracted.
6. Execute “integritycheck_sqlserver2012.cmd” and verify that the feedback matches the following picture:



```
C:\Users\Administrator\Desktop\Integrity Check>integritycheck_sqlserver2012.cmd
d:
The verification process may take several minutes...
The files on the DVD are being scanned to validate their integrity...

All files verified successfully

The DVD in drive d: is an authentic SQL Server 2012


C:\Users\Administrator\Desktop\Integrity Check>_
```

Figure 1: Successful verification of integrity

Note that the integrity check supports embedded licenses in the ISO image (the license files are excluded from the integrity check). The user can therefore use both per Core-based licensing and Client Access licensing (CAL).

3.3.2 Installing the product

The description in this chapter focuses on a typical way of installing the database engine of SQL Server 2012. For a more general overview over all options for the SQL Server setup please refer to [AGD, ms-xhelp:///?method=page&id=6ad23de1-2bab-4933-9122-c09f5565028d&product=SQLServer&productversion=110].

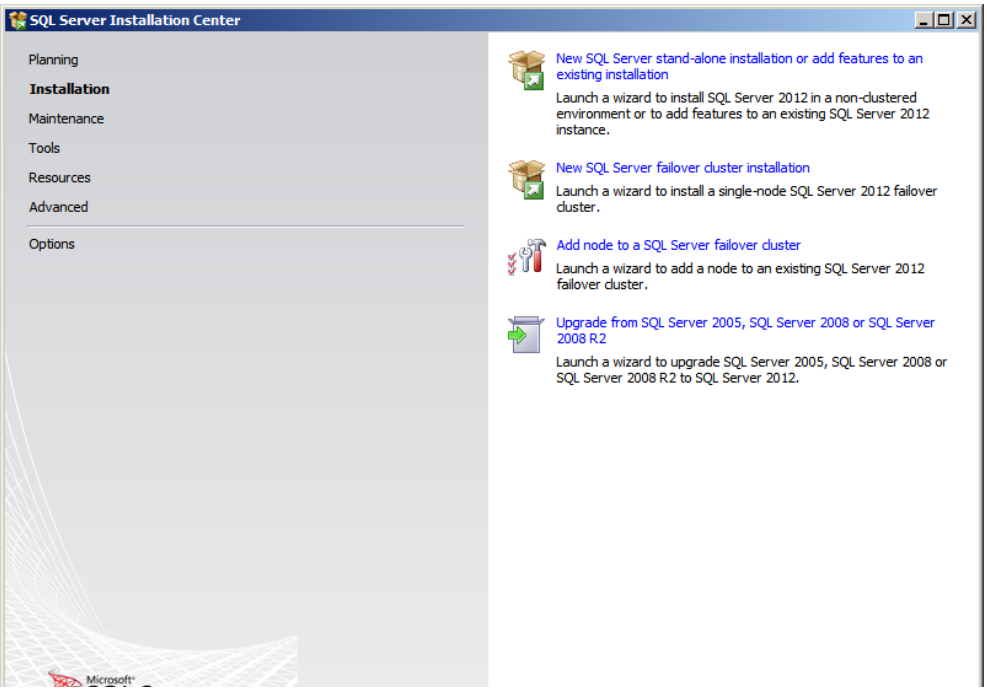
Please note that the installation procedure presented in this chapter belongs to the Enterprise Version of SQL Server 2012. The SQL Server Installation Wizard is Windows Installer-based. It provides a single feature tree for the installation of all SQL Server components.

To install SQL Server 2012 one has to insert the burnt DVD of the ISO image and double-click setup.exe in the root folder. This installer will by default install the version of SQL Server that fits to the installed Operating System (x64).

For local installations, Setup has to be run as administrator.

For the case that the .NET Framework which is required (see also chapter 3.2) is not installed, it has to be added as a new Feature in the Server Manager.

When the prerequisites are installed, the Installation Wizard will run the SQL Server Installation Center as seen in the following figure. To create a new installation of SQL Server 2012, click New SQL Server stand-alone installation or add features to an existing installation”.



SQL Server Installation Center

New SQL Server stand-alone installation or add features to an existing installation

Launch a wizard to install SQL Server 2012 in a non-clustered environment or to add features to an existing SQL Server 2012 instance.

New SQL Server failover cluster installation

Launch a wizard to install a single-node SQL Server 2012 failover cluster.

Add node to a SQL Server failover cluster

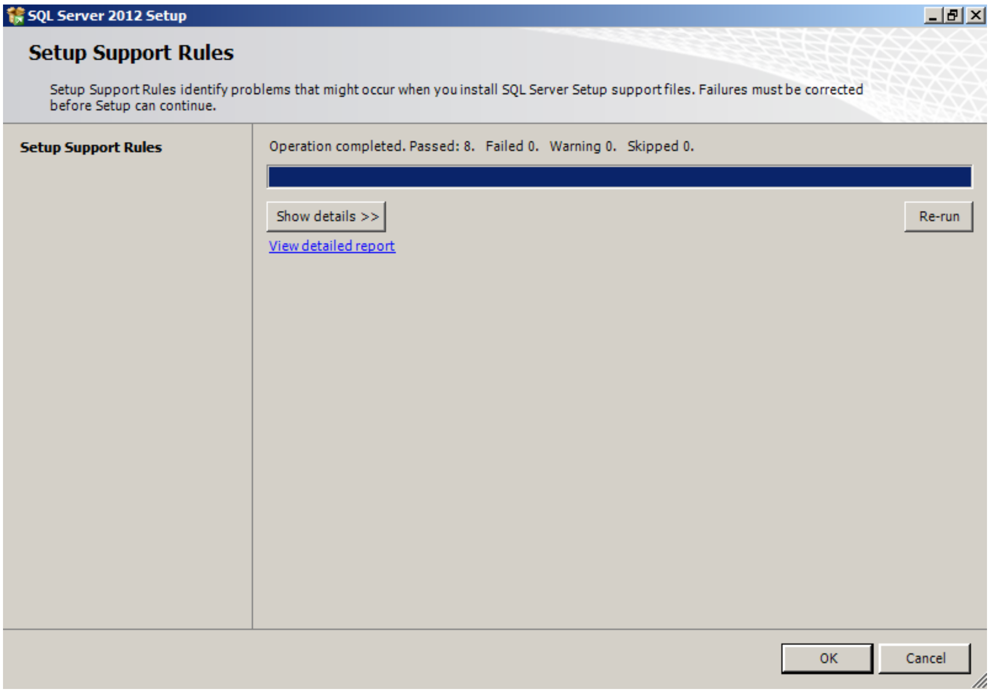
Launch a wizard to add a node to an existing SQL Server 2012 failover cluster.

Upgrade from SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2

Launch a wizard to upgrade SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2 to SQL Server 2012.

Figure 2: Installing SQL Server 2012 (I)

Next the System Configuration Checker (see Figure 3) will run a discovery operation on your computer. Setup log files have been created for the installation. For more information, see [AGD, ms-xhelp:///?method=page&id=8E712C15-6BFA-4D71-B303-9526101E5594&product=SQLServer&productversion=110].



SQL Server 2012 Setup

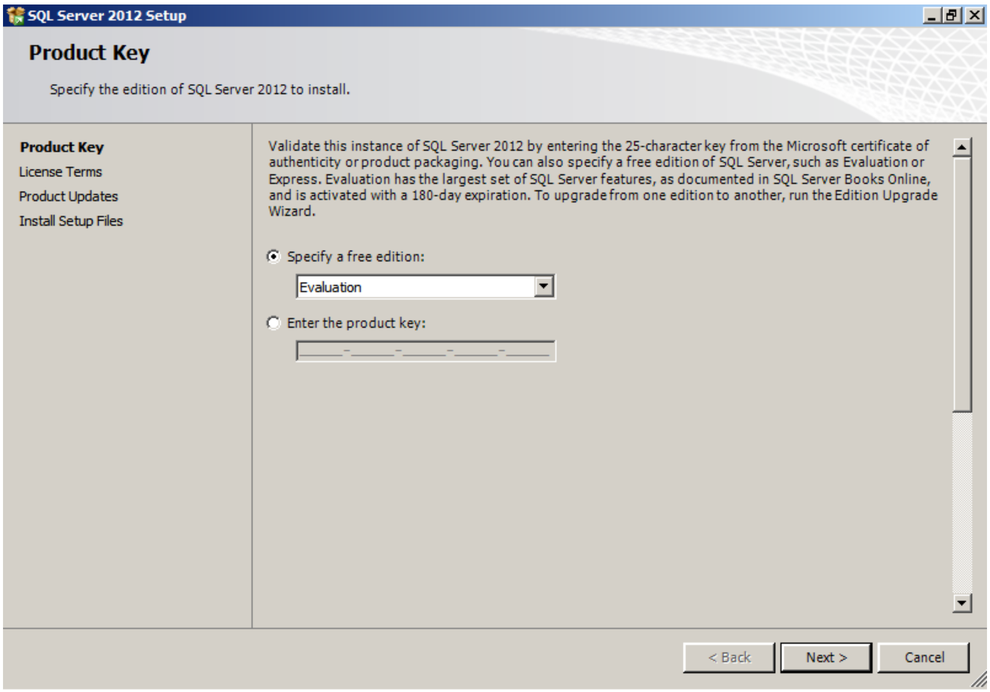
Setup Support Rules

Setup Support Rules identify problems that might occur when you install SQL Server Setup support files. Failures must be corrected before Setup can continue.

Operation completed. Passed: 8. Failed 0. Warning 0. Skipped 0.

Figure 3: Installing SQL Server 2012 (II)

On the Product Key page (see Figure 4), one selects a radio button to indicate whether installing a free edition of SQL Server, or a production version of the product that has a PID key. As only the Enterprise Edition of SQL Server has been certified one has to use a corresponding product key here.



SQL Server 2012 Setup

Product Key

Specify the edition of SQL Server 2012 to install.

Validate this instance of SQL Server 2012 by entering the 25-character key from the Microsoft certificate of authenticity or product packaging. You can also specify a free edition of SQL Server, such as Evaluation or Express. Evaluation has the largest set of SQL Server features, as documented in SQL Server Books Online, and is activated with a 180-day expiration. To upgrade from one edition to another, run the Edition Upgrade Wizard.

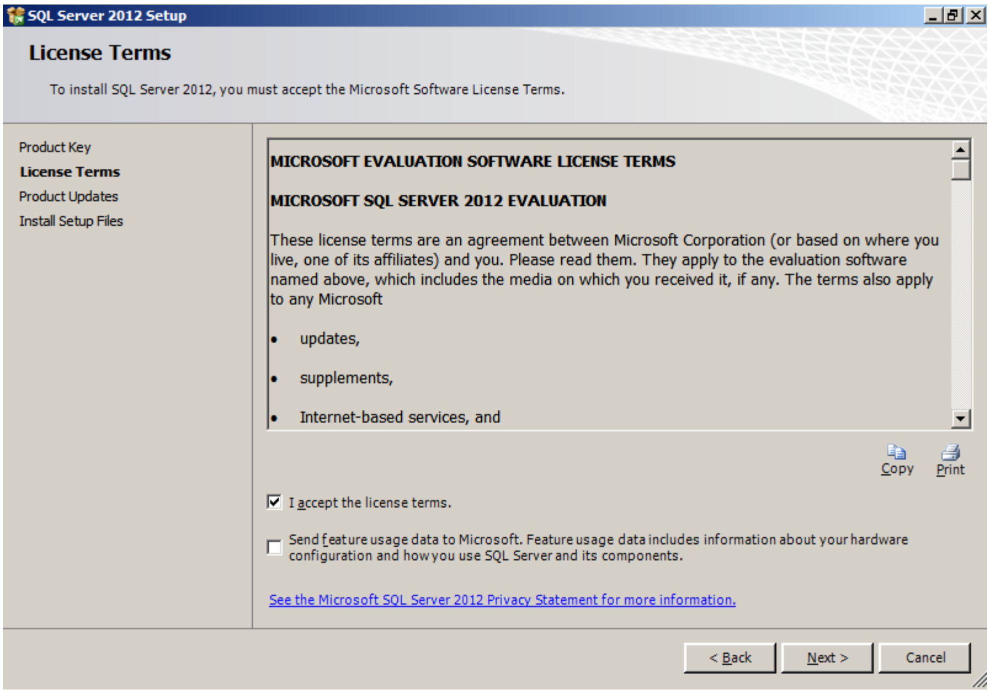
● Specify a free edition:

Evaluation

○ Enter the product key:

Figure 4: Installing SQL Server 2012 (III)

On the License Terms page (Figure 5), one shall read the license agreement, and then select the check box to accept the licensing terms and conditions.



License Terms

To install SQL Server 2012, you must accept the Microsoft Software License Terms.

MICROSOFT EVALUATION SOFTWARE LICENSE TERMS

MICROSOFT SQL SERVER 2012 EVALUATION

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the evaluation software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- updates,
- supplements,
- Internet-based services, and

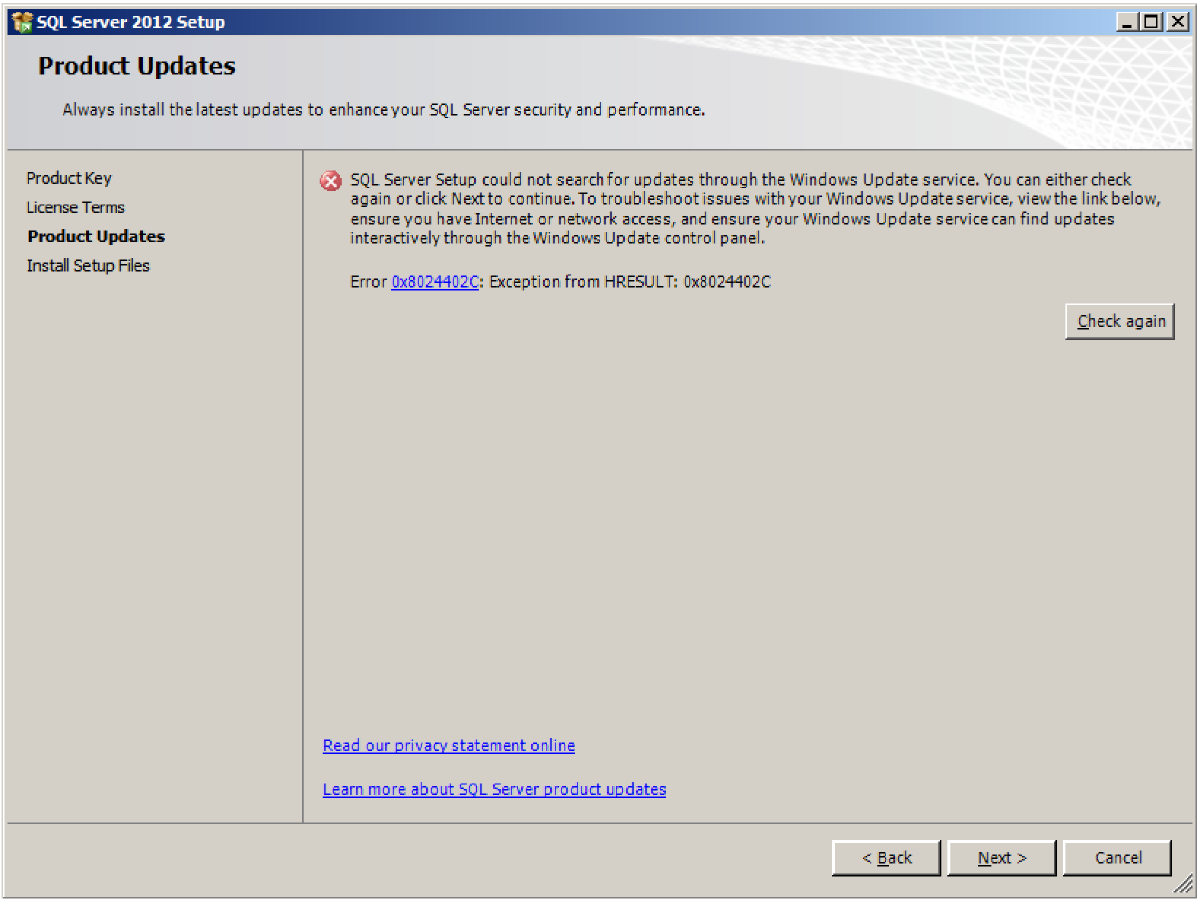
☑ I accept the license terms.

□ Send feature usage data to Microsoft. Feature usage data includes information about your hardware configuration and how you use SQL Server and its components.

See the Microsoft SQL Server 2012 Privacy Statement for more information.

Figure 5: Installing SQL Server 2012 (IV)

If all the other prerequisites have already been installed the Installation Wizard will then check for product updates through the Windows Update service. Due to a missing internet connection, the product update will fail (see Figure 6). For the certified version of SQL Server 2012 no automated product update shall be applied. Therefore, click ‘Next’ to skip this step.



Product Updates

Always install the latest updates to enhance your SQL Server security and performance.

