



**Microsoft SQL Server 2022
Database Engine
Common Criteria Evaluation (EAL4+)**

Guidance Addendum

Author:	Wolfgang Peter (Microsoft Corporation)
Version:	1.2
Date:	2023-04-14

Abstract

This document is the Guidance Addendum for the Common Criteria certification of the Microsoft SQL Server 2022 Database Engine Enterprise Edition x64 (English).

Keywords

CC, AGD, Common Criteria, SQL, Operational Guidance, DBMS, Database Management System

© 2023 Microsoft Corporation. All rights reserved. This data sheet is informational purposes only. Microsoft makes no warranties, express or implied, with respect to the information presented here.

This page intentionally left blank

Table of Contents

1	INTRODUCTION.....	8
1.1	IDENTIFICATION	8
2	SCOPE OF THE EVALUATION.....	9
2.1	SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT.....	9
2.1.1	<i>Trusted and competent Administrator</i>	10
2.1.2	<i>Protection of information</i>	11
2.1.3	<i>No General-purposes capabilities</i>	11
2.1.4	<i>Physical Protection</i>	11
2.1.5	<i>Trusted identification and authentication information</i>	12
2.1.6	<i>Remote IT System Policies</i>	12
2.1.7	<i>Remote Trusted System</i>	12
3	INSTALLATION AND START-UP GUIDE.....	13
3.1	PREREQUISITES	13
3.1.1	<i>Hardware Prerequisites</i>	13
3.1.2	<i>Software Prerequisites</i>	13
3.1.3	<i>TOE Delivery</i>	13
3.2	SQL SERVER 2022 INSTALLATION.....	14
3.2.1	<i>Checking the integrity and the signature of the TOE parts</i>	14
3.2.2	<i>Installing the product</i>	16
3.2.3	<i>Installing the Documentation Contents</i>	32
3.2.4	<i>Installing the Cumulative Update</i>	36
3.2.5	<i>Querying the installed version</i>	42
3.2.6	<i>Enabling the certified version</i>	42
3.2.7	<i>Installing the logon triggers</i>	43
3.2.8	<i>Hardening execution permissions of xp_dirtree</i>	45
4	SQL SERVER TECHNICAL DOCUMENTATION	46
5	GUIDANCE ADDENDUM	47
5.1	SQL SERVER START-UP FLAGS	47
5.2	ADMINISTRATOR SERVER MANAGEMENT ROLES.....	50
5.3	USER DATABASE MANAGEMENT ROLES	53
5.4	RELEVANT FUNCTIONS FOR ADMINISTRATOR SERVER ROLES.....	54
5.4.1	<i>Sysadmin role</i>	55
5.4.2	<i>Serveradmin role</i>	55
5.4.3	<i>Securityadmin role</i>	56
5.4.4	<i>Processadmin role</i>	56
5.4.5	<i>Setupadmin role</i>	56
5.4.6	<i>Bulkadmin role</i>	56
5.4.7	<i>Diskadmin role</i>	57
5.4.8	<i>Dbcreator role</i>	57
5.4.9	<i>Public role</i>	57
5.5	RELEVANT FUNCTIONS FOR DATABASE MANAGEMENT ROLES	58
5.5.1	<i>Db_owner role</i>	58

5.5.2	<i>Db_securityadmin</i> role	58
5.5.3	<i>Db_accessadmin</i> role	59
5.5.4	<i>Db_backupoperator</i> role	59
5.5.5	<i>Db_ddladmin</i> role	60
5.5.6	<i>Db_datawriter</i> role	62
5.5.7	<i>Db_datareader</i> role	62
5.5.8	<i>Db_denydatawriter</i> role	62
5.5.9	<i>Db_denydatareader</i> role	63
5.6	RELEVANT FUNCTIONS FOR FIXED SERVER-ROLES (##MS_) INTRODUCED IN SQL SERVER 2022..	63
6	SQL SERVER AUDIT	65
6.1	SERVER AUDIT	65
6.1.1	<i>Examples of Use</i>	65
6.1.2	<i>Reviewing the Audit Log and Audit Record Contents</i>	66
6.2	SERVER AUDIT SPECIFICATION	67
6.2.1	<i>Configuration</i>	70
6.3	DATABASE AUDIT SPECIFICATION	71
6.4	SECURITY RELEVANT EVENTS	71
7	SECURITY SERVER POLICIES CONFIGURATION	72
8	REQUIREMENTS FOR SECURE ADMINISTRATION, CONFIGURATION AND USAGE	77
8.1	REQUIREMENTS ABOUT SECURITY AUDIT	77
8.2	REQUIREMENTS AND FURTHER INFORMATION ABOUT ACCESS CONTROL	78
8.3	REQUIREMENTS ABOUT IDENTIFICATION AND AUTHENTICATION (SECURE PASSWORDS)	79
8.4	OTHER REQUIREMENTS	79
9	APPENDIX	82
9.1	STORED PROCEDURES	82
9.1.1	<i>sp_Msgetversion</i>	82
9.1.2	<i>xp_dirtree</i>	82
9.1.3	<i>xp_fileexist</i>	82
9.1.4	<i>xp_fixeddriives</i>	83
9.1.5	<i>xp_getnetname</i>	83
9.1.6	<i>xp_qv</i>	83
9.1.7	<i>xp_instance_regread</i>	84
9.1.8	<i>xp_regread</i>	84
9.1.9	<i>Sp_remove_maximum_number_of_connections_limit</i>	84
9.1.10	<i>Sp_revoke_logon_denies</i>	85
9.1.11	<i>Sp_set_maximum_number_of_connections_per_login</i>	85
9.1.12	<i>Sp_deny_logon</i>	85
9.1.13	<i>sp_enable_sql_debug</i>	86
10	REFERENCES	87