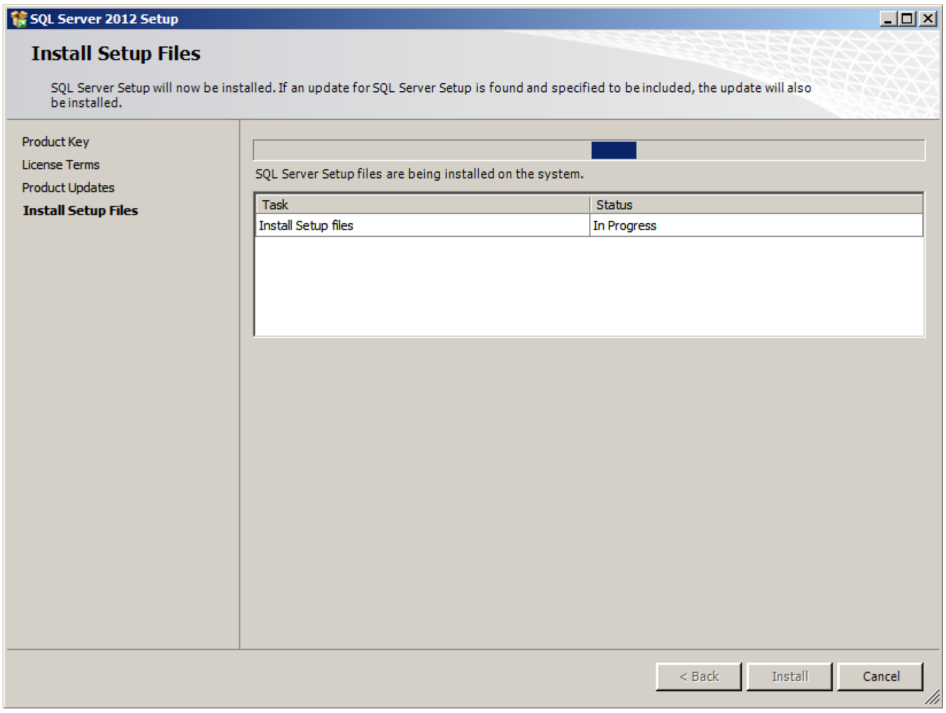
SQL Server Setup could not search for updates through the Windows Update service. You can either check again or click Next to continue. To troubleshoot issues with your Windows Update service, view the link below, ensure you have Internet or network access, and ensure your Windows Update service can find updates interactively through the Windows Update control panel.

Error 0x8024402C: Exception from HRESULT: 0x8024402C

Check again
Read our privacy statement online
Learn more about SQL Server product updates
< Back
Next >
Cancel

Figure 6: Installing SQL Server 2012

The Installation Wizard will then only copy the Setup Files to the hard disk as shown in Figure 7.



SQL Server 2012 Setup

Install Setup Files

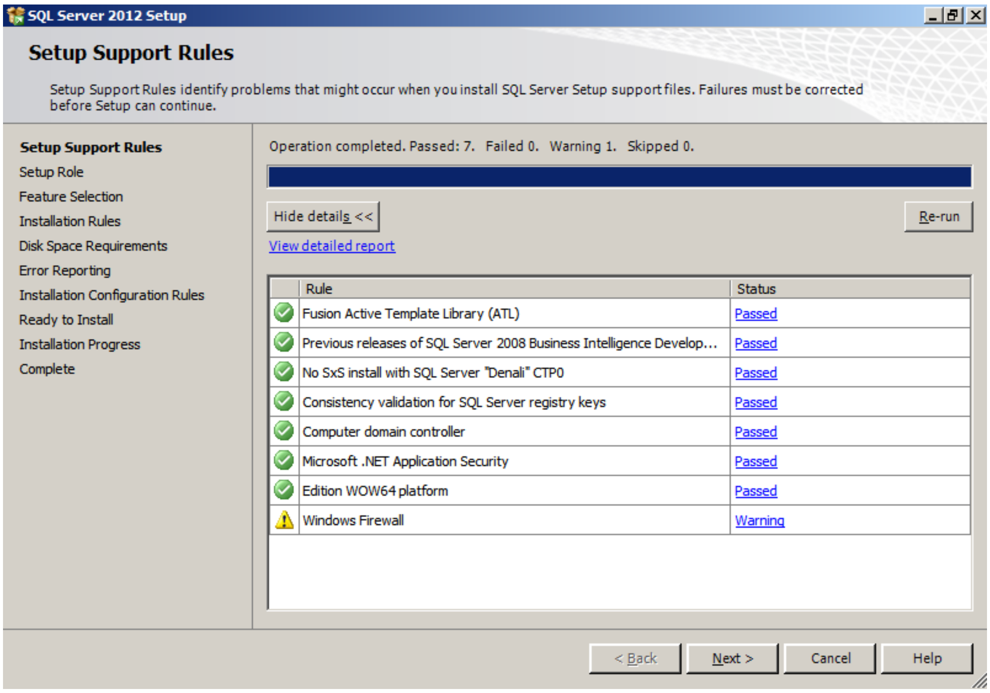
SQL Server Setup will now be installed. If an update for SQL Server Setup is found and specified to be included, the update will also be installed.

SQL Server Setup files are being installed on the system.

Task	Status
Install Setup files	In Progress

Figure 7: Installing SQL Server 2012 (V)

The System Configuration Checker will verify the system state of the machine before Setup continues (Figure 8). Warning messages shown by the Configuration Checker (e.g. the one to see in Figure 8 that is shown because the Windows Firewall is active) shall be carefully considered but do not prevent the further installation.



SQL Server 2012 Setup

Setup Support Rules

Setup Support Rules identify problems that might occur when you install SQL Server Setup support files. Failures must be corrected before Setup can continue.

Operation completed. Passed: 7. Failed 0. Warning 1. Skipped 0.

	Rule	Status
✓	Fusion Active Template Library (ATL)	Passed
✓	Previous releases of SQL Server 2008 Business Intelligence Develop...	Passed
✓	No SxS install with SQL Server "Denali" CTP0	Passed
✓	Consistency validation for SQL Server registry keys	Passed
✓	Computer domain controller	Passed
✓	Microsoft .NET Application Security	Passed
✓	Edition WOW64 platform	Passed
⚠	Windows Firewall	Warning

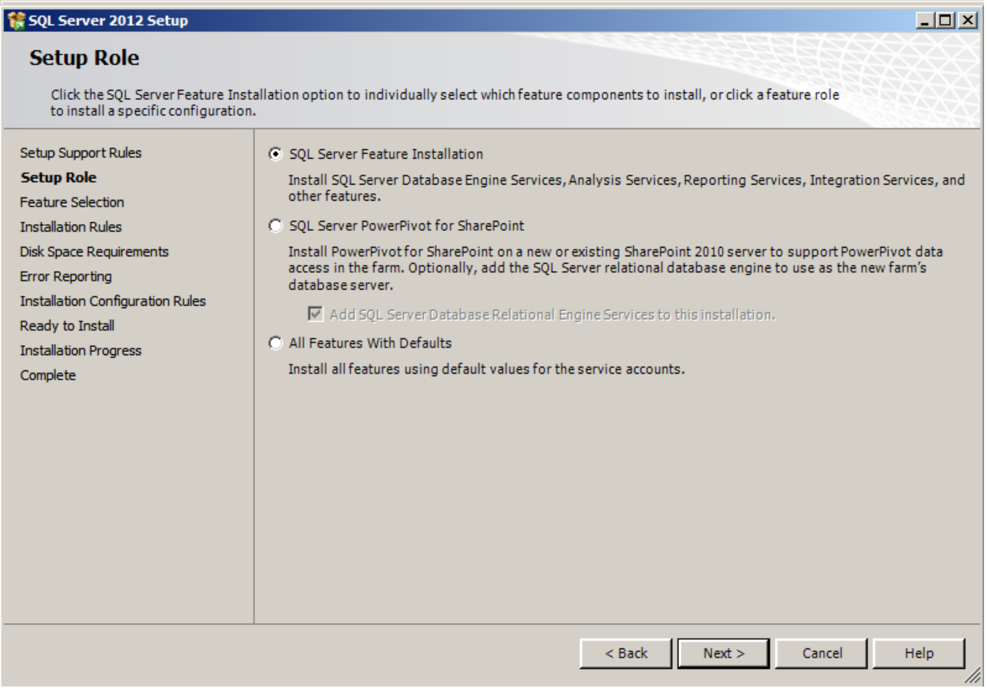
Figure 8: Installing SQL Server 2012 (VI)

Next, the setup process will ask for the Setup Role. To install a local SQL Server one chooses “SQL Server Feature Installation”.

On the Feature Selection page, one can select the components for installation. A description for each component group appears in the right-hand pane after selecting the feature name. One can select any combination of check boxes.

For the certified version of the database engine of SQL Server 2012 the following selection of components is recommended. It will install an instance of the database engine, tools for management and the documentation in form of Documentation Components. According to an assumption of the evaluation process other components may only be installed if they are indispensable for the operation of the database engine.

One can also specify a custom directory for shared components by using the field at the bottom of the Feature Selection page.



SQL Server 2012 Setup

Setup Role

Click the SQL Server Feature Installation option to individually select which feature components to install, or click a feature role to install a specific configuration.

Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Disk Space Requirements
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete

● SQL Server Feature Installation

Install SQL Server Database Engine Services, Analysis Services, Reporting Services, Integration Services, and other features.

○ SQL Server PowerPivot for SharePoint

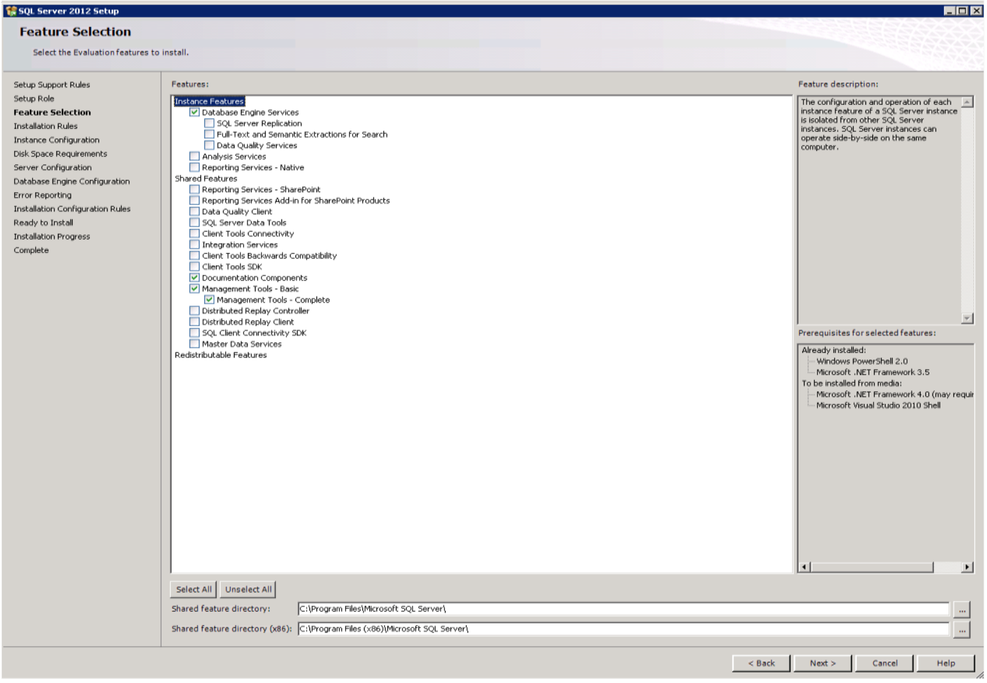
Install PowerPivot for SharePoint on a new or existing SharePoint 2010 server to support PowerPivot data access in the farm. Optionally, add the SQL Server relational database engine to use as the new farm's database server.

☑ Add SQL Server Database Relational Engine Services to this installation.

○ All Features With Defaults

Install all features using default values for the service accounts.

Figure 9: Installing SQL Server 2012 (VII)



Feature Selection

Select the Evaluation features to install.

Features:

Instance Features
- ☑ Database Engine Services
  - □ SQL Server Replication
  - □ Full-Text and Semantic Extractions for Search
  - □ Data Quality Services
- □ Analysis Services
- □ Reporting Services - Native

Shared Features
- □ Reporting Services - SharePoint
- □ Reporting Services Add-in for SharePoint Products
- □ Data Quality Client
- □ SQL Server Data Tools
- □ Client Tools Connectivity
- □ Integration Services
- □ Client Tools Backwards Compatibility
- □ Client Tools SDK
- ☑ Documentation Components
- ☑ Management Tools - Basic
  - ☑ Management Tools - Complete
- □ Distributed Replay Controller
- □ Distributed Replay Client
- □ SQL Client Connectivity SDK
- □ Master Data Services

Redistributable Features

Feature description:

The configuration and operation of each instance feature of a SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on the same computer.

Prerequisites for selected features:

Already installed:
- Windows PowerShell 2.0
- Microsoft .NET Framework 3.5

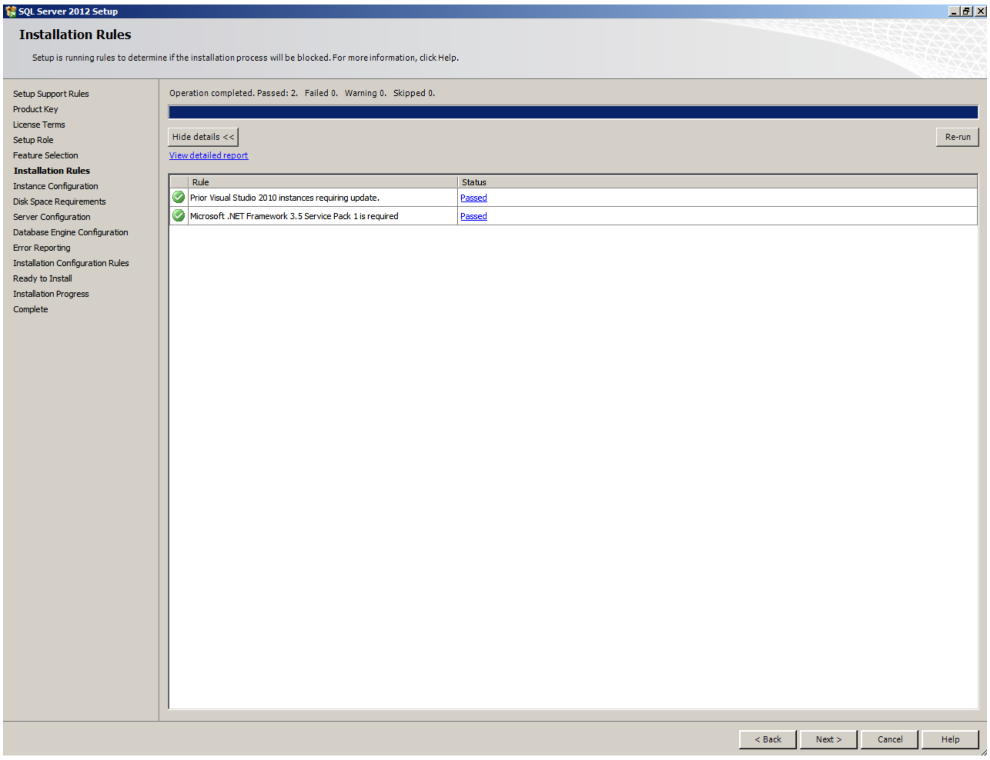
To be installed from media:
- Microsoft .NET Framework 4.0 (may requi
- Microsoft Visual Studio 2010 Shell

Shared feature directory: C:\Program Files\Microsoft SQL Server\

Shared feature directory (x86): C:\Program Files (x86)\Microsoft SQL Server\

Figure 10: Installing SQL Server 2012 (VIII)

Next, the System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features that have been selected.



Installation Rules

Setup is running rules to determine if the installation process will be blocked. For more information, click Help.

Operation completed. Passed: 2. Failed 0. Warning 0. Skipped 0.

	Rule	Status
✓	Prior Visual Studio 2010 instances requiring update.	Passed
✓	Microsoft .NET Framework 3.5 Service Pack 1 is required	Passed

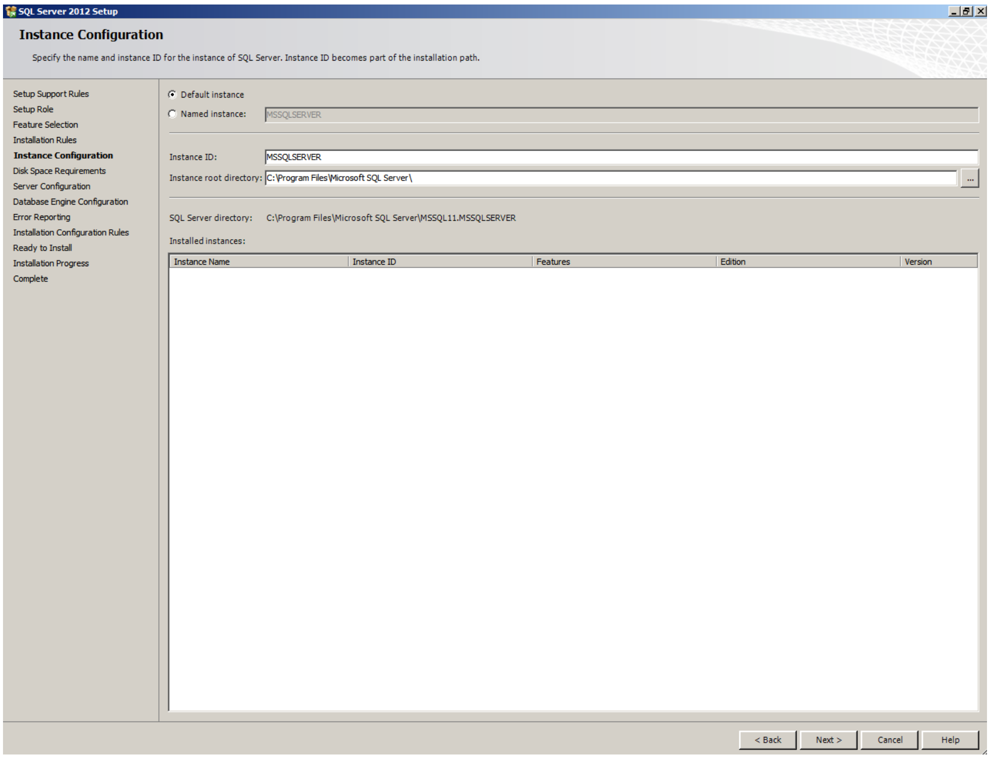
Figure 11: Installing SQL Server 2012 (IX)

On the Instance Configuration page (see Figure 12), one shall specify whether to install a default instance or a named instance. If an instance of SQL Server is not already installed, a default instance will be created unless you specify a named instance.

SQL Server supports multiple instances of SQL Server on a single server or processor, but only one instance can be the default instance. All others must be named instances. A computer can run multiple instances of SQL Server concurrently, and each instance runs independently of other instances.

Default or Named instance — Consider the following information when you decide whether to install a default or named instance of SQL Server:

- If you plan to install a single instance of SQL Server on a database server, it should be a default instance.
- Use a named instance for situations where you plan to have multiple instances on the same computer. A server can host only one default instance.
- Any application that installs SQL Server Express should install it as a named instance. This will minimizes conflict when multiple applications are installed on the same computer.



Instance Configuration

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

● Default instance

○ Named instance: MSSQLSERVER

Instance ID: MSSQLSERVER

Instance root directory: C:\Program Files\Microsoft SQL Server\

SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER

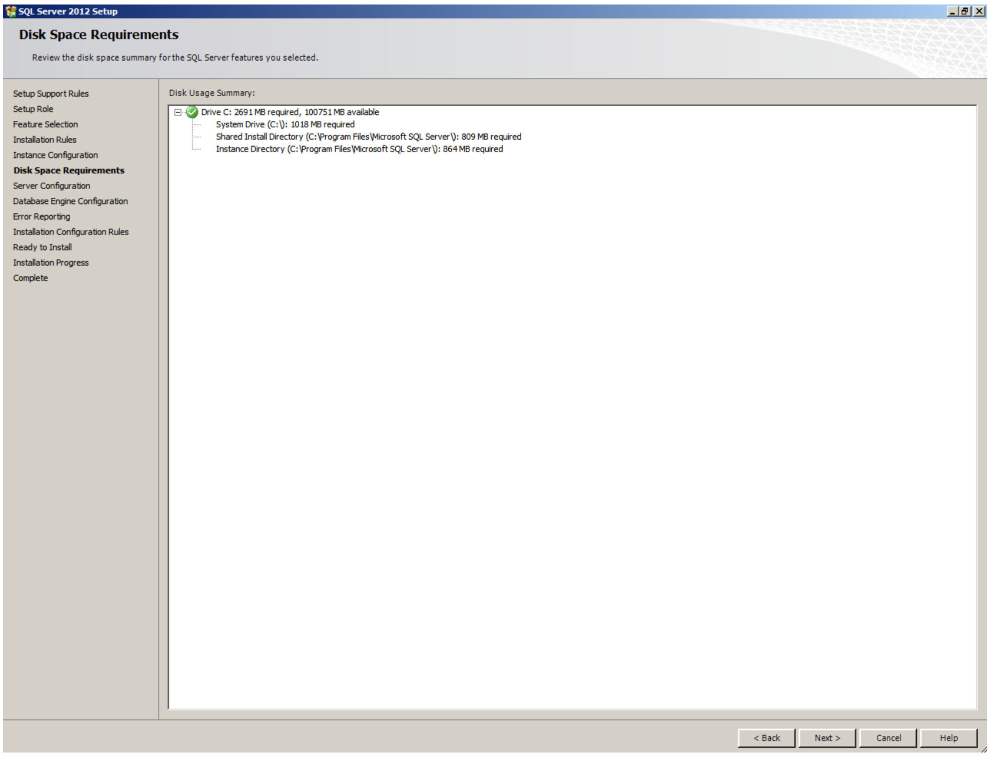
Installed instances:

Instance Name	Instance ID	Features	Edition	Version

Figure 12: Installing SQL Server 2012 (X)

The Disk Space Requirements page (see Figure 13) calculates the required disk space for the features you specify. It then compares the required space to the available disk space. For more information, see [AGD, ms-xhelp:///?method=page&id=09BCF20B-0A40-4131-

907F-B61479D5E4D8&product=SQLServer&productversion=110&locale=en-US&format=html&topicVersion=110].



SQL Server 2012 Setup

Disk Space Requirements

Review the disk space summary for the SQL Server features you selected.

Disk Usage Summary:

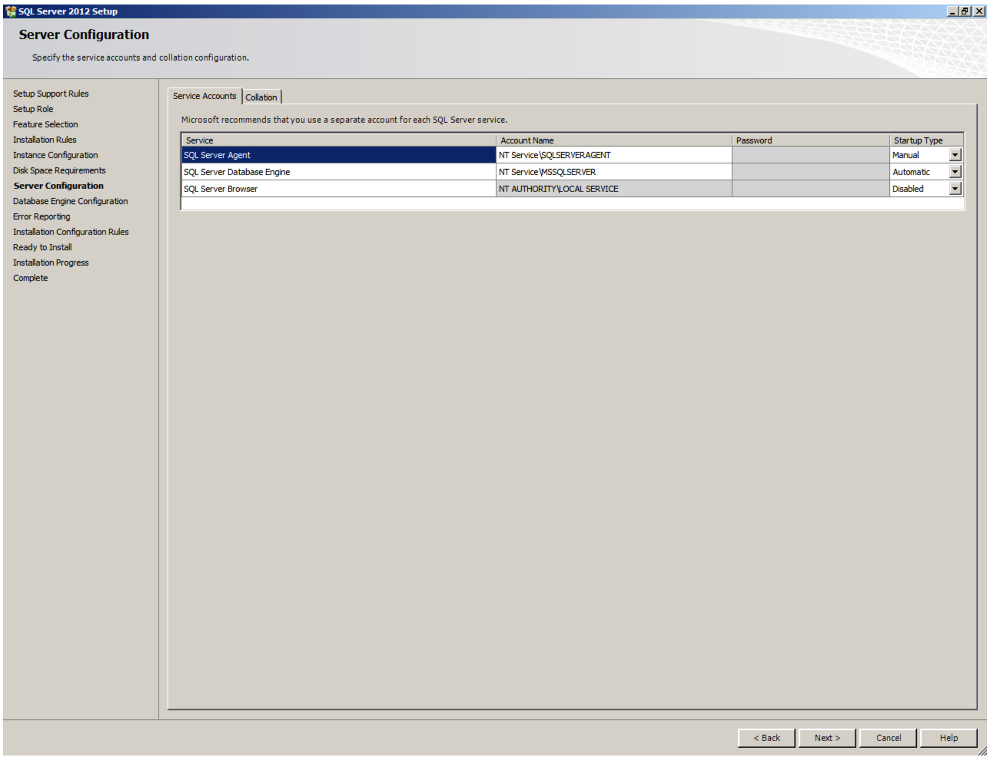
- Drive C: 2691 MB required, 100751 MB available
  - System Drive (C:\): 1018 MB required
  - Shared Install Directory (C:\Program Files\Microsoft SQL Server\): 809 MB required
  - Instance Directory (C:\Program Files\Microsoft SQL Server\): 864 MB required

Figure 13: Installing SQL Server 2012 (XI)

On the Server Configuration — Service Accounts page (see Figure 14), one specifies the login accounts for SQL Server services. The actual services that are configured on this page depend on the features that have been selected for installation.

One can assign the same login account to all SQL Server services, or one can configure each service account individually. One can also specify whether services start automatically, are started manually, or are disabled. It is recommended to 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. In general, it is recommended not to use the service accounts that are created for SQL Server services for any other purposes.

The Server Configuration — Collation tab can be used to specify non-default collations for the Database Engine and Analysis Services. For more information, see [AGD, ms-xhelp:\\?method=page&id=9F38EBA6-39B1-4F1D-BA24-EE4F7E2BC969&product=SQLServer&productversion=110&locale=en-US&format=html&topicVersion=110].



Server Configuration

Specify the service accounts and collation configuration.

Microsoft recommends that you use a separate account for each SQL Server service.

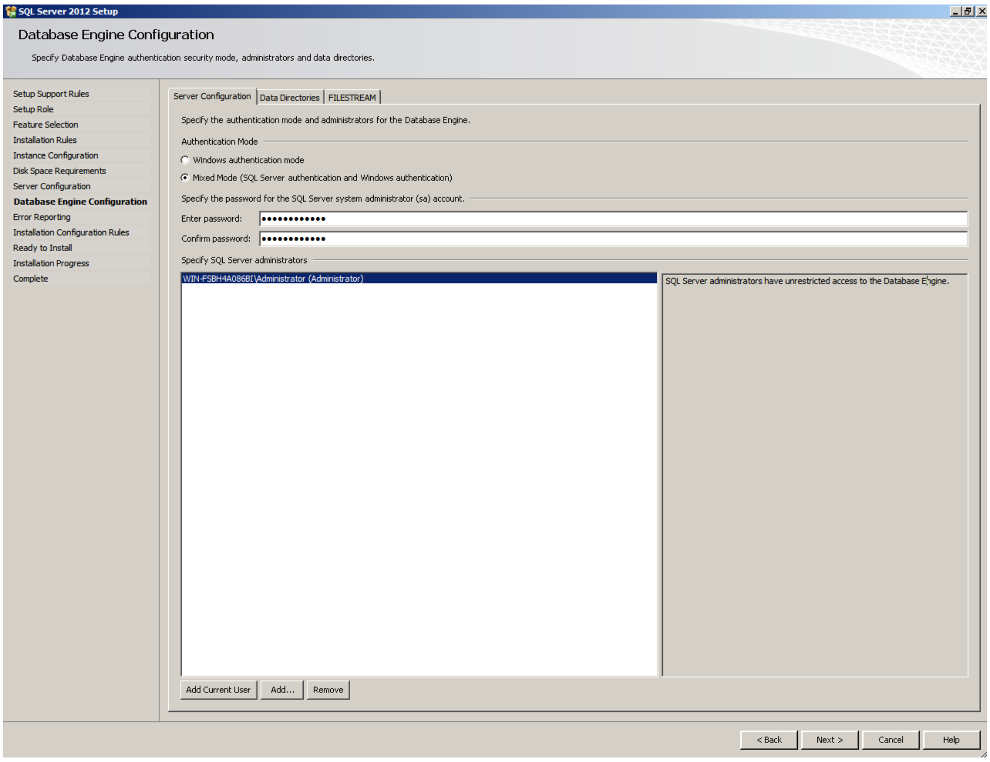
Service	Account Name	Password	Startup Type
SQL Server Agent	NT Service\SQLSERVERAGENT		Manual
SQL Server Database Engine	NT Service\MSSQLSERVER		Automatic
SQL Server Browser	NT AUTHORITY\LOCAL SERVICE		Disabled

Figure 14: Installing SQL Server 2012 (XII)

The Database Engine Configuration - Account Provisioning page (see Figure 15) can be used to specify the following:

- Security Mode — select Windows Authentication or Mixed Mode Authentication for the instance of SQL Server. If selecting Mixed Mode Authentication, one shall provide a strong password for the built-in SQL Server system administrator account. Please note that the SQL Server authentication will only be available if Mixed Mode authentication is chosen here.
- SQL Server Administrators — One must specify at least one system administrator for the instance of SQL Server. Adding the account under which SQL Server Setup is running can be done by clicking Add Current User. For more information, see [AGD, ms-xhelp:///?method=page&id=309B9DAC-0B3A-4617-85EF-C4519CE9D014&product=SQLServer&productversion=110].
- The Database Engine Configuration - Data Directories page (see Figure 15) can be used to specify non-default installation directories.

The Database Engine Configuration - FILESTREAM page (see Figure 15) can be used to enable FILESTREAM for your instance of SQL Server. For more information, see [AGD, ms-xhelp:///?method=page&id=1AD3400D-7FCD-40C9-87AE-F5AFC61E0374&product=SQLServer&productversion=110&locale=en-US&format=html&topicVersion=110].



Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

○ Windows authentication mode

● Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password: ••••••••••••

Confirm password: ••••••••••••

Specify SQL Server administrators

WIN-FSBH4A086BI\Administrator (Administrator)

SQL Server administrators have unrestricted access to the Database Engine.

Figure 15: Installing SQL Server 2012 (XIII)

On the Error and Usage Reporting page (see Figure 16), one specifies the information to be send to Microsoft that will help to improve SQL Server. By default, the option for error reporting is disabled. If you enable the Error Reporting feature, SQL Server is configured to send a report to Microsoft automatically if a fatal error occurs in any of the following SQL Server components:

- The Database Engine
- SQL Server Agent
- Analysis Services
- Reporting Services
- Integration Services
- Replication

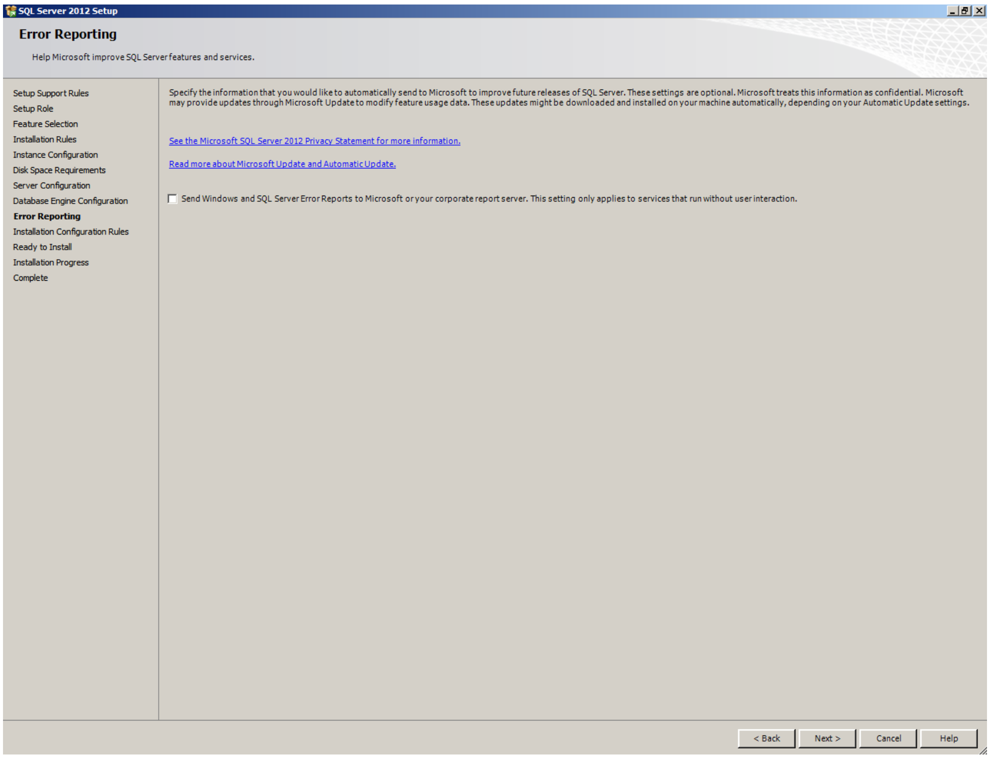
Information about errors is sent over a secure (https) connection to Microsoft, where it is stored with limited access. Alternatively, error reports can be sent to your own Corporate Error Reporting server.

Error reports contain the following information:

- The condition of SQL Server when the problem occurred.
- The operating system version and computer hardware information.
- Your Digital Product ID, which is not used to identify your license.
- The network IP address of your computer or proxy server.

- Information from memory or file(s) of the process that caused the error.

To disable Error or Feature Usage reporting for all instances of SQL Server and its components after Setup completes, go to the Error and Usage Report Settings dialog and clear the check boxes for Feature Usage. If Error Reporting is enabled for multiple components of SQL Server (the SQL Server Database Engine, Analysis Services, Reporting Services, and shared components) you can disable Error Reporting for each instance of an individual component as well as shared components, listed as Others.



SQL Server 2012 Setup

Error Reporting

Help Microsoft improve SQL Server features and services.

Specify the information that you would like to automatically send to Microsoft to improve future releases of SQL Server. These settings are optional. Microsoft treats this information as confidential. Microsoft may provide updates through Microsoft Update to modify feature usage data. These updates might be downloaded and installed on your machine automatically, depending on your Automatic Update settings.

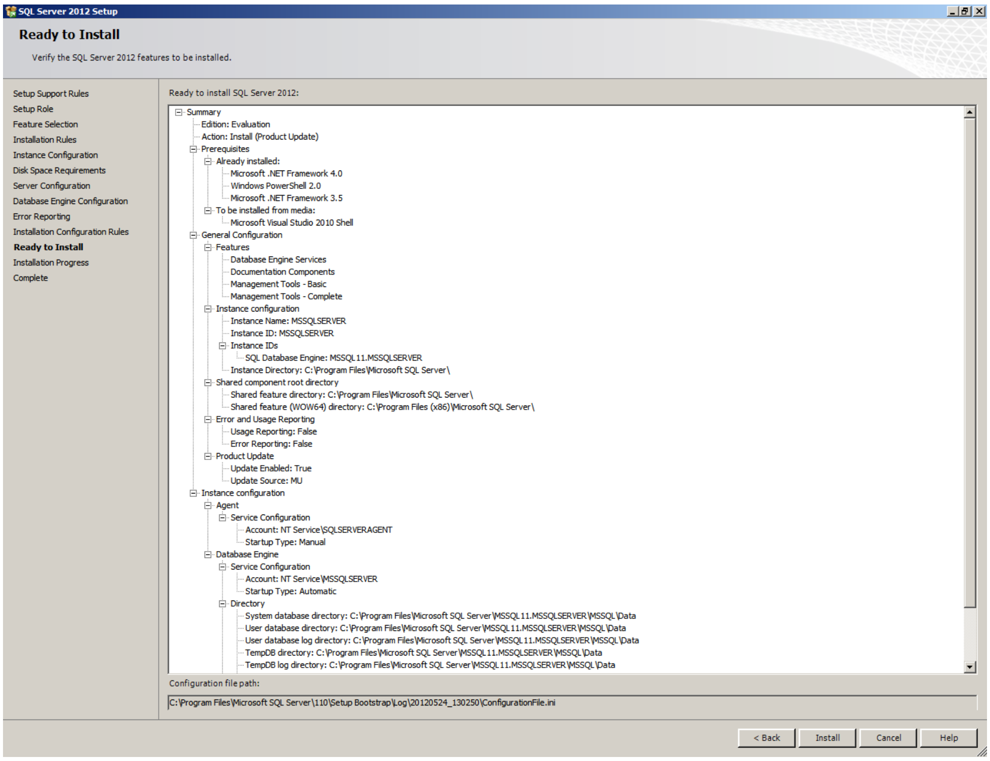
See the Microsoft SQL Server 2012 Privacy Statement for more information.