List of Tables

	Page
Table 1: Security Objectives in the operational environment	10
Table 2: Hash values for deliverables	15
Table 3: Entry Points into SQL Server Technical Documentation	46
Table 4: Start-up Options for "sqlservr.exe"	49
Table 5: Administrator server roles	51
Table 6: Fixed server-roles introduced in SQL Server 2022	53
Table 7: Database roles	54
Table 8: Relevant functions for Administrator server roles	55
Table 9: Sysadmin permissions	55
Table 10: <i>Serveradmin</i> permissions	56
Table 11: <i>Securityadmin</i> permissions	56
Table 12: <i>Processadmin</i> permissions	56
Table 13: <i>Setupadmin</i> permissions	56
Table 14: <i>Bulkadmin</i> permissions	57
Table 15: <i>Diskadmin</i> permissions	57
Table 16: <i>Dbcreator</i> permissions	57
Table 17: <i>Public</i> role permissions	57
Table 18: Relevant functions for database roles	58
Table 19: <i>Db_securityadmin</i> permissions	59
Table 20: <i>Db_accessadmin</i> permissions	59
Table 21: <i>Db_backupoperator</i> permissions	60
Table 22: <i>Db_ddladmin</i> permissions	61
Table 23: <i>Db_datawriter</i> permissions	62
Table 24: <i>Db_datareader</i> permissions	62
Table 25: <i>Db_denydatawriter</i> permissions	62
Table 26: <i>Db_denydatareader</i> permissions	63
Table 27: <i>##MS_</i> roles permissions	64
Table 28: Audit Record	67
Table 29: Audit Events	70

List of Figures

	Page
Figure 1: Microsoft Download Center	14
Figure 2: Successful verification of integrity	15
Figure 3: Checking the digital signature	16

Figure 4: Installing SQL Server 2022	17
Figure 5: Installing SQL Server 2022 (I)	17
Figure 6: Installing SQL Server 2022 (II)	18
Figure 7: Installing SQL Server 2022 (III)	19
Figure 8: Installing SQL Server 2022 (IV)	20
Figure 9: Installing SQL Server 2022 (V)	21
Figure 10: Installing SQL Server 2022 (VI)	22
Figure 11: Installing SQL Server 2022 (VII)	23
Figure 12: Installing SQL Server 2022 (VIII)	23
Figure 13: Installing SQL Server 2022 (IX)	24
Figure 14: Installing SQL Server 2022 (X)	25
Figure 15: Installing SQL Server 2022 (XI)	26
Figure 16: Installing SQL Server 2022 (XII)	27
Figure 17: Installing SQL Server 2022 (XIII)	28
Figure 18: Installing SQL Server 2022 (XIV)	29
Figure 19: Installing SQL Server 2022 (XV)	30
Figure 20: Installing SQL Server 2022 (XVI)	31
Figure 21: Installing SQL Server 2022 (XVII)	32
Figure 22: Setting Help Preference in SQL Server Management Studio	33
Figure 23: Choosing the installation source for the contents in Help Viewer	33
Figure 24: Retrieving relevant content from the Technical Documentation	34
Figure 25: Search tab	34
Figure 26: Local Server Settings	35
Figure 27: IE Enhanced Security Configuration	36
Figure 28: Updating SQL Server 2022 (I)	37
Figure 29: Updating SQL Server 2022 (II)	38
Figure 30: Updating SQL Server 2022 (III)	39
Figure 31: Updating SQL Server 2022 (IV)	40
Figure 32: Updating SQL Server 2022 (V)	41
Figure 33: Complete installation of Cumulative Update	42
Figure 34: Tools option selection	72
Figure 35: Local Security Policy option selection	72
Figure 36: Account Policies modification	73
Figure 37: Number of attempts modification	73
Figure 38: Selection of attempts	74
Figure 39: Suggested values change	74
Figure 40: Password Policy Tab	75

Figure 41: Change password duration to 30 days 75

Figure 42: Change password length to 12 characters 76

1 Introduction

This document has been created as part of the Common Criteria (CC) Evaluation of Microsoft SQL Server 2022 Database Engine Enterprise Edition x64 (English). It covers the specific aspects that shall be considered when operating SQL Server 2022 in its certified version and extends the general guidance of the product given in SQL Server Technical Documentation. The document follows the following structure:

Chapter 2 of this document gives more details about the scope of the certification for SQL Server 2022 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 2022 in its certified version.

Chapter 4 introduces the concept of the SQL Server Technical Documentation 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 2022.

Chapter 6 introduces the concept and the important aspects of the audit mechanism of SQL Server 2022.

Chapter 7 contains the appropriate configuration of the Security Server Policies to avoid some possible attacks to the TOE.

Finally, **chapter 8** gives requirements for the secure operation and proper configuration of the TOE.

1.1 Identification

Document title	Microsoft SQL Server 2022 Database Engine - Common Criteria Evaluation (EAL4+) – Guidance Addendum
Version	1.2
TOE name	Microsoft SQL Server 2022 Database Engine Enterprise Edition x64 (English)
TOE version	16.0.4025.1

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 SQL Server 2022 Enterprise Edition x64 (English) and its related guidance documentation. Please see [ST, chapter 1.1] 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 comprises the complete set of security functions as described in [ST, chapter 6.1] except the FPT_TRC.1 SFR since the TOE does not contain physically separated parts in the evaluated version.

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

The following chapter describes the security objectives, which have been made about the operational 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 security objectives can be addressed.

2.1 Security objectives for the Operational Environment

According to [ST] the following security objectives for the TOE environment are defined in the table below:

Objective	Description
OE.ADMIN	Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.
OE.INFO_PROTECT	Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequate protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.
OE_NO_GENERAL_PURPOSE	There will be no general-purposes computing 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.