Read more about Microsoft Update and Automatic Update.

☐ Send Windows and SQL Server Error Reports to Microsoft or your corporate report server. This setting only applies to services that run without user interaction.

Figure 16: Installing SQL Server 2012 (XIV)

The Ready to Install page (see Figure 17) shows a tree view of installation options that were specified during Setup.



Ready to Install

Verify the SQL Server 2012 features to be installed.

Ready to install SQL Server 2012:

```
Summary
    Edition: Evaluation
    Action: Install (Product Update)
    Prerequisites
        Already installed:
            Microsoft .NET Framework 4.0
            Windows PowerShell 2.0
            Microsoft .NET Framework 3.5
        To be installed from media:
            Microsoft Visual Studio 2010 Shell
    General Configuration
        Features
            Database Engine Services
            Documentation Components
            Management Tools - Basic
            Management Tools - Complete
        Instance configuration
            Instance Name: MSSQLSERVER
            Instance ID: MSSQLSERVER
            Instance IDs
                SQL Database Engine: MSSQL11.MSSQLSERVER
            Instance Directory: C:\Program Files\Microsoft SQL Server\
        Shared component root directory
            Shared feature directory: C:\Program Files\Microsoft SQL Server\
            Shared feature (WOW64) directory: C:\Program Files (x86)\Microsoft SQL Server\
        Error and Usage Reporting
            Usage Reporting: False
            Error Reporting: False
        Product Update
            Update Enabled: True
            Update Source: MU
    Instance configuration
        Agent
            Service Configuration
                Account: NT Service\SQLSERVERAGENT
                Startup Type: Manual
        Database Engine
            Service Configuration
                Account: NT Service\MSSQLSERVER
                Startup Type: Automatic
            Directory
                System database directory: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data
                User database directory: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data
                User database log directory: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data
                TempDB directory: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data
                TempDB log directory: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data
```

Configuration file path:

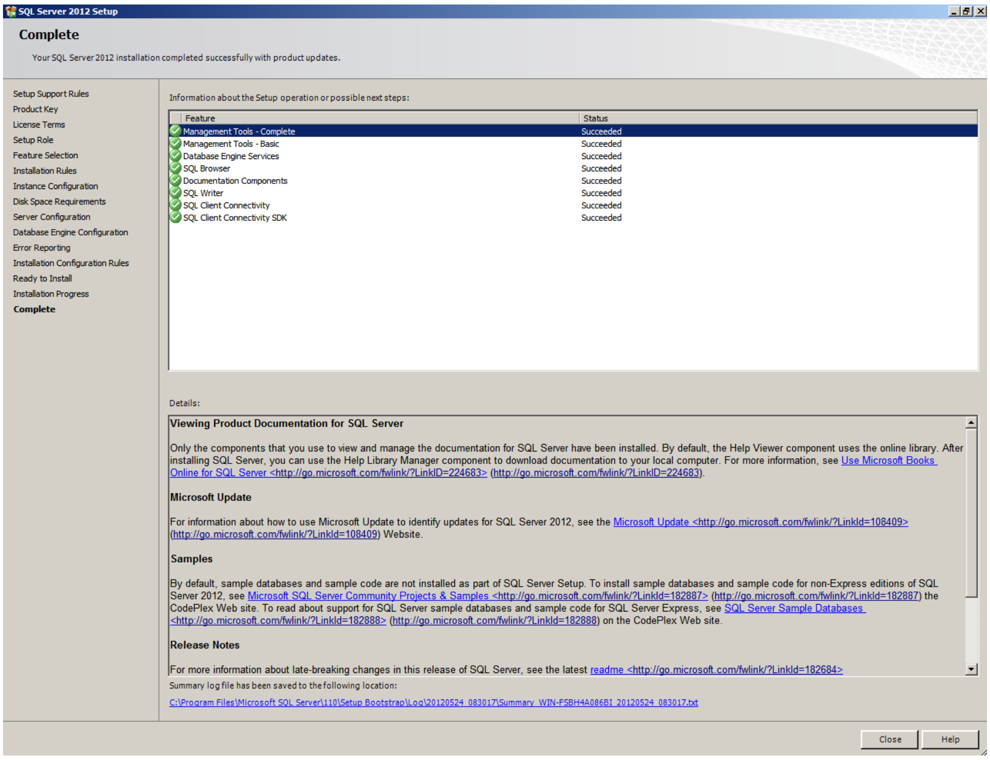
C:\Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log\20120524_130250\ConfigurationFile.ini

Figure 17: Installing SQL Server 2012 (XV)

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

After installation, the complete page (see Figure 18) provides a link to the summary log file for the installation and other important notes. The SQL Server installation process is finished after clicking Close.

If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup.



Complete

Your SQL Server 2012 installation completed successfully with product updates.

Information about the Setup operation or possible next steps:

Feature	Status
✓ Management Tools - Complete	Succeeded
✓ Management Tools - Basic	Succeeded
✓ Database Engine Services	Succeeded
✓ SQL Browser	Succeeded
✓ Documentation Components	Succeeded
✓ SQL Writer	Succeeded
✓ SQL Client Connectivity	Succeeded
✓ SQL Client Connectivity SDK	Succeeded

Details:

Viewing Product Documentation for SQL Server

Only the components that you use to view and manage the documentation for SQL Server have been installed. By default, the Help Viewer component uses the online library. After installing SQL Server, you can use the Help Library Manager component to download documentation to your local computer. For more information, see Use Microsoft Books Online for SQL Server <http://go.microsoft.com/fwlink/?LinkID=224683> (http://go.microsoft.com/fwlink/?LinkID=224683).

Microsoft Update

For information about how to use Microsoft Update to identify updates for SQL Server 2012, see the Microsoft Update <http://go.microsoft.com/fwlink/?LinkId=108409> (http://go.microsoft.com/fwlink/?LinkId=108409) Website.

Samples

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 2012, see Microsoft SQL Server Community Projects & Samples <http://go.microsoft.com/fwlink/?LinkId=182887> (http://go.microsoft.com/fwlink/?LinkId=182887) the CodePlex Web site. To read about support for SQL Server sample databases and sample code for SQL Server Express, see SQL Server Sample Databases <http://go.microsoft.com/fwlink/?LinkId=182888> (http://go.microsoft.com/fwlink/?LinkId=182888) on the CodePlex Web site.

Release Notes

For more information about late-breaking changes in this release of SQL Server, see the latest readme <http://go.microsoft.com/fwlink/?LinkId=182684>

Summary log file has been saved to the following location:

C:\Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log\20120524_083017\Summary_WIN-FSBH4A086BI_20120524_083017.txt

Figure 18: Installing SQL Server 2012 (XVI)

After the installation of SQL Server 2012 has completed, please run the SP1 update SQLServer2012SP1-KB2674319-x64-ENU.exe and follow the instructions on screen.

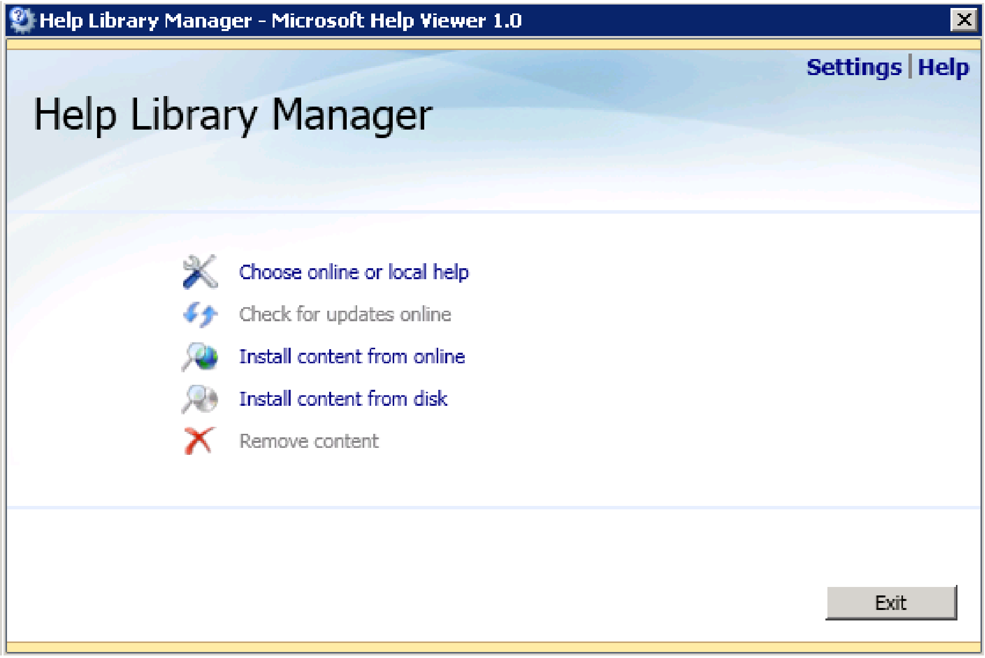
3.3.3 Installing the Documentation Contents

The description in this chapter focuses on a typical way of installing the SQL Server 2012 Documentation (Books Online). The Help Viewer 1.0 which is installed with SQL Server 2012 gives access to reference information, sample codes, technical articles, and guidance for Microsoft Products.

The Documentation contents do not come with SQL Server 2012 but have to be downloaded separately from the Common Criteria website [WEB]. After unzipping the content files to a folder they can be installed to be viewed with the Help Viewer.

For the installation of the Documentation contents the *Help Library Manager* has to be launched which can be found under Start > Microsoft SQL Server 2012 > Documentation & Community > Manage Help Settings.

Then the *Install content from the disk* option has to be started on the home screen of the Help Library Manager:

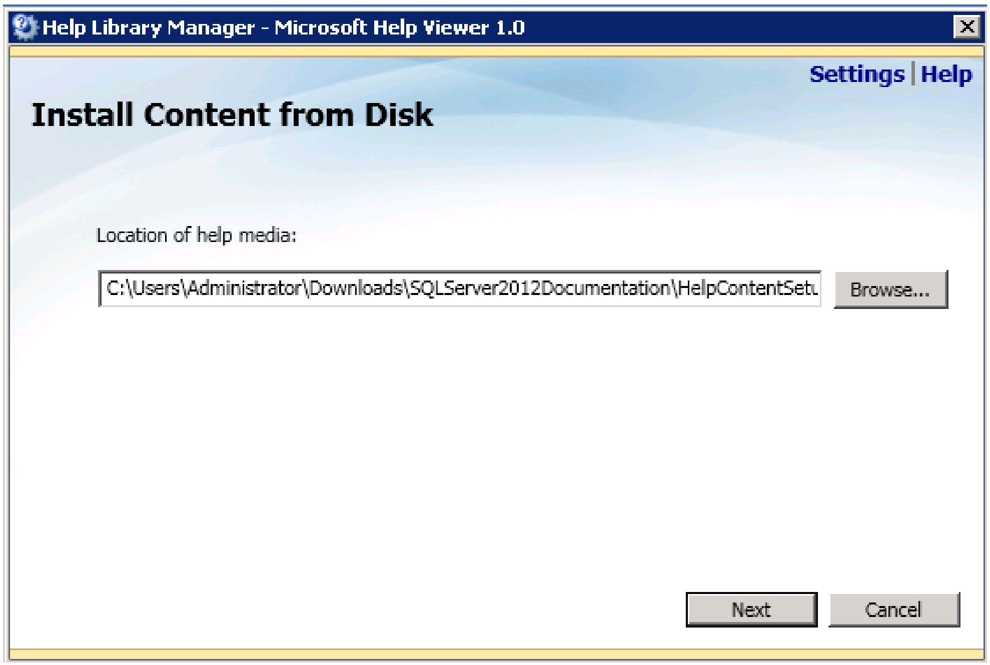


Help Library Manager - Microsoft Help Viewer 1.0
Settings | Help
Help Library Manager
Choose online or local help
Check for updates online
Install content from online
Install content from disk
Remove content
Exit

Figure 19: Installation of content from the disk

Note that the version of the Help Library Manager (1.0) differs from the version of the Help Viewer (1.1).

The next figure shows the settings of the path for the downloaded content. This screen appears after launching *the Install content from the disk* option.



Help Library Manager - Microsoft Help Viewer 1.0

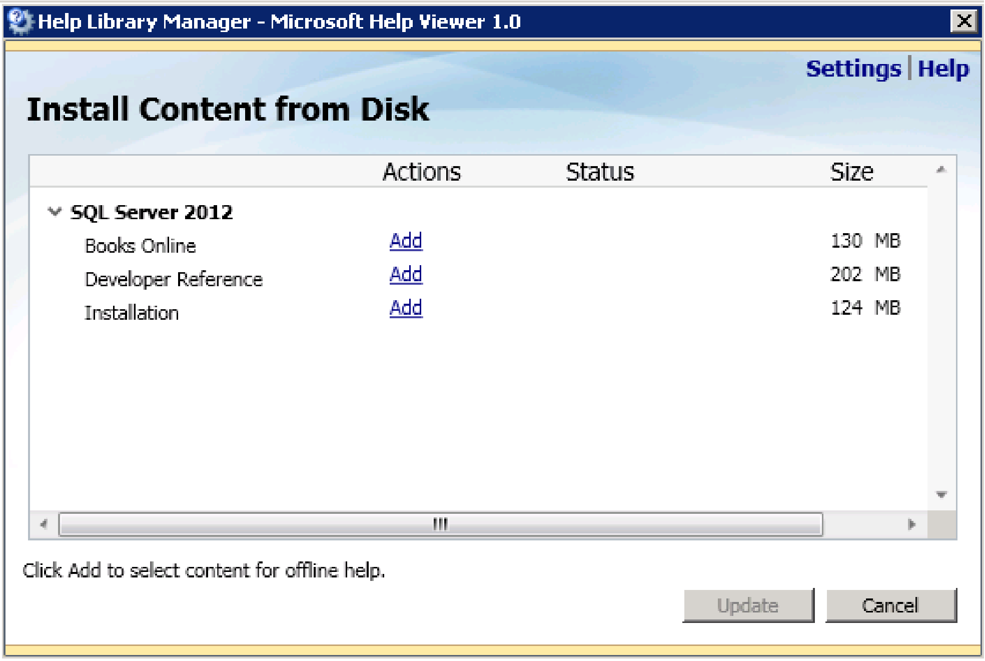
Install Content from Disk

Location of help media:

C:\Users\Administrator\Downloads\SQLServer2012Documentation\HelpContentSetu

Figure 20: Setting the path for the content

After setting the path to the “HelpContentSetup.msha” file located in the folder where the documentation contents have been unzipped, and clicking “Next”, the contents are listed as the figure below shows. The content to install can be selected by clicking the *Add* action next to the content title. When the selection of the content to install is finished, the *Update* button at the bottom of the screen has to be clicked and the update starts.



Help Library Manager - Microsoft Help Viewer 1.0

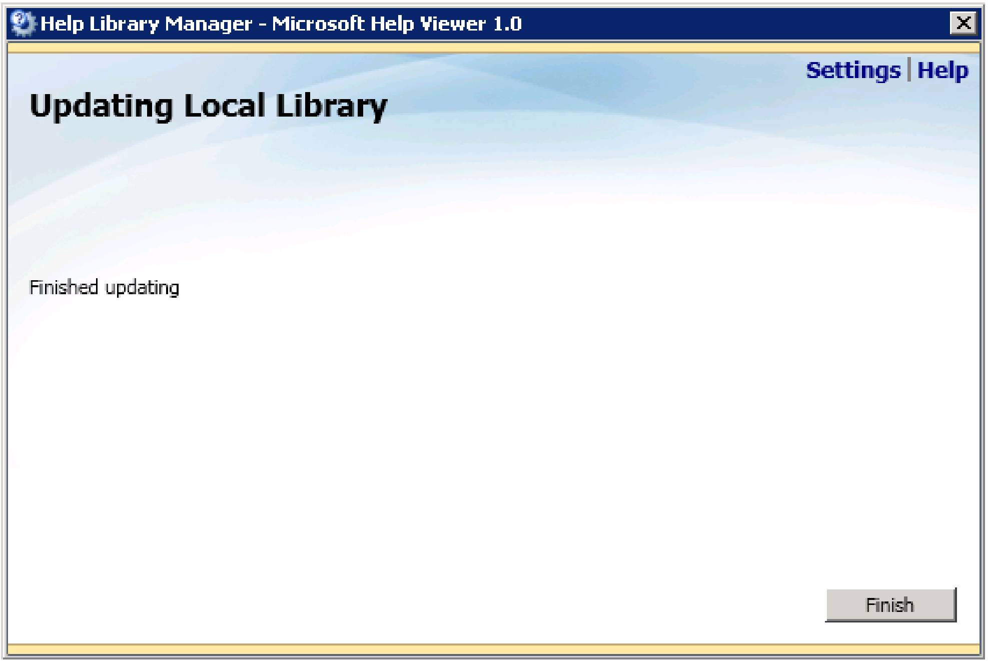
Install Content from Disk

	Actions	Status	Size
SQL Server 2012			
Books Online	Add		130 MB
Developer Reference	Add		202 MB
Installation	Add		124 MB

Click Add to select content for offline help.