Objective	Description
OE_PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection must be commensurate with the value of the IT assets protected by the TOE.
OE.IT_I&A	Any information provide by a trusted entity in the environment and used to support user authentication and authorization used by the TOE is correct and up to date.
OE.IT_REMOTE	If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide that the functions and any data used by the TOE in making policy decisions, required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.
OE.IT_TRUSTED_SYSTEM	The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. These remote trusted IT systems are managed according to known, accepted, and trusted policies based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.

Table 1: Security Objectives in the operational environment

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

For the current configuration of the TOE there are no remote trusted IT systems or separate parts of the TSF involved.

2.1.1 Trusted and competent Administrator

Users responsible of the TOE shall be trustworthy and capable of managing the TOE, security functions and information that is contained on it. Furthermore, these users shall receive a proper training to develop their capabilities related to security concerns on the TOE.

SQL Server users who are members of the public server role without EXECUTE permissions (section 8.2) and no additional permission added, are considered untrusted users. However, if the SQL Server user role membership changes to another server management role or a database management permission is assigned to it, the SQL Server user is considered as a management user and thus become a trusted user, according to the security objective for the operational environment OE.ADMIN in [ST], section 4.2 Security Objectives for the operational Environment.

The users involved in the installation and configuration of the TOE during the preparative procedures will be considered trustworthy and reliable.

This objective will be fulfilled by any role of the TOE, except on users who have the role public without any additional permissions.

2.1.2 Protection of information

Users responsible of the TOE shall establish and implement procedures to ensure the protection of the information in an appropriate manner. There exist some remarkable parts that shall be take into consideration:

- Every network or peripheral cabling must be approved for transmit sensitive data over the link. Physical links shall be protected in a proper way against threats defined in [ST] to ensure the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques.
- Policy control access or DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. This policy control access shall be established in order to create different roles with different privileges on the TOE that rightly control access to sensitive information.

Users shall be proper authorize to access concrete parts of the data managed of the TOE and trained to exercise control over their own data.

This objective does not apply to any specific role of the TOE. This objective will be fulfilled by the secure facility where it is installed the TOE and the people who are in charge of this facility.

2.1.3 No General-purposes capabilities

It shall be ensured that TOE does not have general-purposes computing capabilities (compilers or user applications) available on DBMS servers. TOE only needs services related to operation, administration and support for the DBMS.

Those responsible for the TOE must ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection must be commensurate with the value of the IT assets protected by the TOE.

This objective does not be accomplished by any specific role. This objective will be fulfilled by the person or people who has/have the task of installation of the TOE.

2.1.4 Physical Protection

It shall be ensured by the responsible of the TOE that critical parts of the TOE are conveniently protected and enforced with an adequate security policy that prevent from a physical attack that can compromise security objectives. Physical protection of the wire can provide a right level of security for the information transmitted between the clients and the TOE as same as other connections from clients to the TOE. The physical security of the machine that contains the TOE is a notorious factor to take into account if the user data should be correctly protected. The machine needs to be installed in a secure place that has convenient physical measures to avoid non-privileged users to gain physical access to the TOE. It is a remarkable fact that an attacker with physical access to the machine could easily gain complete control of user data which is stored into the database.

This objective does not apply to any specific role of the TOE. This objective will be fulfilled by the secure facility where it is installed the TOE and the people who are in charge of this facility.

2.1.5 Trusted identification and authentication information

It shall be ensured that any information used to support user authentication and authorization in the TOE shall be provided by a trusted entity in the environment and shall be correct and up-to-date. It refers to the login information associated with the user accounts of the Windows Server Operating System in the environment.

Due to the TOE does not generate its own timestamps, user(s) responsible for the installation and administration of the TOE shall ensure that the date and time configured in the underlying operating system is correct, because the TOE will use these timestamps as its own.

This objective does not apply to any specific role of the TOE. This objective will be fulfilled by the person or people who is in charge of the TOE and its environment.

2.1.6 Remote IT System Policies

When the TOE is relied on remote trusted IT systems, the TOE shall protect the data integrity of these systems, avoiding false results. In addition, all machines that compose the TOE and its environment (Active Directory, the server machine where the TOE is installed and all client machines that are connected to the TOE) are located in a secure facility with the adequate physical measures. This fact can be done with a strong security policy and powerful integrity mechanisms in the process of sending and receiving information. In this configuration this security objective is not relevant because it does not exist any remote Trusted IT that is connected to the TOE. In other words, the TOE does not have TSF-distributed parts.

Therefore, regarding the separated parts of the TOE, there are non-additional security measures required to fulfil this security objective for the operational environment.

This objective does not apply to any specific role of the TOE. This objective will be fulfilled by the person or people who is in charge of the TOE and its environment.

2.1.7 Remote Trusted System

It shall be ensured that the remote trusted IT system implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. Moreover, these remote IT systems shall be managed according to the same rules and policies applicable to the TOE and also are logical and physically protected with the same security measures than the TOE. In addition, all machines that compose the TOE and its environment (Active Directory, the server machine where the TOE is installed and all client machines that are connected to the TOE) are located in a secure facility with the adequate physical measures. Moreover, in this configuration this security objective is not relevant because there are no remote Trusted IT that are connected to the TOE. In other words, the TOE does not have TSF-distributed parts.

Therefore, regarding the separated parts of the TOE, there are non-additional security measures required to fulfil this security objective for the operational environment.

This objective does not apply to any specific role of the TOE. This objective will be fulfilled by the person or people who is in charge of the TOE and its environment.

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. x64-compatible
- 1 gigabyte (GB) of RAM minimum.
- Approximately 6 GB of available hard-disk space for the recommended installation
- DVD-ROM drive
- SVGA (1,024 x 768) 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).

3.1.2 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 2022 (English), Standard Edition
- Microsoft .NET Framework 4.7.2
- Windows PowerShell 3.0 or higher

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.

3.1.3 TOE Delivery

The TOE is available as a download and can be retrieved through Microsoft's Volume Licensing Service Center (VLSC). The user needs to login to the following address <https://www.microsoft.com/licensing/servicecenter/default.aspx>. Then, choose "SQL Server 2022 Enterprise Edition" in the download area. Download the ISO file and either burn it to a DVD or mount it.

3.2 SQL Server 2022 Installation

3.2.1 Checking the integrity and the signature of the TOE parts

It is assumed that the administrator has already successfully verified the integrity of the SQL Server 2022 Guidance Addendum (this document) as described on [WEB].

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] (click on “View our Common Criteria certification” and a PDF document will be downloaded. Within this PDF, click in the following links):
 - Microsoft SQL Server 2022 Permission Poster: Microsoft_SQL_Server_2017_and_Azure_SQL_Database_permissions_infographic.pdf¹
 - Microsoft SQL Server 2022 Technical Documentation: Offline-Book_SQL-Server-2022_1.0_2022-12-23.zip
 - Installer Triggers Script: SQL22_W_Install_cc_triggers_1.0_2022-12-20.sql
 - Integrity Check Validation Data: hash_dir_1.0_2022-12-20.bat
2. Download Cumulative Update 3 (CU3) (SQLServer2022-KB5024396-x64) from Microsoft Update Catalog website (as it is shown in image Figure 1): <https://www.catalog.update.microsoft.com/Search.aspx?q=sql%20server%202022>

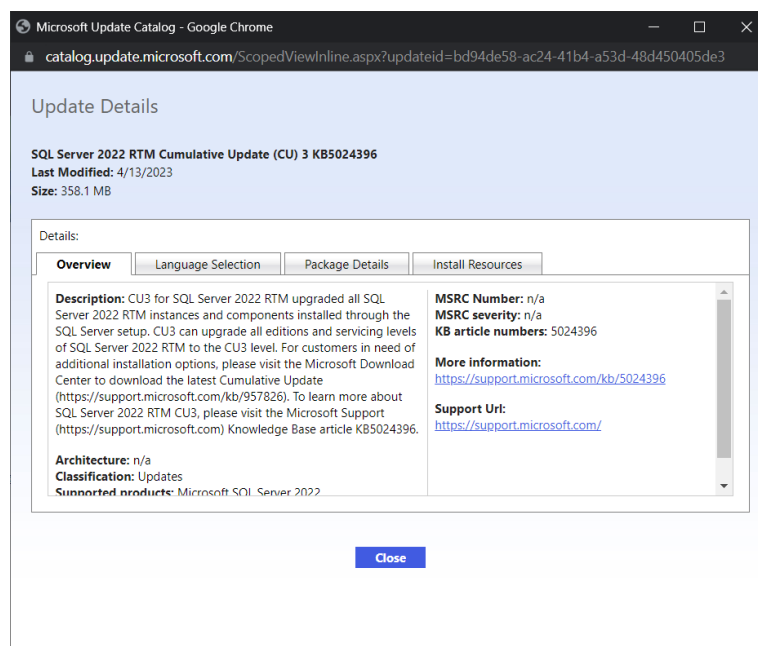


Figure 1: Microsoft Download Center

3. Download the DVD image (SQLServer2022-x64-ENU.iso file) via the Microsoft volume licensing service center (<https://www.microsoft.com/licensing/servicecenter/default.aspx>).

¹ Although the permission poster refers to SQL Server 2017 is also applicable for the evaluated TOE.

4. Calculate the SHA256 hash values for each downloaded file with the following command in a Windows terminal or PowerShell 3.0 application: `certutil.exe -hashfile <file_name> SHA256`
5. Compare the output hash with the following hashes:

File	SHA256 Hash
hash_dir_1.0_2022-12-20.bat	BD9E61C4DCE7775B7999CC313124B5C94770873F49E268880E4206F508B18AEA
Microsoft_SQL_Server_2017_and_Azure_SQL_Database_permissions_infographic_1.0.pdf	4C2119AD0CB54B388D900590351FEB53758139EE6574B50EAB6BEF6192EC368B
Offline-Book_SQL-Server-2022_1.0_2022-12-23.zip	FBB163468C5088CECD0D0808D3BAC9261FCC8CD77E8C8EB8F0F4C6AA2061EE39
SQL22_W_Install_cc_triggers_1.0_2022-12-20.sql	043AC79021C549AB198BE5DB18AC7AE160C0624AA9C870D6F606FA68BE7987C5
SQLServer2022-KB5024396-x64.exe NOTE: SHA1 value may be included as part of the filename	FD9412F77876358473E08C9866F1678DDDC66739A2A9E81C8EC6514D61577405

Table 2: Hash values for deliverables

6. Mount drive the ISO file or burn into a DVD and put it on DVD.
7. The script “hash_dir_1.0_2022-12-20.bat” creates a list of file hashes of files on a specified drive and outputs the hash of this list. For the hash operations it uses the Windows CertUtil tool. Please note that during operation the script creates the file “hash_dir_output.txt” in the directory where it is executed. If the file already exists, it is automatically deleted and a new output file is created.

Execute “hash_dir_1.0_2022-12-20.bat E:” (replace E: by the appropriate drive) and verify that the final SHA256 hash output matches the one in the following picture (Figure 2):

```

Administrator: C:\Windows\System32\cmd.exe
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop>hash_dir_1.0_2022-12-20.bat E:
Hashing contents of drive: E:
.....
SHA256 hash of integrity list:
cdc8463d96d722e00d006df9204d308177e9f1cc1b28d8b6b475af8d53ac9cf
  
```

Figure 2: 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).