Figure 21: Selection of the content to install

After a successful update of the local library the next figure appears on the screen and can be closed by clicking the *Finish* button.

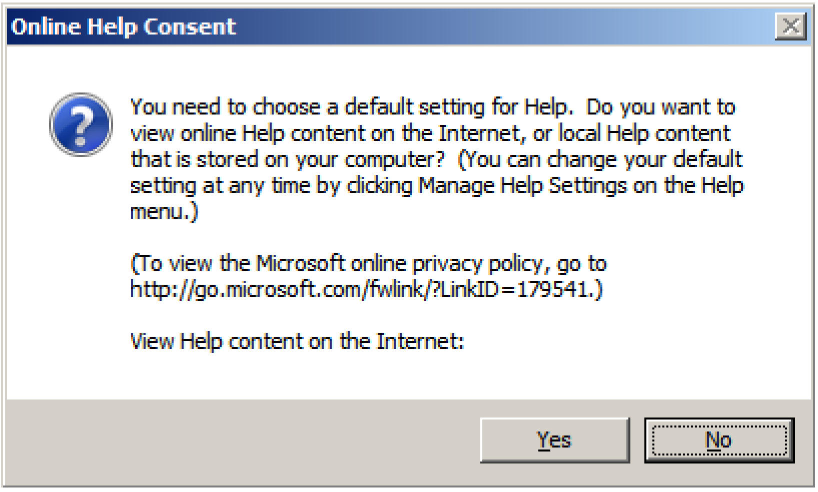


Updating Local Library

Finished updating

Figure 22: Updating the Local Library was successful

For the contents being listed on the Help Viewer 1.0 screen the application has to be started. It can be found under Start > Microsoft SQL Server 2012 > Documentation & Community > SQL Server Documentation. Upon first start the user is asked whether to view online Help. As only the local Help is part of the certified version of the TOE, choose “No” here:

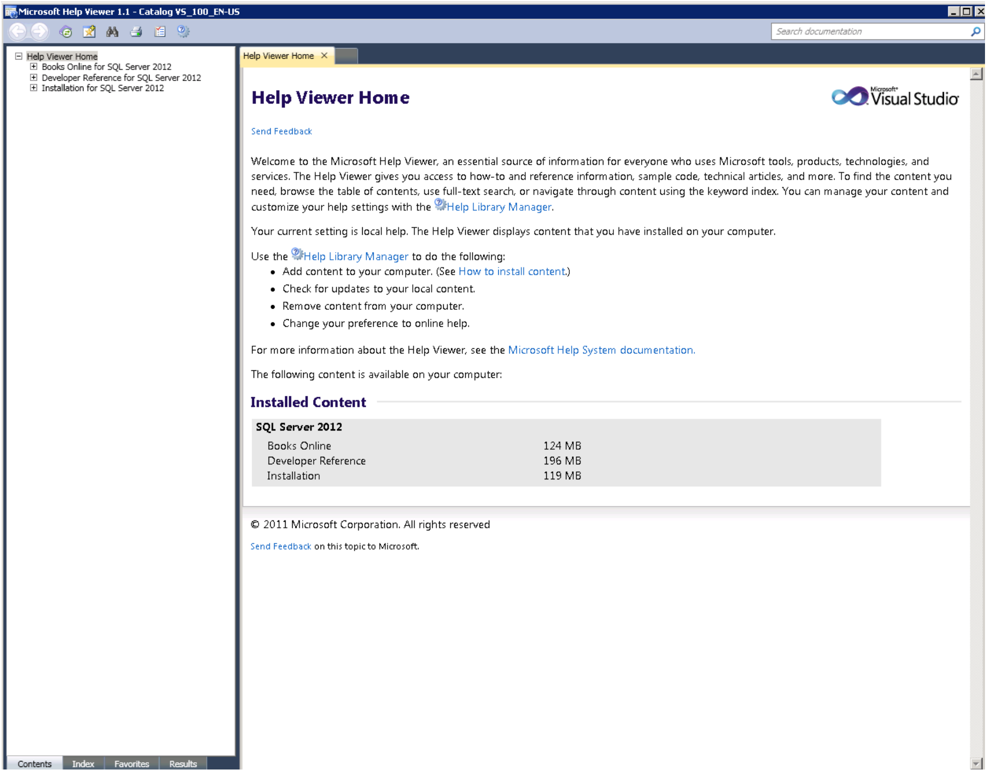


You need to choose a default setting for Help. Do you want to view online Help content on the Internet, or local Help content that is stored on your computer? (You can change your default setting at any time by clicking Manage Help Settings on the Help menu.)

(To view the Microsoft online privacy policy, go to http://go.microsoft.com/fwlink/?LinkID=179541.)

View Help content on the Internet:

The next figure shows the main screen of Help Viewer 1.0 after a successful update of the content files. The Documentations are visible in the left column.


Microsoft Help Viewer 1.1 - Catalog VS_100_EN-US
Search documentation
Help Viewer Home
Books Online for SQL Server 2012
Developer Reference for SQL Server 2012
Installation for SQL Server 2012
Contents
Index
Favorites
Results
Help Viewer Home
Microsoft Visual Studio

Help Viewer Home

Send Feedback

Welcome to the Microsoft Help Viewer, an essential source of information for everyone who uses Microsoft tools, products, technologies, and services. The Help Viewer gives you access to how-to and reference information, sample code, technical articles, and more. To find the content you need, browse the table of contents, use full-text search, or navigate through content using the keyword index. You can manage your content and customize your help settings with the Help Library Manager.

Your current setting is local help. The Help Viewer displays content that you have installed on your computer.

Use the Help Library Manager to do the following:

- Add content to your computer. (See How to install content.)
- Check for updates to your local content.
- Remove content from your computer.
- Change your preference to online help.

For more information about the Help Viewer, see the Microsoft Help System documentation.

The following content is available on your computer:

Installed Content

SQL Server 2012	
Books Online	124 MB
Developer Reference	196 MB
Installation	119 MB

© 2011 Microsoft Corporation. All rights reserved

Send Feedback on this topic to Microsoft.

Figure 23: After the restart the Documentations contents are updated

Note that setting local help as default can also be done by launching *Choose online or local help* in the Help Library Manager*:*

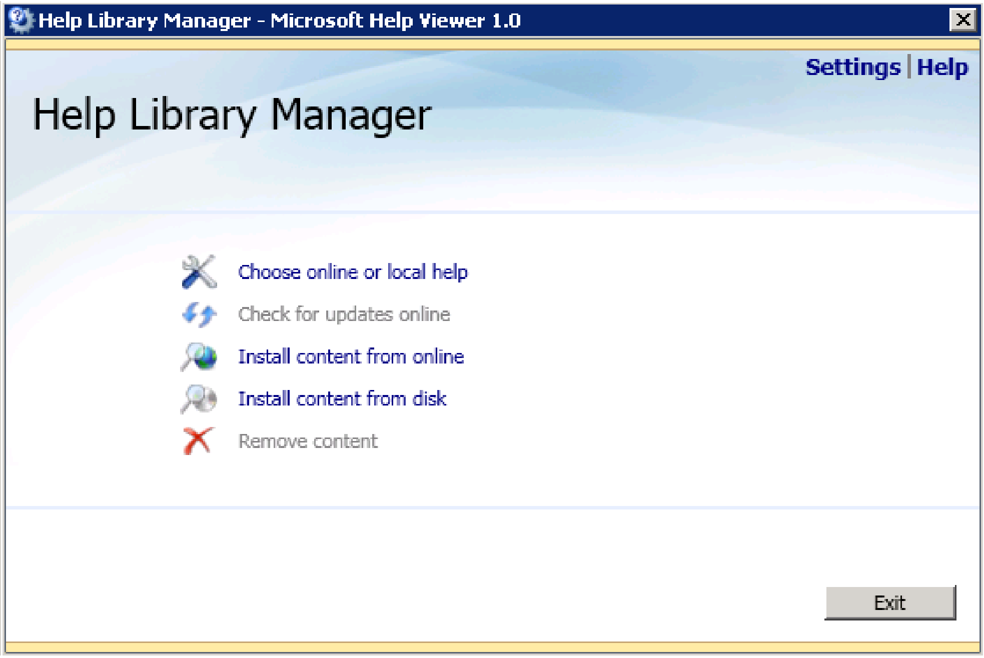
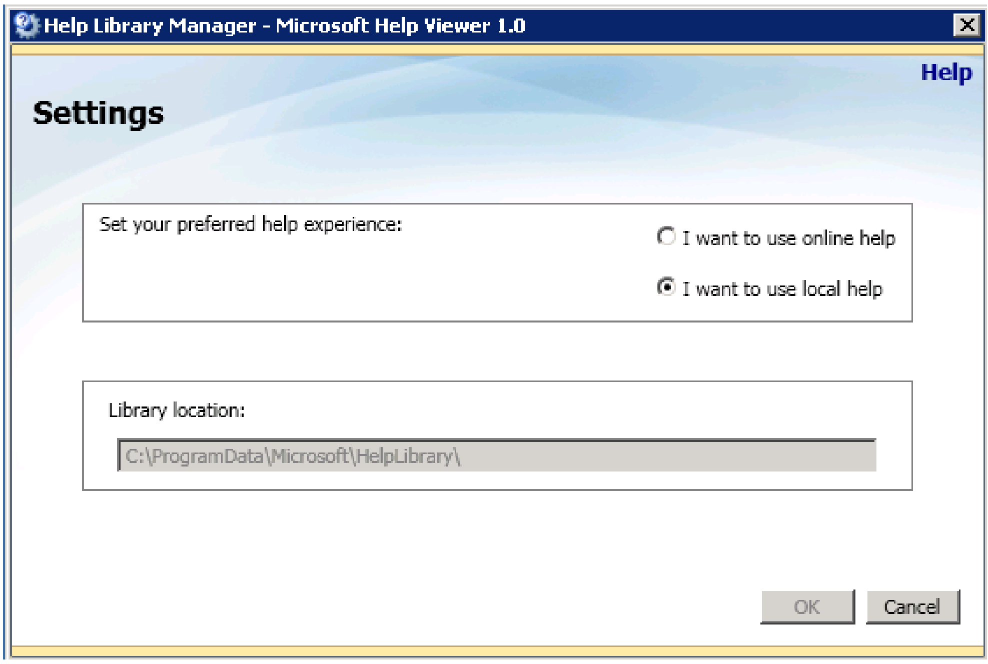

Help Library Manager - Microsoft Help Viewer 1.0
Settings | Help
Help Library Manager
Choose online or local help
Check for updates online
Install content from online
Install content from disk
Remove content
Exit

Figure 24: Choosing online or local help

The preferred help can be set like the following figure shows. The local help has to be chosen. The setting can be confirmed by clicking the *OK* button at the bottom of the screen.



Set your preferred help experience:

○ I want to use online help

● I want to use local help

Library location: C:\ProgramData\Microsoft\HelpLibrary\

Figure 25: Setting the preferred help experience

3.3.4 Checking the version of the product

After the installation process has been finished the admin shall finally determine whether the correct version of SQL Server 2012 is installed. To do this he has to connect to the running database engine (using any T-SQL client) and execute the following command:

```
SELECT @@VERSION
go
```

Using this command the TOE will return the name of the product platform (of which the TOE is the central part), the version number of the TOE and information about the Operating System.

For the certified version (which does not include the IA64 edition) the string that is returned in response to this command shall start with

- Microsoft SQL Server 2012 - 11.0.3000.0 (X64)

These strings include information on the concrete version of the SQL Server that has been installed (11.0.3000.0) and also show that the x64 edition has been installed.

3.3.5 Enabling the certified version

In the default installation of SQL Server 2012 some of the Security Features that are important in the context of the evaluated version are not enabled.

Thus the administrator has to enable the Common Criteria Compliance option. After the common criteria compliance enabled option is enabled, a table-level **DENY** takes **precedence** over a column-level GRANT. When the option is not enabled, a column-level GRANT takes precedence over a table-level DENY.

Residual information protection: This feature requires a memory allocation to be overwritten with a known pattern of bits before memory is reallocated to a new resource. Meeting the RIP standard can contribute to improved security; however, overwriting the memory allocation can slow performance. After the common criteria compliance enabled option is enabled, the overwriting occurs.

Login auditing will be enabled. Each time a user successfully logs in to SQL Server, information about the last successful login time, the last unsuccessful login time, and the number of attempts between the last successful and current login times is made available. These login statistics can be viewed by querying the sys.dm_exec_sessions dynamic management view.

To enable this option the administrator shall connect to the database engine and issue the following commands:

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'common criteria compliance enabled', 1;
GO
RECONFIGURE
GO
```

These setting takes effect directly after the server has been restarted.

For more information please refer to [AGD, ms-xhelp://?method=page&id=61766eea-c450-408d-af33-fbe7ef8c9ff2&product=SQLServer&productversion=110]

In addition to this setting, a CC-compliant audit log has to be configured. Please refer to chapter 6.2 for an example of a CC-compliant audit specification.

3.3.6 Installing the logon triggers

The Security Function for Session Handling allows an administrator to restrict the ability of users to connect to the TOE based on

- The number of concurrent sessions per login
- User identity and the day of the week and time of the day

This functionality is implemented using the logon triggers of the TOE. (For more information about logon triggers please refer to [AGD, ms-xhelp://?method=page&id=2f0ebb2f-de10-482d-9806-1a5de5b312b8&product=SQLServer&productversion=110])

This means that a trigger is executed every time a user is attempting to connect to the TOE. This trigger determines whether the user is allowed to establish a session at this time and denies session establishment if necessary.

The tables that store the information for this Security Function, the triggers and the Stored Procedures to manage this functionality have to be installed as they do not ship together with the database engine of SQL Server.

The installation can easily be done by executing the script "Install_cc_triggers.sql" that can be obtained via [WEB].

This script will install/create:

The tables:

- dbo.denied_logins_A54E382458CA11DB8373B622A1EF5492
  This table contains the weekly intervals in which logins are not allowed to connect to SQL Server. The table should not be modified directly. The following stored procedures should be used instead:
  - master.dbo.sp_deny_logon
  - master.dbo.sp_revoke_logon_denies
- dbo.maximum_number_of_connections_per_login_A54E382458CA11DB8373B622A1EF5492
  This table contains the value for the maximum number of connections per login. It should not be modified directly. Use the following stored procs instead:
  - master.dbo.sp_set_maximum_number_of_connections_per_login
  - master.dbo.sp_remove_maximum_number_of_connections_limit

The view:

- dbo.denied_logins
  This view dumps the contents of the table with the weekly intervals in human readable format.

The function

- dbo.fn_is_original_login_denied_A54E382458CA11DB8373B622A1EF5492
  This function checks whether the original login (the one who created the session) is allowed to logon at this time. EXECUTE permission for this function is granted to everyone.

The logon trigger

- trig_deny_access_A54E382458CA11DB8373B622A1EF5492
  This trigger is executed on every LOGON attempt. It checks whether the login is allowed to logon at this time (based on the time of the day and the day of the week) and if NOT rejects the connection by raising an exception.

- trig_max_connections_A54E382458CA11DB8373B622A1EF5492
  This trigger is executed on every LOGON attempt. It checks whether the login is allows to logon at this time (based on the maximum number of concurrent session per user) and if NOT rejects the connection by raising an exception.

The Stored Procedures

- dbo.sp_deny_logon_internal_A54E382458CA11DB8373B622A1EF5492
  This is an utility stored procedure and it is not supposed to be called directly
- dbo.sp_deny_logon (see chapter 5.4.1.8.1)
- dbo.sp_revoke_logon_denies (see chapter 5.4.1.8.2)
- dbo.sp_set_maximum_number_of_connections_per_login (see chapter 5.4.1.8.3)
- dbo.sp_remove_maximum_number_of_connections_limit (see chapter 5.4.1.8.4)

4 SQL SERVER BOOKS ONLINE

The TOE is the security relevant part of a database management system, which primary purpose is to store and retrieve user data in a secure way.

Thus it is impossible to define, who the user of the TOE will be in practice. Many scenarios for the use of a database management system are possible. E.g.

- A user, who uses a T-SQL client for interaction with the database engine of SQL Server 2012
- An application using the database engine of SQL Server 2012

Books Online ([AGD]) provides all kinds of users with the necessary information, how the database engine of SQL Server 2012 can be used.

The following links can be used as entry points into Books Online

Topic	Reference
What's New (Database Engine)	[AGD, ms-xhelp:///?method=page&id=8F625D5A-763C-4440-97B8-4B823A6E2439&product=SQLServer&productversion=110&locale=en-US&format=html&topicVersion=110]
SQL Server Database Engine Backward Compatibility	[AGD, ms-xhelp://?method=page&id=10de5ec6-d3cf-42ef-aa62-1bdf3fbde841&product=sqlserver&productversion=110]
Database Features and Tasks	[AGD, ms-xhelp://?method=page&id=d9efe145-3306-4d61-bd77-e2af43e19c34&product=sqlserver&productversion=110]
Technical Reference	[AGD, ms-xhelp:///?method=page&id=e9533f6b-c48a-4f53-a7a0-379e092bb667&product=sqlserver&productversion=110]
Transact-SQL Reference	[AGD, ms-xhelp://?method=page&id=dbba47d7-e08e-4435-b876-35dced1f325d&product=sqlserver&productversion=110]
XQuery Reference	[AGD, ms-xhelp://?method=page&id=8a69344f-2990-4357-8160-cb26aac95b91&product=sqlserver&productversion=110]

Table 3: Entry Points into Books Online

The following chapters are going to introduce the aspects for the secure administration and usage of SQL Server 2012, which are specific to the certified version.

5 GUIDANCE ADDENDUM

This chapter contains the guidance addendum for the secure administration and usage of the TOE. It only covers the aspects of guidance, which are specific to the certified version of the database engine of SQL Server 2012. It should be seen as a supplement to [AGD].

5.1 SQL Server startup flags

In its default configuration the process of the SQL Server database engine is running as a service under Windows 2012 Server and automatically started after the start of the Operating System.

However in some situations it can be useful to start the engine using the "sqlservr.exe" directly and using certain modes of operation.

The following table lists the available options to be used with the "sqlservr.exe" that result in a certain mode of operation:

Option	Description
-c	Shortens startup time when starting SQL Server from the command prompt. Typically, the SQL Server Database Engine starts as a service by calling the Service Control Manager. Because the SQL Server Database Engine does not start as a service when starting from the command prompt, use -c to skip this step.
-f	Starts an instance of SQL Server with minimal configuration. This is useful if the setting of a configuration value (for example, over-committing memory) has prevented the server from starting.
-g	Specifies an integer number of megabytes (MB) of memory that SQL Server will leave available for memory allocations within the SQL Server process, but outside the SQL Server memory pool. The memory outside of the memory pool is the area used by SQL Server for loading items such as extended procedure .dll files, the OLE DB providers referenced by distributed queries, and automation objects referenced in Transact-SQL statements. The default is 256 MB.
-m	Starts an instance of SQL Server in single-user mode. When you start an instance of SQL Server in single-user mode, only a single user can connect, and the CHECKPOINT process is not started. CHECKPOINT guarantees that completed transactions are regularly written from the disk cache to the database device. (Typically, this option is used if you experience problems with system databases that should be repaired.) Enables the sp_configure allow updates option. By default, allow updates is disabled. One can also limit the connection to a specified client application, however

Option	Description
	this feature shall not be used as a security feature because the provided client application name can be easily spoofed.
-n	Does not use the Windows application log to record SQL Server events. If you start an instance of SQL Server with -n, we recommend that you also use the -e startup option. Otherwise, SQL Server events are not logged.
-s	Allows you to start a named instance of SQL Server 2012. Without the -s parameter set, the default instance will try to start. You must switch to the appropriate BINN directory for the instance at a command prompt before starting sqlservr.exe. For example, if Instance1 were to use \mssql$Instance1 for its binaries, the user must be in the \mssql$Instance1\binn directory to start sqlservr.exe -s instance1.
-T trace#	Indicates that an instance of SQL Server should be started with a specified trace flag (trace#) in effect. Trace flags are used to start the server with nonstandard behavior. For more information, see [AGD, ms-xhelp://?method=page&id=b971b540-1ac2-435b-b191-24399eb88265&product=SQLServer&productversion=110]
-x	Disables monitoring features such as the keeping of CPU time and cache-hit ratio statistics. Allows maximum performance.
-E	Increases the number of extents that are allocated for each file in a filegroup. This option may be helpful for data warehouse applications that have a limited number of users running index or data scans.

Table 4: Startup Options for "sqlservr.exe"

The following modes shall not be used within the scope of the certified version as aspects of one or more Security Function as defined in [ST] may be affected.

- -f shall not be used within the scope of the certified version as aspects of one or more Security Function as defined in [ST] may be affected.
- -m: It cannot be guaranteed that all Security Functions are working in single user mode. Thus this mode must not be used within the certified version.

The following modes will require special care of the administrator. It is highly recommended not to use these modes within a productive environment within the scope of the certified configuration. However it can be necessary to use these modes for debugging or maintenance purposes or within a specific environment:

- -n: Though the application log is not a direct part of any Security Function (Audit uses audit files) it is highly recommended not to use this mode within the certified configuration.
- -E is specific for data warehouse applications. It is highly recommended not to use this mode within the certified configuration.

- -T Trace#: Indicates that an instance of SQL Server should be started with a specified trace flag (trace#) in effect. Trace flags are used to start the server with nonstandard behavior. For more information, see [AGD, ms-xhelp://?method=page&id=b971b540-1ac2-435b-b191-24399eb88265&product=SQLServer&productversion=110].

The following modes will not affect the behavior of the database engine with respect to the Security Functions and can therefore be used in the scope of the certified version:

- -c: will only shorten the startup process of the engine but not affect the behavior of any Security Function
- -g: is an option for tuning the way memory is handled. No Security Function is affected by this mode. However, to ensure the correct operation of the database engine, this parameter shall not be used with values less than 64 MB.
- -s: Simply starts a further instance of the engine. The instances will work independently and enforce all Security Functions.
- -x: This mode can be used as the tuning which is done in this mode to allow maximum performance does not impact the Security Functions as defined in [ST].

Please note that the "sqlservr.exe" provides more options than listed in the previous table. However the other options do not represent a different mode of operation but would e.g. allow the administrator to specify another path for database files or error logs.

A complete overview of the options for "sqlservr.exe" can be found in [AGD, ms-xhelp://?method=page&id=d373298b-f6cf-458a-849d-7083ecb54ef5&product=SQLServer&productversion=110].

5.2 Administration Interface

The complete management functionality of the TOE as defined in the ST is available via the use of T-SQL commands or Stored Procedures which can be called using a T-SQL command. In this way any T-SQL conformant client can be used for administration.

The SQL Server Management Studio which ships together with the TOE comprises a T-SQL client in a comfortable GUI can be used for administration.

However the functionality of the GUI has not been evaluated.

To manage the services associated with SQL Server, to configure the network protocols used by SQL Server, and to manage the network connectivity configuration from SQL Server client computers the SQL Server Configuration Manager tool can be used. The settings are stored and changed in the Operating System.

SQL Server Configuration Manager is a Microsoft Management Console snap-in that is available from the Start menu, or can be added to any other Microsoft Management Console display.

- SQL Server Configuration Manager can be used to start, pause, resume, or stop the services of SQL Server 2012, to view service properties, or to change service properties.

- SQL Server 2012 supports Shared Memory, TCP/IP, and Named Pipes protocols for its communication. These protocols can be managed (e.g. disabled and enabled) using SQL Configuration Manager. For information about choosing a network protocols see also [AGD, ms-xhelp://?method=page&id=e6beaea4-164c-4078-95ae-b9e28b0aefe8&product=SQLServer&productversion=110].

More detailed information about the functionality which is provided by the SQL Server Configuration Manager can be found in [AGD, ms-xhelp://?method=page&id=e6beaea4-164c-4078-95ae-b9e28b0aefe8&product=SQLServer&productversion=110].

5.3 User Interface

A user without administrative permissions can only connect to the TOE via the T-SQL interface using any T-SQL client via the protocols, which have been enabled by the administrator (see chapter 5.2).

The SQL Server Management Studio which ships together with the TOE comprises a T-SQL client, which can be used. (See also [AGD, ms-xhelp://?method=page&id=f289e978-14ca-46ef-9e61-e1fe5fd593be&product=SQLServer&productversion=110]). However the functionality of the GUI has not been evaluated.

For a complete overview over the T-SQL language please refer to the links under [AGD, ms-xhelp://?method=page&id=dbba47d7-e08e-4435-b876-35dced1f325d&product=SQLServer&productversion=110].

5.4 Security Functions relevant for administration and use of the TOE

The following chapters list the Security Functions of the TOE as defined in [ST] and describe which parts of these Security Functions are accessible for administrators and users.

5.4.1 Security Management

As the name implies, the Security Function “Security Management (SF.SM)” as defined in [ST] is the core function for the secure management of the TOE.

For users without any administrative permission this Security Function does not have any accessible part.

For administrators this Security Functions comprises the following aspects:

- Add and delete logins on an instance level
- Add and delete users on a database level
- Add and delete group memberships (for database groups and server groups)
- Create and destroy database groups
- Create, Start and Stop Security Audit

- Include and exclude Auditable events
- Define the mode of authentication for every login
- Modify the action to take in case the audit file is full
- Manage Attributes for Session Establishment

The Security Function SF.SM comprises all aspects, which are relevant for the administration of the Security Functions Identification & Authentication (SF.I&A), Security Audit (SF.AU) and Session Handling (SF.SE). Only the Security Function Access Control (SF.AC) contains an additional aspect for administration: The possibility to grant and deny permissions to users (see chapter 5.4.2).

The following chapters introduce the commands which can be used via any T-SQL client to perform the operations mentioned before. More details about the commands can be found in [AGD]. Note: For all operations the following chapters list stored procedures to start the operation as well as T-SQL commands. For these cases the T-SQL commands shall be used primarily as the Stored Procedures are legacy commands and will be removed in a future release.

5.4.1.1 Add and delete logins on an instance level

To add and delete logins on an instance level the following T-SQL commands can be used. These commands are also used to specify the type of the login (whether it is associated with a Windows user account or a SQL login) as one has to decide about the type of the login during creation time. Please note that SQL logins are only available if the Mixed Mode Authentication has been chosen during the installation process.

Command	Purpose	Reference in [AGD]
sp_addlogin	Add a login	[AGD, ms-xhelp:///?method=page&id=030F19C3-A5E3-4B53-BFC4-DE4BFCA0FDDC&product=SQLServer&productversion=110]
Create Login	Add a login	[AGD, ms-xhelp://?method=page&id=eb737149-7c92-4552-946b-91085d8b1b01&product=SQLServer&productversion=110]
sp_droplogin	Delete a login	[AGD, ms-xhelp://?method=page&id=e58684d1-c394-48de-906e-da6ee91100c3&product=SQLServer&productversion=110]
Drop Login	Delete a login	[AGD, ms-xhelp://?method=page&id=acb5c3dc-7aa2-49f6-9330-573227ba9b1a&product=SQLServer&productversion=110]

Table 5: Commands to add and delete logins

5.4.1.2 Add and delete users on a database level

To add or delete users from/to a database the following commands can be used:

Command	Purpose	Reference in [AGD]
sp_adduser	Add user	[AGD, ms-xhelp://?method=page&id=61a40eb4-573f-460c-9164-bd1bbfaf8b25&product=SQLServer&productversion=110]

Command	Purpose	Reference in [AGD]
Create user	Add user	[AGD, ms-xhelp://?method=page&id=01de7476-4b25-4d58-85b7-1118fe64aa80&product=SQLServer&productversion=110]
sp_dropuser	Delete user	[AGD, ms-xhelp://?method=page&id=e28f18f9-7ecf-4568-89f4-fe5c520df386&product=SQLServer&productversion=110]
Drop user	Delete user	[AGD, ms-xhelp://?method=page&id=d6e0e21a-7568-4321-b6d6-bcfba183a719&product=SQLServer&productversion=110]

Table 6: Commands to add and delete users

5.4.1.3 Add and delete group memberships

To add or delete users from/to a database role/group or a server scoped group the following commands can be used:

Command	Purpose	Reference in [AGD]
sp_addrolemember	Add a database user to a group	[AGD, ms-xhelp://?method=page&id=a583c087-bdb3-46d2-b9e5-3921b3e6d10b&product=SQLServer&productversion=110]
ALTER ROLE	Add a database user to a group	[AGD, ms-xhelp:///?method=page&id=E1E83CAA-17CC-4871-B2DB-2711339FB64F&product=SQLServer&productVersion=110]
sp_addsrvrolemember	Adds a login to a server scoped group	[AGD, ms-xhelp://?method=page&id=777f0e09-8ee5-4cb2-a3ac-939d02c3cd22&product=SQLServer&productversion=110]
ALTER SERVER ROLE	Adds a login to a server scoped group	[AGD, ms-xhelp:///?method=page&id=7A4DB7BB-C442-4E12-9A8A-114DA5BC7710&product=SQLServer&productVersion=110]
sp_droprolemember	Remove a database user from a group	[AGD, ms-xhelp://?method=page&id=c2f19ab1-e742-4d56-ba8e-8ffd40cf4925&product=SQLServer&productversion=110]
ALTER ROLE	Add a database user to a group	[AGD, ms-xhelp:///?method=page&id=E1E83CAA-

Command	Purpose	Reference in [AGD]
		17CC-4871-B2DB-2711339FB64F&product=SQLServer&productVersion=110]
sp_dropsrvrolemember	Remove a login from a server scoped role	[AGD, ms-xhelp://?method=page&id=7be99181-d221-49d0-9cb2-c930d8c044a0&product=SQLServer&productversion=110]
ALTER SERVER ROLE	Adds a login to a server scoped group	[AGD, ms-xhelp:///?method=page&id=7A4DB7BB-C442-4E12-9A8A-114DA5BC7710&product=SQLServer&productVersion=110]

Table 7: Commands to add and delete users from database and server groups

An overview over the predefined server roles that ship together with the product and their permissions can be found in [AGD, ms-xhelp:///?method=page&id=T%3AMICROSOFT.SQLSERVER.MANAGEMENT.COMMON.FIXEDSERVERROLES&product=SQLServer&productversion=110&locale=en-US&format=html&topicVersion=110].

5.4.1.4 Create and delete database groups

The following commands can be used to create and delete database scoped groups.

Command	Purpose	Reference in [AGD]
sp_addrole	Add a group	[AGD, ms-xhelp://?method=page&id=e8a21642-8440-419a-8585-93d3d9d44f00&product=SQLServer&productversion=110]
Create role	Add a group	[AGD, ms-xhelp://?method=page&id=b0cd54ad-e81d-4d71-acec-8a6d7261ca08&product=SQLServer&productversion=110]
sp_droprole	Delete a group	[AGD, ms-xhelp://?method=page&id=889ee074-00f8-40a9-bddb-d7d3ef0cbc19&product=SQLServer&productversion=110]
Drop Role	Delete a group	[AGD, ms-xhelp://?method=page&id=1f6f13ae-56a2-4ef1-93f5-8e6151b83e1d&product=SQLServer&productversion=110]

Table 8: Commands to create and destroy database groups

An overview over the predefined database roles that ship together with the product and their permissions can be found in [AGD, ms-xhelp:///?method=page&id=7F3FA5F6-6B50-43BB-9047-1544ADE55E39&product=SQLServer&productversion=110].

5.4.1.5 Create, Start and Stop Security Audit

The following commands can be used to create, enable and disable different kinds of audits. When creating a new audit one has to specify, what should happen in the case where the audit file is full.

Command	Purpose	Reference in [AGD]
CREATE SERVER AUDIT	Create a new server audit	[AGD, ms-xhelp://?method=page&id=1C321680-562E-41F1-8EB1-E7FA5AE45CC5&product=SQLServer&productversion=110]
CREATE SERVER AUDIT SPECIFICATION	Create a new server audit specification	[AGD, ms-xhelp://?method=page&id=DB77FA77-FEDB-40AC-83E6-06343063E518&product=SQLServer&productversion=110]
CREATE DATABASE AUDIT SPECIFICATION	Create a new database audit specification	[AGD, ms-xhelp://?method=page&id=0544DA48-0CA3-4A01-BA4C-940E23DC315B&product=SQLServer&productversion=110]
ALTER SERVER AUDIT name WITH (STATE = mode)	Enable and disable server audit	[AGD, ms-xhelp://?method=page&id=63426D31-7A5C-4378-AA9E-AFCF4F64CEB3&product=SQLServer&productversion=110]
ALTER SERVER AUDIT SPECIFICATION name WITH (STATE = mode)	Enable and disable server audit specification	[AGD, ms-xhelp://?method=page&id=9CAC288B-940E-4C16-88D6-DE06AEED2B47&product=SQLServer&productversion=110]
ALTER DATABASE AUDIT SPECIFICATION name WITH (STATE = mode)	Enable and disable database audit specification	[AGD, ms-xhelp://?method=page&id=85F4E7E6-A330-4DE0-9048-64F386CCC314&product=SQLServer&productversion=110]

Table 9: Commands to create, start and stop audit

5.4.1.6 Include and exclude Auditable events

The following commands can be used to include and exclude auditable events from/to an audit file and to apply a filter to an audit.

Command	Purpose	Reference in [AGD]
ALTER SERVER AUDIT SPECIFICATION audit_name ADD / DROP audit_action_group_name	Include and exclude auditable events	[AGD, ms-xhelp://?method=page&id=9CAC288B-940E-4C16-88D6-DE06AEED2B47&product=SQLServer&productversion=110]
ALTER DATABASE AUDIT SPECIFICATION audit_name ADD / DROP audit_action_specification	Apply a filter to an audit	[AGD, ms-xhelp://?method=page&id=85F4E7E6-A330-4DE0-9048-64F386CCC314&product=SQLServer&productversion=110]

Table 10: Commands to include and exclude auditable event

5.4.1.7 Define the mode of authentication for every login

The mode of authentication for every login of the TOE has to be determined at creation time. The administrator, who creates a new login has to specify, whether a Windows login should be created or a SQL login. This is done via the parameter WINDOWS of the CREATE LOGIN command (See Chapter 5.4.1.1).

5.4.1.8 Manage Attributes for Session Establishment

The following stored procedures can be used to manage the attributes for session establishment. After a default installation of the engine as described in chapter 3 of this document the maximum number of sessions per user is set to 5 and initially no further default deny rules are existing.

5.4.1.8.1 Sp_deny_logon

This Stored Procedure allows the administrator to deny session establishment to a certain login based on the day of the week and the time of the day.

Syntax