Furthermore, before a release or a Service Pack is published, every file which is of one of the following types: *.cab, *.cat, *.ctl, *.dll, *.exe, *.ocx is digitally signed.

After downloading a file from the internet the Administrator can check the digital signature of these files with a right click on it (see Figure 3 which shows the check process for setup.exe).

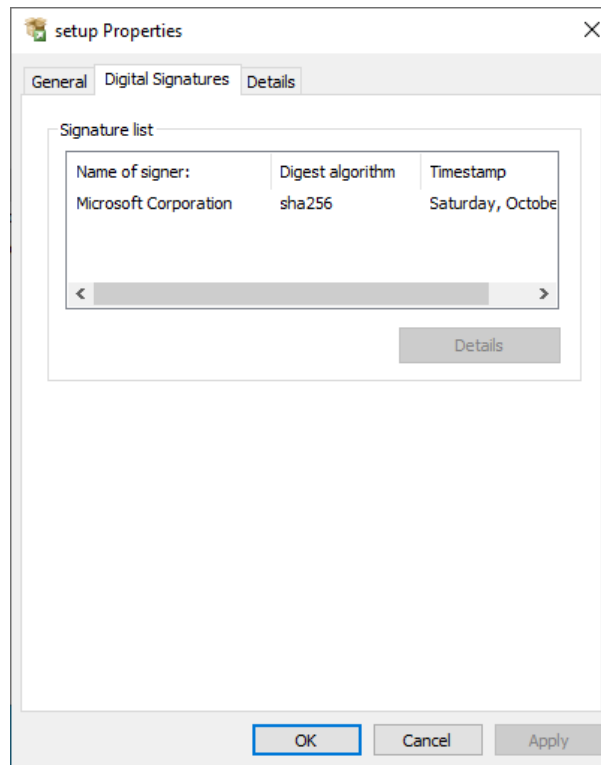


Figure 3: Checking the digital signature

3.2.2 Installing the product

The description in this chapter focuses on a typical way of installing the database engine of SQL Server 2022. For a more general overview over all options for the SQL Server setup please refer to [AGD, section: "Install SQL Server from the Installation Wizard (Setup)"].

Please note that the installation procedure presented in this chapter belongs to the Enterprise Version of SQL Server 2022. The SQL Server Installation Wizard is Windows Installer-based. It provides a single feature tree for the installation of all SQL Server components.

The first step of the installation consists on insert the burnt DVD of the ISO image or mount the ISO image and double-click *setup.exe* in the root folder.

Warning: Setup.exe file needs to be run as administrator.

The Installation Wizard will run the SQL Server Installation Center as seen in the following image. Go to menu Installation in order to begin the installation process.

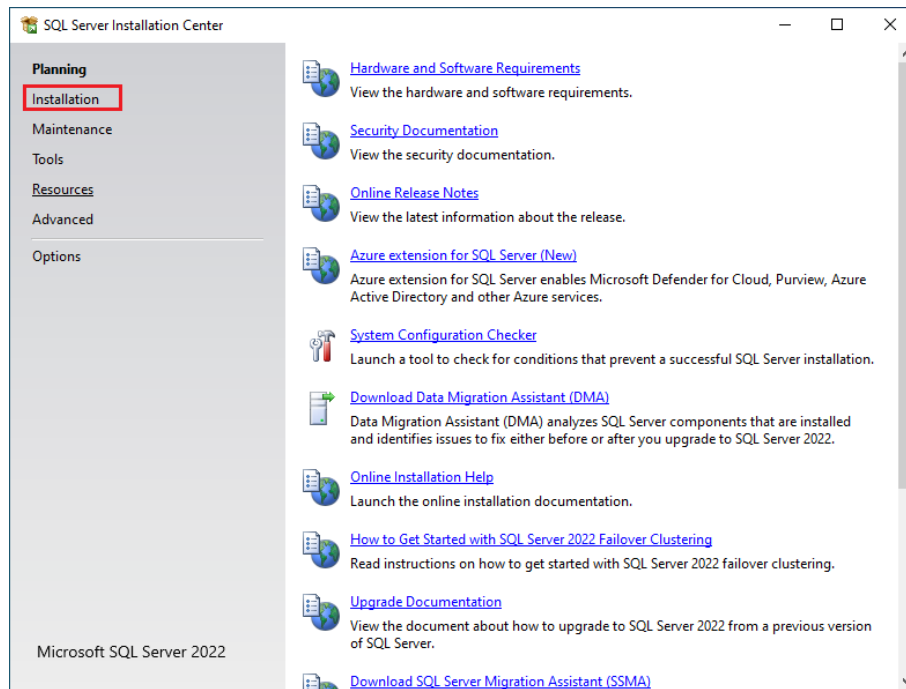


Figure 4: Installing SQL Server 2022

Next, to create a new installation of SQL Server 2022 go to menu Installation and click “New SQL Server stand-alone installation or add features to an existing installation”.

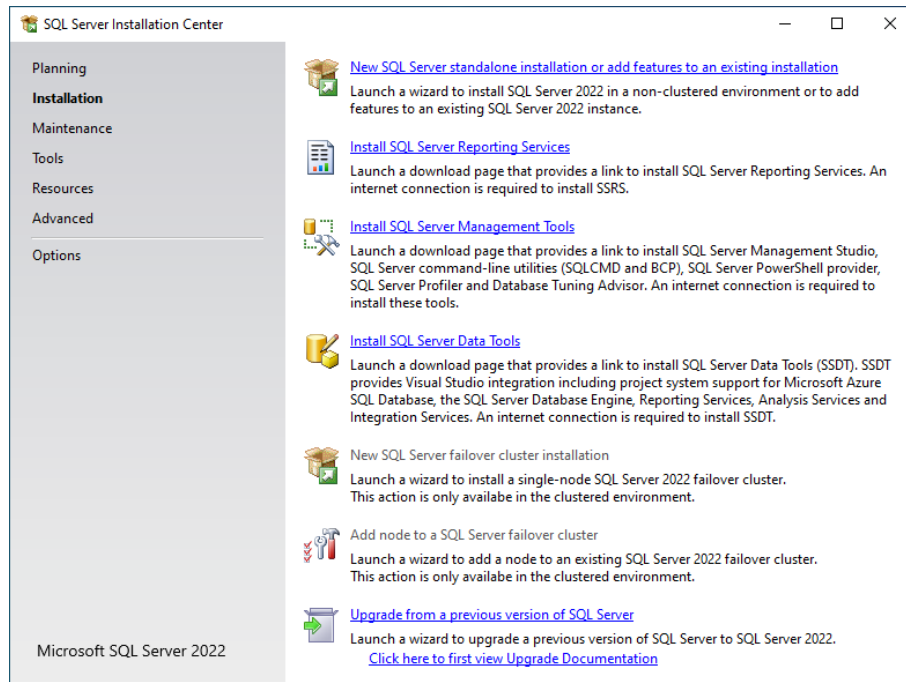


Figure 5: Installing SQL Server 2022 (I)

On the Product Key page (see Figure 6), exists three possible options: installing a free SQL Server Edition or Enterprise, billing through Microsoft Azure or use a key to install the Enterprise Edition. It is

remarkable that only the Enterprise Edition of SQL Server has been certified, so it is necessary to have a product license key to install the TOE.

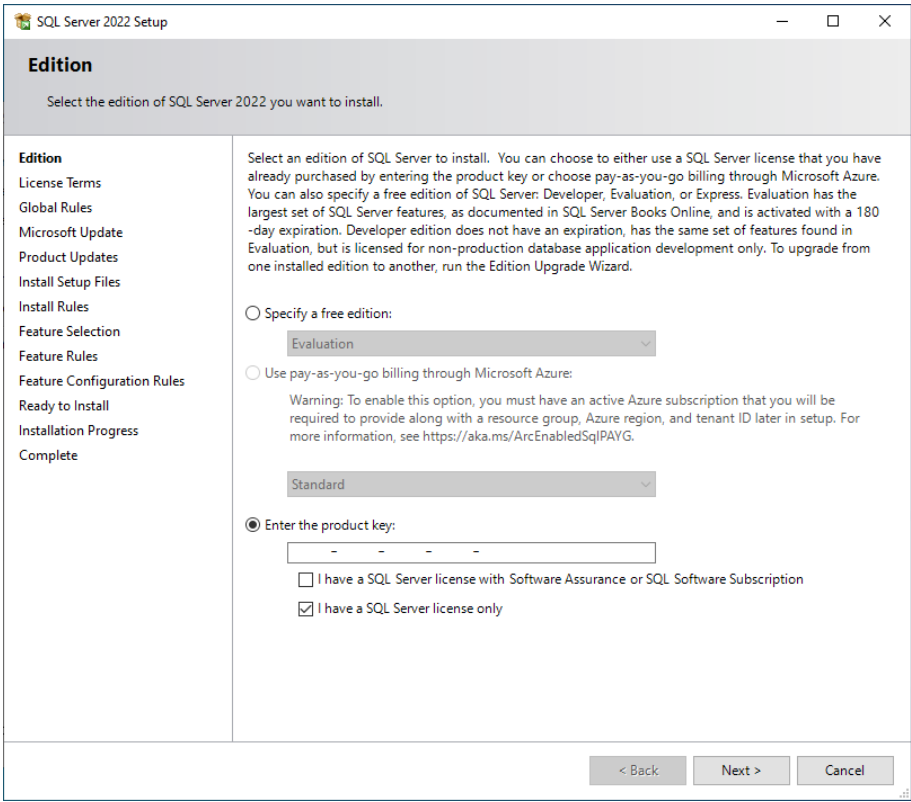


Figure 6: Installing SQL Server 2022 (II)

Next the License Terms page appears (Figure 7), it is important to read carefully the license agreement and then select the check box to accept the licensing terms and conditions.