```
sp_deny_logon [@login_name=] 'login'
        ,[@start_weekday=] start_weekday
        , [@start_time =] 'start_time'
        ,[@end_weekday=] end_weekday
        ,[@end_time=] 'end_time'
```

Arguments

[@login_name=] 'login'

Is the name of the login. 'login is of data type **sysname**.

[@start_weekday=] start_weekday

Is the day of the week where the session deny should start. Start_weekday is **tinyint** according to the @@DATEFIRST setting (i.e. 1 means Sunday in the default setting for @@DATEFIRST).

[@start_time =] 'start_time'

Is the time of the day where the session deny should start. Start_time is of **nvarchar(12),** in format hh:mm:ss.000 (the last three digits represent milliseconds)

[@end_weekday=] end_weekday

Is the day of the week where the session deny should end. end_weekday is **tinyint** according to the @@DATEFIRST setting (i.e. 1 means Sunday in the default setting for @@DATEFIRST).

[@end_time=] 'end_time'

Is the time of the day where the session deny should end. end_time is of **nvarchar(12),** in format hh:mm:ss.000 (the last three digits represent milliseconds)

Return Values

0 (Success) or >0 (Failure)

Remarks

This Stored Procedure can be called with any @@datefirst setting and the start of the interval given can be > than the end of the interval. In this case it splits the passed interval into two intervals.

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

Permissions

Requires the CONTROL SERVER permission.

5.4.1.8.2 Sp_revoke_logon_denies

This Stored Procedure allows an administrator to revoke all denies from a certain login.

Syntax

sp_revoke_logon_denies [@login_name=]'login'

Arguments

[@login_name=] 'login'

Is the name of the login for which all denies shall be revoked. 'login is of data type **sysname**.

Return Values

0 (Success) or >0 (Failure)

Remarks

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

Permissions

Requires the CONTROL SERVER permission.

5.4.1.8.3 Sp_set_maximum_number_of_connections_per_login

This Stored Procedure allows the administrator to set the maximum number of connections that are allowed per login. This value is a global value that is valid for all logins.

Syntax

dbo.sp_set_maximum_number_of_connections_per_login
[@max_connections=] max_connections

Arguments

[@max_connections=] max_connections

New value for the maximum number of allowed connection per login. Max_connections is of data type INT.

Return Values

0 (Success) or >0 (Failure)

Remarks

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

Permissions

Requires the CONTROL SERVER permission.

5.4.1.8.4 Sp_remove_maximum_number_of_connections_limit

This Stored Procedure allows the administrator to remove the setting for the maximum number of connections that are allowed per login. After successfully executing this Stored Procedure the TOE will no longer enforce any limitation on the number of concurrent sessions per login.

Syntax

dbo.sp_remove_maximum_number_of_connections_limit

Arguments

-

Return Values

0 (Success) or >0 (Failure)

Remarks

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

Permissions

Requires the CONTROL SERVER permission.

5.4.2 Access Control

The Security Function Access Control ensures that only users, which have appropriate permissions, are able to perform operations on objects, under the control of the Security Function. The complete description of the Security Function can be found in [ST].

For users without any administrative permission the Security Function Access Control is only accessible in so far that for every command, which is issued to the TOE, the Security Function will check, whether the user has the appropriate permissions.

A part of the Security Function is that it is possible for administrators to grant, revoke or deny permissions to users using the following commands:

Command	Purpose	Reference
Grant	Grant a permission	[AGD, ms-xhelp://?method=page&id=a760c16a-4d2d-43f2-be81-ae9315f38185&product=SQLServer&productversion=110]
Deny	Deny a permission	[AGD, ms-xhelp://?method=page&id=c32d1e01-9ee9-4665-a516-fcfece58078e&product=SQLServer&productversion=110]
Revoke	Revoke a granted or denied permission	[AGD, ms-xhelp://?method=page&id=9D31D3E7-0883-45CD-BF0E-F0361BBB0956&product=SQLServer&productversion=110]

Table 11: Commands to grant, revoke and deny permissions

Based on the identity of the user, the group membership of the user and the granted or denied permissions the database engine will decide based on the following rules whether an operation that is requested by a user is allowed:

1. If the requested mode of access is denied to the user, the access will be denied
2. If the requested mode of access is denied to any role of which the user is a member, the access will be denied
3. If the requested mode of access is permitted to that user, the access will be permitted
4. If the requested mode of access is permitted to any role of which the user is a member, the operation will be permitted
5. Else: The access will be denied

It should be noted that the permission check on an object includes the permissions of its parent objects. The permissions for the object itself and all its parent objects are accumulated together before the aforementioned rules are evaluated.

However, there are two cases for which the aforementioned rules are overridden:

1. A sysadmin, the owner of an object and owners of parent objects always have access

2. In the case of "Ownership Chaining" (see also [AGD, ms-xhelp:///?method=page&id=7B2D49F2-B91C-4AEE-A52B-6CC49BED03AF&product=SQLServer&productversion=110]) the access is allowed.

5.4.3 Identification & authentication

The Security Function Identification & Authentication ensures that each user has been successfully authenticated before any other operations on behalf of that user are allowed. The complete description of the Security Function can be found in [ST].

The Security Function Identification & Authentication only has a small user-accessible part, which is that every user will be authenticated when connecting to the TOE. This applies to administrators and to users without any administrative permissions.

The complete administrative part of this Security Function is covered by the Security Function Security Management (see chapter 5.4.1).

5.4.4 Security Audit

This Security Function ensures that the TOE produces audit logs for a set of security relevant actions. These audit logs are stored into audit files in the environment of the TOE. The complete description of the Security Function including the complete list of events can be found in [ST].

For the user of the database engine without any administrative permission the Security Function Security Audit does not have any user-accessible functionality.

The complete administrative part of this Security Function is covered by the Security Function Security Management (see chapter 5.4.1).

For further information about the audit functionality of the TOE please refer to chapter 6.

5.4.5 Session Handling

The information about the

- last successful attempt to establish a session
- last unsuccessful attempt to establish a session
- number of unsuccessful login attempts since the last successful login

can be obtained via the dynamic management view sys.dm_exec_sessions.

SELECT permission on this management view is granted to public by default so that every user is able to retrieve the information from this view. The user will retrieve information about their current session plus the date and time of the last unsuccessful and successful login attempt (before the current session was established) and the number of unsuccessful login attempts since the last successful login. A user who has the VIEW SERVER STATE permission will see this information for all active sessions.

6 SQL Server Audit

SQL Server supports auditing an instance of the Database Engine (server audit) or an individual database (database audit). SQL Server Audit is configured through the objects SERVER AUDIT, SERVER AUDIT SPECIFICATION and DATABASE AUDIT SPECIFICATION which can be created, modified, and dropped through the Data Definition Language (DDL) statements CREATE, ALTER, and DROP respectively.

This chapter presents an excerpt from the product documentation (see [AGD, ms-xhelp://?method=page&id=0C1FCA2E-F22B-4FE8-806F-C87806664F00&product=SQLServer&productversion=110] for detailed information) and gives a summary of these objects and their configuration.

6.1 SERVER AUDIT

The SERVER AUDIT object defines the audit target which can be a file, the Windows Application log, or the Windows Security log. If the audit target is a file, the path of the audit log, the maximum size of one audit file and the maximum number of audit files can be determined.

Furthermore, the TOE behaviour in the following situations can be defined:

- If the maximum number of audit files is reached the TOE can either stop logging or roll over old files.
- If an audit write failure occurs, the TOE can be configured to continue operation, to shut down or to fail the database action that caused the event to be audited. However, the latter configuration (ON_FAILURE=FAIL_OPERATION option) shall not be used for the certified TOE.

The SERVER AUDIT object can also be configured with a predicate expression to filter audit events to be written to the audit target. To filter audit events to be audited one can also use the WHERE clause (see [AGD, ms-xhelp://?method=page&id=1C321680-562E-41F1-8EB1-E7FA5AE45CC5&product=SQLServer&productversion=110]).

6.1.1 Examples of Use

The following example creates a CC-compliant SERVER AUDIT object[3]:

```
CREATE SERVER AUDIT CCAudit
    TO FILE ( FILEPATH ='C:\CCAudit\', MAX_ROLLOVER_FILES=10 );
```

This example overwrites the oldest stored audit record if the audit trail is full. To stop the TOE operation instead, the example can be modified as follows:

```
CREATE SERVER AUDIT CCAudit
    TO FILE ( FILEPATH ='C:\CCAudit\' ) WITH (ON_FAILURE=SHUTDOWN);
```

[3] The directory to which the file path points to has to be created before.

The following example shows how to select the set of events to be audited based on the user identity, event type, object identity and outcome using the WHERE clause:

```
CREATE SERVER AUDIT CCAudit
    TO FILE ( FILEPATH ='C:\CCAudit\', MAX_ROLLOVER_FILES=10 )
    WHERE server_principal_name= 'sa';
```

Similarly, one can use the “action_id”, “object_id” and the “succeeded” field names in the WHERE clause to filter for event type, object identity and outcome. Please note that the action_id cannot be specified as a varchar(4) as retrieved via sys.fn_get_audit_file but needs to be specified as an integer. To translated the action_id to an integer the following function can be used:

```
create function action_id ( @action_id varchar(4)) returns int
begin
declare @x int
SET @x = convert(int, convert(varbinary(1),
upper(substring(@action_id, 1, 1))))
if LEN(@action_id)>=2
SET @x = convert(int, convert(varbinary(1),
upper(substring(@action_id, 2, 1)))) * power(2,8) + @x
else
SET @x = convert(int, convert(varbinary(1), ' ')) * power(2,8) + @x
if LEN(@action_id)>=3
SET @x = convert(int, convert(varbinary(1),
upper(substring(@action_id, 3, 1)))) * power(2,16) + @x
else
SET @x = convert(int, convert(varbinary(1), ' ')) * power(2,16) + @x
if LEN(@action_id)>=4
SET @x = convert(int, convert(varbinary(1),
upper(substring(@action_id, 4, 1)))) * power(2,24) + @x
else
SET @x = convert(int, convert(varbinary(1), ' ')) * power(2,24) + @x
return @x
end
```

Note that when a SERVER AUDIT object is created, it is in a disabled state and needs to be enabled using ALTER SERVER AUDIT, e.g.:

```
ALTER SERVER AUDIT CCAudit WITH STATE = ON;
```

To remove the SERVER AUDIT object, execute:

```
ALTER SERVER AUDIT CCAudit WITH (STATE = OFF);
DROP SERVER AUDIT CCAudit;
```

6.1.2 Audit Record Contents

To read in an audit file from disk the function sys.fn_get_audit_file or an external viewer (such as the Log File Viewer in SQL Server Management Studio) can be used.

For each logged event the following information is recorded as required by the Security Target (please see [AGD, ms-xhelp://?method=page&id=7A291015-DF15-44FE-8D53-C6D90A157118&product=SQLServer&productversion=110] for a complete list of information that is audited for an event):

Information to be recorded	Column name	Type
Date and time of the event	event_time	datetime2
Type of event	action_id	char(4)
Subject identity (if applicable)	server_principal_name	sysname

Table 12: Audit Record

To get an overview of all auditable actions, execute the following SQL query:

```
SELECT action_id, name, class_desc, containing_group_name FROM
sys.dm_audit_actions
WHERE action_in_log=1
ORDER BY action_id;
```

To get a mapping of a class_type (which appears in the log file instead of the full class_desc) execute the following SQL query:

```
SELECT class_type, class_type_desc FROM sys.dm_audit_class_type_map
ORDER bY class_type;
```

6.2 SERVER AUDIT SPECIFICATION

A SERVER AUDIT SPECIFICATION object adds audit action groups[4] to a SERVER AUDIT object, thus defining which server-level audit events shall be logged in the target defined in the SERVER AUDIT object. A complete list of server-level audit action groups can be found in [AGD].

Event	Action_id (name)	Containing Action Group Name
Start-up and shutdown of the audit functions	AUSC (AUDIT SESSION CHANGED)	NULL
Start-up and shutdown of the DBMS	SVSR (SERVER STARTED) SVSD (SERVER SHUTDOWN)	SERVER_STATE_CHANGE_GROUP

[4] An audit action group is a set of actions which are functionally related.

Event	Action_id (name)	Containing Action Group Name
All modifications to the audit configuration that occur while the audit collection functions are operating.	AL (ALTER)	AUDIT_CHANGE_GROUP
Successful requests to perform an operation on an object covered by the SFP.	CR (CREATE) DR (DROP) AL (ALTER)[5]	DATABASE_OBJECT_ACCESS_GROUP DATABASE_CHANGE_GROUP DATABASE_OBJECT_CHANGE_GROUP SCHEMA_OBJECT_ACCESS_GROUP SCHEMA_OBJECT_CHANGE_GROUP SERVER_OBJECT_CHANGE_GROUP
Unsuccessful revocation of security attributes[6]	R (REVOKE)	DATABASE_OBJECT_PERMISSION_CHANGE_GROUP DATABASE_PERMISSION_CHANGE_GROUP SCHEMA_OBJECT_PERMISSION_CHANGE_GROUP SERVER_PERMISSION_CHANGE_GROUP
Use of the management functions:		
• Add and delete logins	CR (CREATE) DR (DROP)	SERVER_PRINCIPAL_CHANGE_GROUP
• Add and delete users	CR (CREATE) DR (DROP)	DATABASE_PRINCIPAL_CHANGE_GROUP
• Change role membership for DB scoped roles	APRL (ADD MEMBER) DPRL (DROP MEMBER)	DATABASE_ROLE_MEMBER_CHANGE_GROUP
• Change role	APRL (ADD	SERVER_ROLE_MEMBER_CHANGE_GROUP

[5] This is only a subset of all possible action ids for operations on objects covered by the SFP. To get an overview of all auditable actions, please refer to previous section.

[6] This applies to revocation of database role memberships and permissions.

Event	Action_id (name)	Containing Action Group Name
membership for Server scoped roles	MEMBER) DPRL (DROP MEMBER)	
• Create and destroy database scoped groups	CR (CREATE) DR (DROP)	DATABASE_PRINCIPAL_CHANGE_GROUP
• Create, Start and Stop Audit	CR (CREATE) AUSC (AUDIT SESSION CHANGED)	AUDIT_CHANGE_GROUP NULL
• Include and Exclude Auditable events	CR (CREATE) DR (DROP) AL (ALTER)	AUDIT_CHANGE_GROUP
• Define the mode of authentication	CR (CREATE) AL (ALTER)	SERVER_ PRINCIPAL_CHANGE_GROUP SERVER_OBJECT_CHANGE_GROUP DATABASE_PRINCIPAL_CHANGE_GROUP
• Define the action to take in case the audit file is full	CR (CREATE) AL (ALTER)	AUDIT_CHANGE_GROUP
Modifications to the group of users that are part of a role	APRL (ADD MEMBER) DPRL (DROP MEMBER)	SERVER_ROLE_MEMBER_CHANGE_GROUP DATABASE_ROLE_MEMBER_CHANGE_GROUP
Every use of the authentication mechanism	LGIF (LOGIN FAILED) LGIS (LOGIN SUCCEEDED) DBAF (DATABASE AUTHENTICATION FAILED)	FAILED_LOGIN_GROUP SUCCESSFUL_LOGIN_GROUP FAILED_DATABASE_AUTHENTICATION_GROUP SUCCESSFUL_DATABASE_AUTHENTIATION_GROUP
The final decision on authentication		

Event	Action_id (name)	Containing Action Group Name
	DBAS (DATABASE AUTHENTICATIO N SUCCEEDED)	
Every modifications to the setting preventing the loss of audit data (i.e. roll-over or stop TOE operation).	CR (CREATE) AL (ALTER)	AUDIT_CHANGE_GROUP

Table 13: Audit Events

6.2.1 Examples of Use

The following example creates a CC-compliant SERVER AUDIT SPECIFICATION based on the server-level events listed in Table 13. This configuration is mandatory for a Security Target conform operation of the TOE.