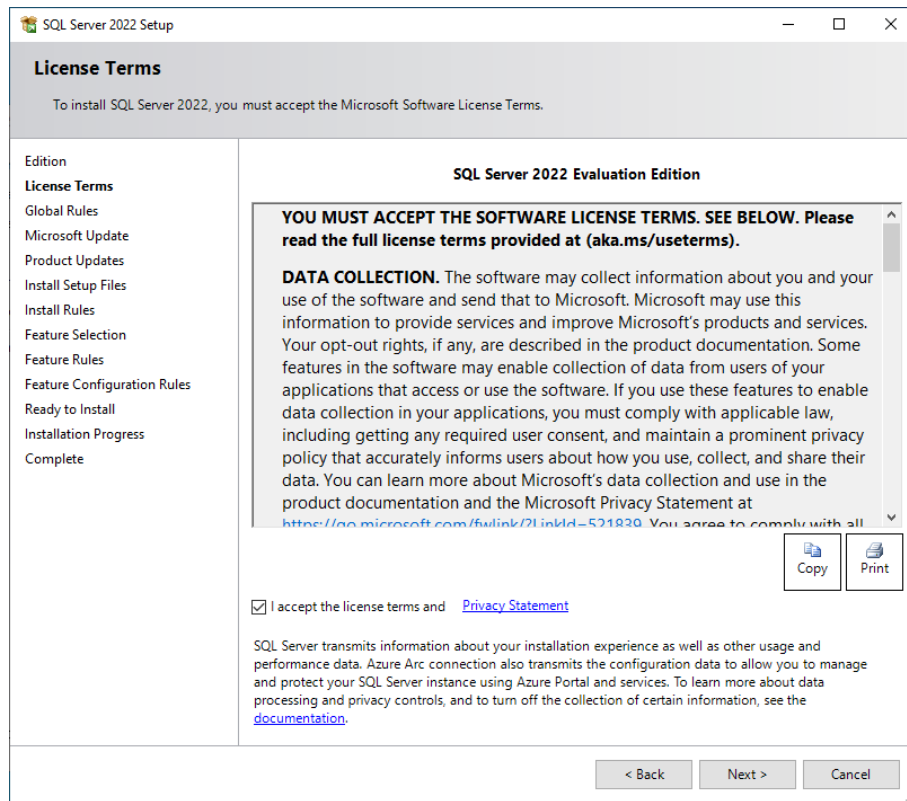
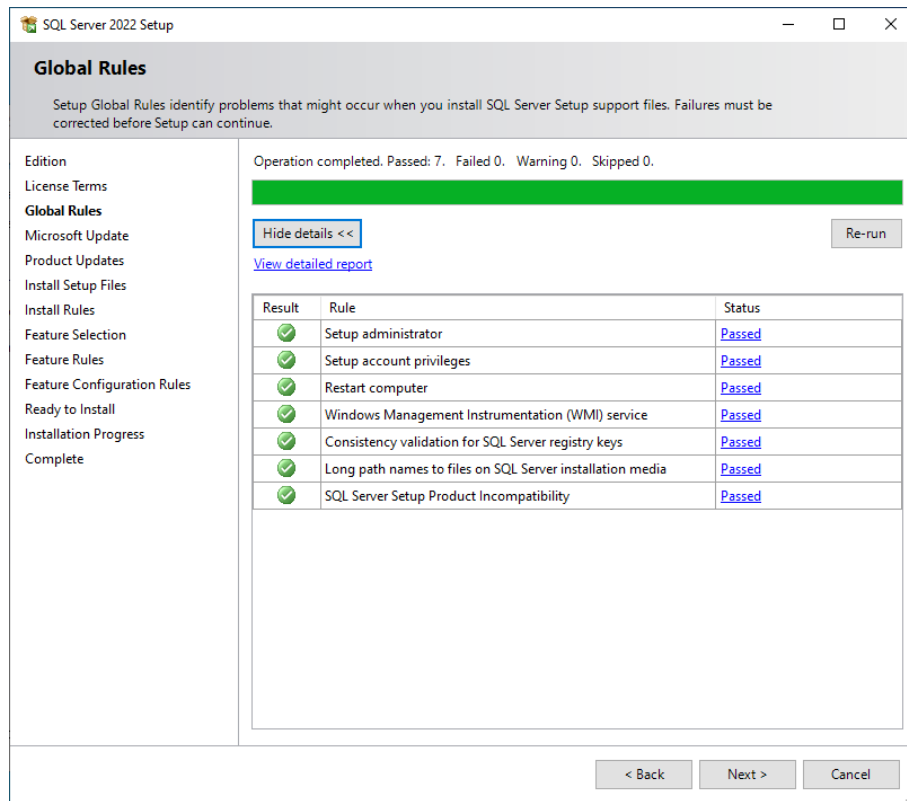


Figure 7: Installing SQL Server 2022 (III)

Next, the installer will perform some checks to identify problems with SQL Server Setup support files (Figure 8).

**Figure 8: Installing SQL Server 2022 (IV)**

When the prerequisites have already been installed, the Installation Wizard asks if Microsoft Update shall be used to check for updates (Figure 9). The installation of updates shall not be enabled because any update version of the TOE leaves the certified version.

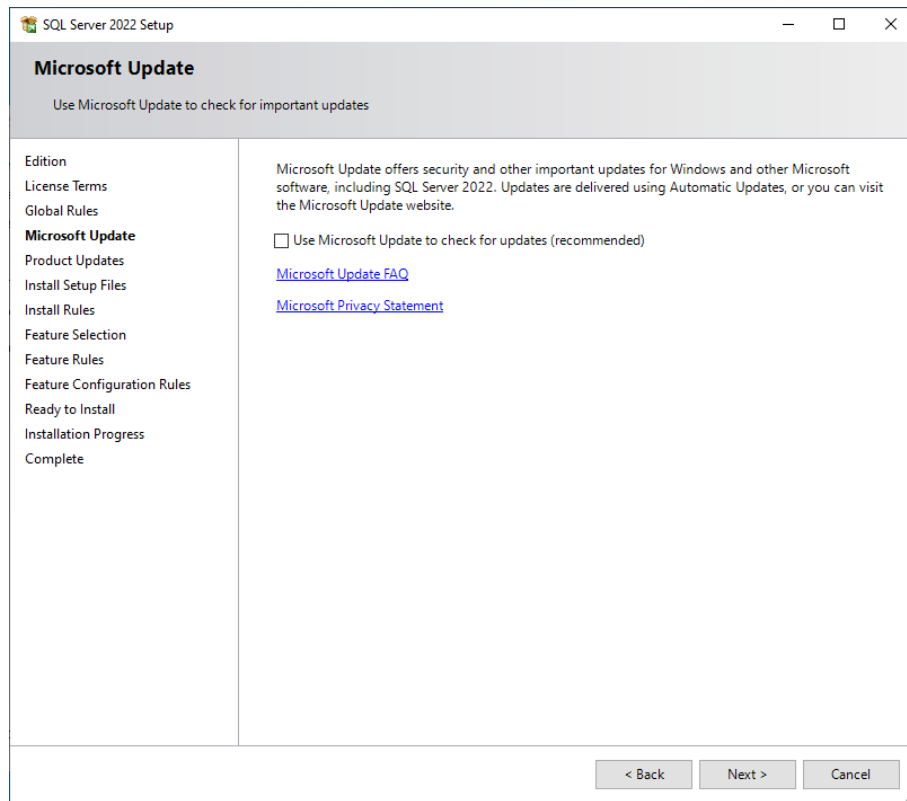


Figure 9: Installing SQL Server 2022 (V)

The setup will now check for product updates through the Windows Update service. For the certified version of SQL Server 2022 no automated product update shall be applied. Therefore, click 'Next' to skip this step.