```
CREATE SERVER AUDIT SPECIFICATION CCAuditServerSpec
       FOR SERVER AUDIT CCAudit
       -- Audit changes
       ADD (AUDIT_CHANGE_GROUP),
       -- Authentication Failure/Success
       ADD (FAILED_LOGIN_GROUP),
       ADD (SUCCESSFUL_LOGIN_GROUP),
       ADD (FAILED_DATABASE_AUTHENTICATION_GROUP),
       ADD (SUCCESSFUL_DATABASE_AUTHENTICATION_GROUP),
       -- Schema-level
       ADD (SCHEMA_OBJECT_ACCESS_GROUP),
       ADD (SCHEMA_OBJECT_CHANGE_GROUP),
       ADD (SCHEMA_OBJECT_PERMISSION_CHANGE_GROUP),
       -- Database-level
       ADD (DATABASE_CHANGE_GROUP),
       ADD (DATABASE_OBJECT_ACCESS_GROUP),
       ADD (DATABASE_OBJECT_CHANGE_GROUP),
       ADD (DATABASE_OBJECT_PERMISSION_CHANGE_GROUP),
       ADD (DATABASE_PERMISSION_CHANGE_GROUP),
       ADD (DATABASE_PRINCIPAL_CHANGE_GROUP),
       ADD (DATABASE_ROLE_MEMBER_CHANGE_GROUP),
       -- Server-level
       ADD (SERVER_OBJECT_CHANGE_GROUP),
       ADD (SERVER_PERMISSION_CHANGE_GROUP),
       ADD (SERVER_PRINCIPAL_CHANGE_GROUP),
       ADD (SERVER_ROLE_MEMBER_CHANGE_GROUP),
       ADD (SERVER_STATE_CHANGE_GROUP)
       WITH (STATE=ON);
```

6.3 DATABASE AUDIT SPECIFICATION

A DATABASE AUDIT SPECIFICATION object adds audit action groups or single actions performed (on a securable by a principal) to a SERVER AUDIT object, thus defining which database-level audit events shall be logged in the target defined in the SERVER AUDIT object. A complete list of database-level audit action groups and database-level auditable actions can be found in [AGD, ms-xhelp://?method=page&id=B7422911-7524-4BCD-9AB9-E460D5897B3D&product=SQLServer&productversion=110].

6.4 Security Relevant Events

The audit capabilities of the TOE are a powerful mechanism to detect potential security breaches. However the secure operation of the TOE needs the attention of the administrator. He shall review the audit files regularly and pay attention to any suspicious events or events that require a certain action.

As the definition of “suspicious” depends on the concrete installation and environment of the TOE it is not possible to provide a comprehensive definition of what suspicious events are. For example 1000 unsuccessful authentication attempts or failed read attempts per hour may not be suspicious in an installation that serves millions of users while it would be highly suspicious in installations with only a few users.

Classical suspicious events could e.g. be

- An unusual high amount of unsuccessful authentication attempts, which could point to a brute force attack.
- An unusual high amount of events recorded in the audit files could be an indication for an attacker, who is trying to flood the audit files in order to conceal an unauthorized operation.

Other events that can require an action by the administrator include:

- An audit file that is running out of disc space. Depending on the concrete configuration the database engine may shut down or overwrite old audit files if a certain size is reached. In those cases the administrators shall consider to back-up the audit files and start over the audit process with a new set of files.
- There are multiple event types in a company that may require the administrator to change settings of the database engine. A classic example is a user owning a login in the database engine leaving the company. In such a case the administrator would usually consider to delete or block the login of the user.

7 Recommendations and requirements for secure administration, configuration and usage

The administrator of the TOE shall follow the following recommendations and requirements to ensure a secure operation of the TOE:

7.1 Recommendations/requirements about Security Audit

- It is recommended to use a separate CC audit for all the events which have to be captured according to [ST]. See also chapter 6 for further guidance to create this audit.
- The CC audit should always be running. If it is necessary to stop the audit while the TOE is still running (e.g. to change the configuration of the audit) it should be considered to create an audit which contains all the relevant events as listed in chapter 6 and to start this new audit before the CC audit is stopped. In this way it can be ensured that the admin misses no important event. It is also recommended to check that the audit is running before connecting the TOE to an untrusted network.
- For the “CC audit” it has to be ensured that the option “ON_FAILURE=SHUTDOWN” is used, i.e. that the TOE will stop operation in case an error occurs. Please note that the TOE will stop immediately and will not write any further audit events to disk. This option can be combined with the option “MAX_ROLLOVER_FILES”.
- For the case that the MAX_ROLLOVER_FILES option is used it is possible that an attacker floods the audit and intentionally causes an event to be overwritten. Thus the administrator has to ensure that sufficient disc space is available for the audit files and appropriate settings are used for the audit processes. Specifically – in cases where the audit of certain event is more important than the availability of the server – it should be considered not to use the MAX_ROLLOVER_FILES option (i.e. to ensure that the server will shut down if the audit file is full) for all or certain audit processes.
- Audit can be configured to write logs synchronously (QUEUE_DELAY = 0, i.e. buffer is not used) or asynchronously (QUEUE_DELAY = n, i.e. buffer of length n is used). In case of an audit failure (e.g. the TOE stops or the disk is full), the TOE does not write events contained in the buffer.
- Please note that some Audit Action Groups do not log the outcome of the contained actions. Therefore it is required to create a separate event session that logs all reported errors of SQL statements by executing the following statement (adapt filename accordingly):

```
CREATE EVENT SESSION [CCOutcome] ON SERVER
ADD EVENT sqlserver.error_reported(
    ACTION(sqlserver.server_principal_name,sqlserver.sql_text)
    WHERE ([sqlserver].[sql_text]<>N'NULL'))
```

```
ADD TARGET package0.event_file(SET filename=N'C:\CCOutcome.xel',
max_file_size=(10),max_rollover_files=(2))
WITH (STARTUP_STATE=ON)
GO
```

7.2 Recommendations and further information about Access Control

- It should be mentioned that some permissions of the database engine of SQL Server do imply other permissions. A good example of such a permission is the CONTROL SERVER permission that covers all other permissions. The complete hierarchy of permissions within the SQL Server database engine is contained in the file permission_hierarchy.zip ([PERM]) that can be downloaded from [WEB]. This file contains 4 charts that show the permission hierarchy on the 4 levels: server, database, object and column. In each chart it is shown, which permissions on an object imply which other permissions. For example the extract shown in Figure 26 (from Column.pdf) shows that permissions on a table do automatically imply permissions on the columns of the table. More specific: The CONTROL permission on a table implies permissions for Reference, Select and Update on the columns of the table.

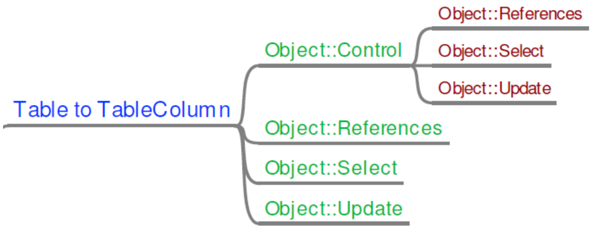


Figure 26: Extract of permission hierarchy

- According to the concept for Access Control in SQL Server 2012 it is possible (if not likely) that two users/administrators have the same permission for one object. This could lead into a situation, where administrators/users cause conflicting operation (e.g. that one administrator grants access to an object while a second administrator denies the same access). These situations can only be avoided by organizational mechanisms and the administrator should be well aware of this fact.
- In its default configuration the database engine of SQL Server 2012 grants the EXECUTE permission on many Stored Procedures to public. This has been done to ensure a maximum level of compatibility to applications. However, some of the Stored Procedures do provide access to sensitive information or open channels for potential attacks. Therefore the administrator shall consider to revoke the EXECUTE permission on all Stored Procedures from public and grant those EXECUTE permissions to specific users or their corresponding groups if necessary.

- The internal access control functionality of the Stored Procedures 'sp_replsendtoqueue' and 'sp_replwritetovarbin' is not compliant to the certification. Therefore these two procedures must not be accessible by any user within the scope of the certified version of the database engine. After a default installation however the execute permission on these Stored Procedures is granted to public. Therefore the administrator shall revoke the execute permissions from theses Stored Procedures from public.
- The description of the sp_dropsrvrolemember in [AGD, ms-xhelp://?method=page&id=7BE99181-D221-49D0-9CB2-C930D8C044A0&product=SQLServer&productversion=110] describes that the membership in the sysadmin fixed server role, or both ALTER ANY LOGIN permission on the server and membership in the role from which the member is being dropped. However to successfully execute this Stored Procedure the pure membership in the role from which a user should be removed is sufficient. The administrator should be aware of the fact that a login who is added to a server role does in this way implicitly inherit the permission to remove all other logins from that role.
- The description of the CREATE LOGIN statement in [AGD, ms-xhelp://?method=page&id=eb737149-7c92-4552-946b-91085d8b1b01&product=SQLServer&productversion=110] describes that the ALTER ANY LOGIN permission on the server is needed. However – as an exception – the CREATE LOGIN statement can also be executed by a user to create a login for her own Windows account (in this case the user would have access due to the membership in a Windows group).
- Please note that database names as retrieved by metadata function DB_NAME are visible to all users. Therefore care should be taken that the database name does not leak information about its contents.

7.3 Recommendations/requirements about Identification and Authentication (Secure Passwords)

- The administrator(s) shall ensure that passwords for all accounts (service accounts, user accounts and administrative accounts) are of sufficient quality. General guidance, how to create strong passwords can be found under https://www.microsoft.com/security/pc-security/password-checker.aspx.
- The concrete settings for the enforcement of minimum password requirements on the underlying Operating System depend on the concrete installation. To allow the secure operation of the TOE the administrator shall ensure that the OS enforces strong password using not less than the following settings:
  - Password must be at least 8 characters in length

    - "password must meet complexity requirements" setting of the OS is enabled. This will ensure that passwords:
        - Do not contain all or part of the user's account name
        - Contain characters from three of the following four categories:
            - English uppercase characters (A through Z)
            - English lowercase characters (a through z)
            - Base 10 digits (0 through 9)
            - Non-alphabetic characters (for example, !, $, #, %)
- The SQL Server engine supports the enforcement of password policies for SQL Server logins based on the policies of the underlying Operating System. This option shall be enabled by using the ALTER or CREATE LOGIN command for each login as follows: 'CHECK_POLICY=on'.

7.4 Other Recommendations and requirements

- It is recommended that beside the accounts that are necessary for the administration of the database engine no accounts are created on the machine that the database engine is operating on. Specifically there shall not be any user accounts for users of the database engine that would allow a direct access to the Operating System.
- It should be noted that any changes to logins that occur while a user is connected to the database engine may require the user to log off and log on again before the updated settings take effect. The administrator should therefore consider to terminate a user session (using the KILL command, see also [AGD, ms-xhelp://?method=page&id=071cf260-c794-4b45-adc0-0e64097938c0&product=SQLServer&productversion=110]) in case of important changes to the login of that user (e.g. the change of group memberships of a user). Further it is possible that sessions are cached after a user disconnected and that a cached session may be reused in case a user logs in again. Changes to login may not be applied to cached sessions under certain circumstances. To avoid this behaviour the administrator shall consider to run the command "DBCC FREESYSTEMCACHE 'ALL'" after important changes to one or more logins. If the server is involved in scenarios of distributed queries the administrator shall further consider to run the "DBCC FREESESSIONCACHE" command in those cases.
- The Service Broker and Database Mirroring endpoints can be used to circumvent the Security Functionality of the TOE. Therefore the administrator shall not install applications on the TOE that make the TSF or any data controlled by the TSF accessible through these endpoints.
- Per default the connections to the database engine are not encrypted and the encryption features of SQL Server 2012 have not been considered during the

evaluation. Thus the administrator has to ensure that all connections to the database engine are appropriately protected, e.g. by using and enforcing an encrypted connection or by using a physically secured connection.

- The use of the column data types text/ntext and image is a deprecated feature (see also [AGD, ms-xhelp:///?method=page&id=B0D8769C-7598-4F97-8162-ACE5F182B5BC&product=SQLServer&productversion=110]) and has not been considered during the evaluation and certification process with respect to the access control functionality. Therefore the administrator shall ensure that user defined objects do not use this data type. The following SQL query can be used to show all columns that use this data type within the current database.
  select b.name, a.name from sys.columns a inner join sys.objects b on a.object_id = b.object_id where b.is_ms_shipped=0 and (a.user_type_id=35 or a.user_type_id=99 or a.user_type_id=34)
- Please note that it is possible that after the evaluation and certification process of the TOE as described in this document additional security patches are issued. Therefore the administrator shall regularly visit the Microsoft technet website (http://www.microsoft.com/technet/security/current.aspx) to get informed about new security bulletins. For each new security patch the administrator shall carefully consider to install it (depending on the needs of the specific installation). The authenticity of each downloadable package can be verified using the digital signature of the file: a file can be considered authentic if it is digitally signed by Microsoft Corporation.
- The Microsoft technet also has a site that explains, how the development group of Microsoft products can be contacted for the case that an administrator finds a security bug (https://www.microsoft.com/technet/security/bulletin/alertus.aspx).

8 Appendix

8.1 Stored Procedures

The following chapters contain information on Stored Procedures that are contained in SQL Server 2012 but not documented in [AGD].

All these Stored Procedures have been developed for internal use only and are documented for information purposes only. These Stored Procedures are not officially supported by Microsoft and no future compatibility is guaranteed.

8.1.1 sp_MSgetversion

This Stored Procedure can be used to get the current version of Microsoft SQL Server.

Input:	no input parameters
Returns:	0 / Error number
Output:	row(s) with the Version Number in Character_value
Syntax:	exec sp_MSgetversion

8.1.2 xp_dirtree

Returns a complete listing of all subdirectories on the server; for each subdirectory listed its depth in the directory tree is also returned. If a *depth* is specified then only subdirectories up to and including the specified depth will be returned. If *IncludeFiles* is specified (as a 1) then files will also be returned and the result set will include an additional column to indicate if a row is a file or a directory.

Input:	@filepath, @depth, @IncludeFiles
Output:	subdirectory, depth, file

Note: file is only displayed if @IncludeFiles = 1

Permission If the calling user is ‘sa’ this Stored Procedure is executed in the context of the SQL Server system account. In all other cases the Stored Procedure will be executed in the context of the calling user (i.e. the Stored Procedure will impersonate the user). This impersonation will fail for the case that a SQL login is used and an empty set will be returned.

Syntax: xp_dirtree <filepath>, <depth>, <IncludeFiles>

Examples:	exec xp_dirtree 'c:'	- Lists all dirs and sub-dirs on C:
	exec xp_dirtree 'c:', 1	- Lists all dirs at the root level of C:
	exec xp_dirtree 'c:', 1, 1	- Lists all dirs and files at the root level of C:

8.1.3 xp_fileexist

This Stored Procedure can be used to determine whether a particular file exists on disk or not.

Input: <filename>

Result: 0 / Error number

Permission If the calling user is ‘sa’ this Stored Procedure is executed in the context of the SQL Server system account. In all other cases the Stored Procedure will be executed in the context of the calling user (i.e. the Stored Procedure will impersonate the user). This impersonation will fail for the case that a SQL login is used and an empty set will be returned.

Syntax: EXECUTE xp_fileexist <filename> [, <file_exists INT> OUTPUT]

Example: For example, to check whether the file boot.ini exists on disk c: or not, run:

EXEC master..xp_fileexist 'c:\boot.ini'

8.1.4 xp_fixeddrives

Returns a row for each fixed drive containing the drive name and the amount of disk space available in MB.

Input: no input parameters

Output: (two columns – drive, MB free)

Permission If the calling user is ‘sa’ this Stored Procedure is executed in the context of the SQL Server system account. In all other cases the Stored Procedure will be executed in the context of the calling user (i.e. the Stored Procedure will impersonate the user). This impersonation will fail for the case that a SQL login is used and an empty set will be returned.

Syntax: exec @retval=xp_fileexist

Example: To see the list of drives, run:

EXEC master..xp_fixeddrives

8.1.5 xp_getnetname

This extended stored procedure returns the WINS name of the SQL Server that you're connected to.

Input: no input parameters

Output: (optional) one column (Server Net Name)

Else single-row, single-column result set is returned

Syntax: exec xp_getnetname

8.1.6 xp_qv

This Stored Procedure wraps SQLBOOT's QueryProductValue function.

USAGE: xp_qv '<setting>' [, '<instancename>']

If the optional instance name is not provided, then the default instance ('MSSQLSERVER') is assumed.

RETURNS: A signed int return value from QueryProductValue or VALUE_ERROR (-1), if an error occurred. VALUE_NOT_FOUND (-2) is returned if the input value is not a valid VALUE_* const.

Example:

```
declare @sqlbootvalue int
exec @sqlbootvalue = xp_qv '2745196162'
select @sqlbootvalue 'VALUE_REPLICATION'
```

8.1.7 xp_instance_regread

See xp_regread for details

8.1.8 xp_regread

Functionality: This Stored Procedure is used to read from the registry.

Input: @rootkey, @key, @value_name, [,@value] (can have 5 input parameters)

Comments: Error if <2 input parameters

5th param – “no_output” then no output is displayed

No error check if >5 params are given

Permission If the calling user is ‘sa’ this Stored Procedure is executed in the context of the SQL Server system account. The Stored Procedure ensures that other users are only granted access to a limited set of registry values.

Return: 0/ Error number

Syntax: EXECUTE xp_regread [@rootkey=]'rootkey', [@key=]'key' [, @value_name=]'value_name'] [, [@value=]@value OUTPUT]

Example: To read into the variable @test from the value 'TestValue' from the key 'SOFTWARE\Test' from the 'HKEY_LOCAL_MACHINE', run:

```
DECLARE @test varchar(20)
EXEC master..xp_regread @rootkey='HKEY_LOCAL_MACHINE', @key='SOFTWARE\Test', @value_name='TestValue', @value=@test OUTPUT
SELECT @test
```

8.1.9 sp_enable_sql_debug

Functionality: Returns a marshaled COM interface pointer that implements IHostDebugServerInstance, as varbinary(8000). IHostDebugServerInstance is the entry point to the integrated Transact-SQL/CLR debugging interfaces. A debugger calls sp_enable_sql_debug and then unmarshals the returned blob to get IHostDebugServerInstance. All methods of IHostDebugServerInstance and related interface implementations in SQL

Server verify the caller is ‘sa’ and return E_ACCESSSDENIED if the check fails.

This Stored Procedure has been developed for debugging purposes only and must not be used in a productive environment.

Input: none

Permission Only ‘sa’ can call this stored procedure; otherwise permission error 300 will be returned.

Syntax: sp_enable_sql_debug @interface_blob output

Example:

```
declare @v varbinary(8000);
exec master.dbo.sp_enable_sql_debug @v output;
select @v
```

8.2 References

Reference	Title	Version	Date
[AGD]	SQL Server Books Online	-	December 2012
[ST]	Security Target, SQL Server 2012 Common Criteria Evaluation	1.0	2013-01-11
[WEB]	https://www.microsoft.com/sqlserver/en/us/common-criteria.aspx (tab “SQL Server 2012 SP1”)	-	-
[PERM]	permission_hierarchy.zip, Available via [WEB]	-	-