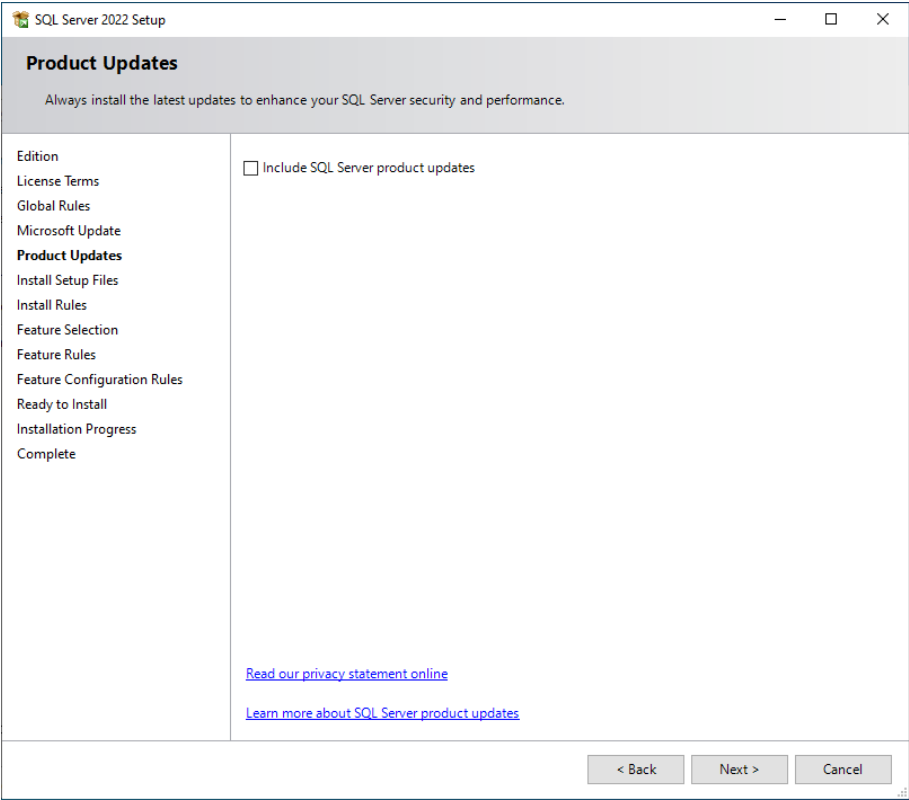


Figure 10: Installing SQL Server 2022 (VI)

The Installation Wizard will then only copy the Setup Files to the hard disk as shown in Figure 11:

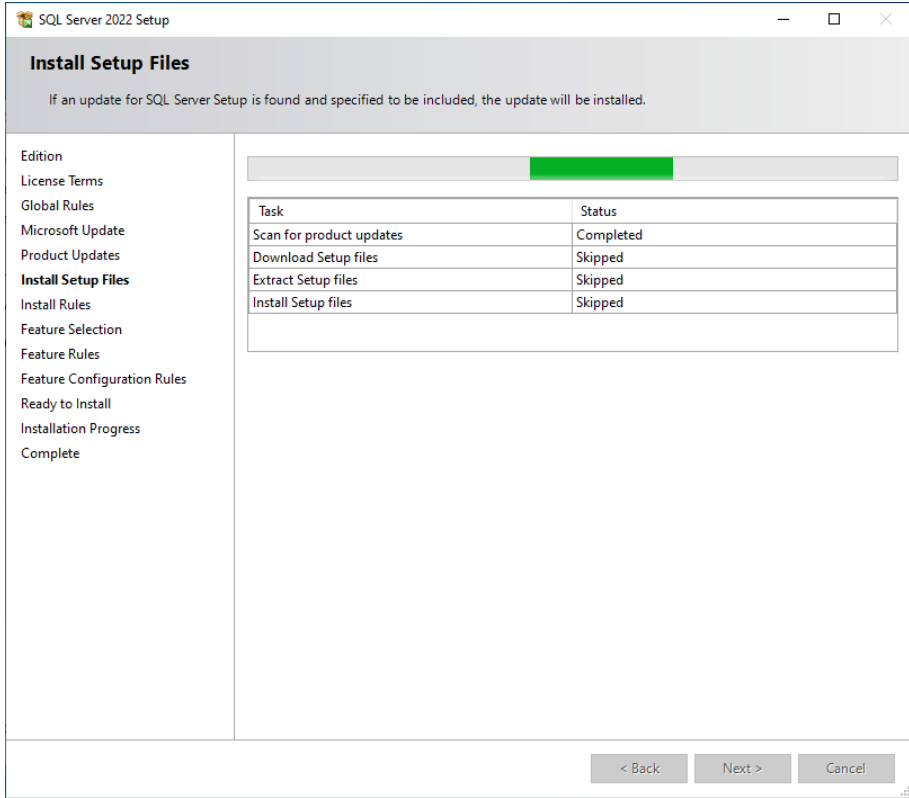
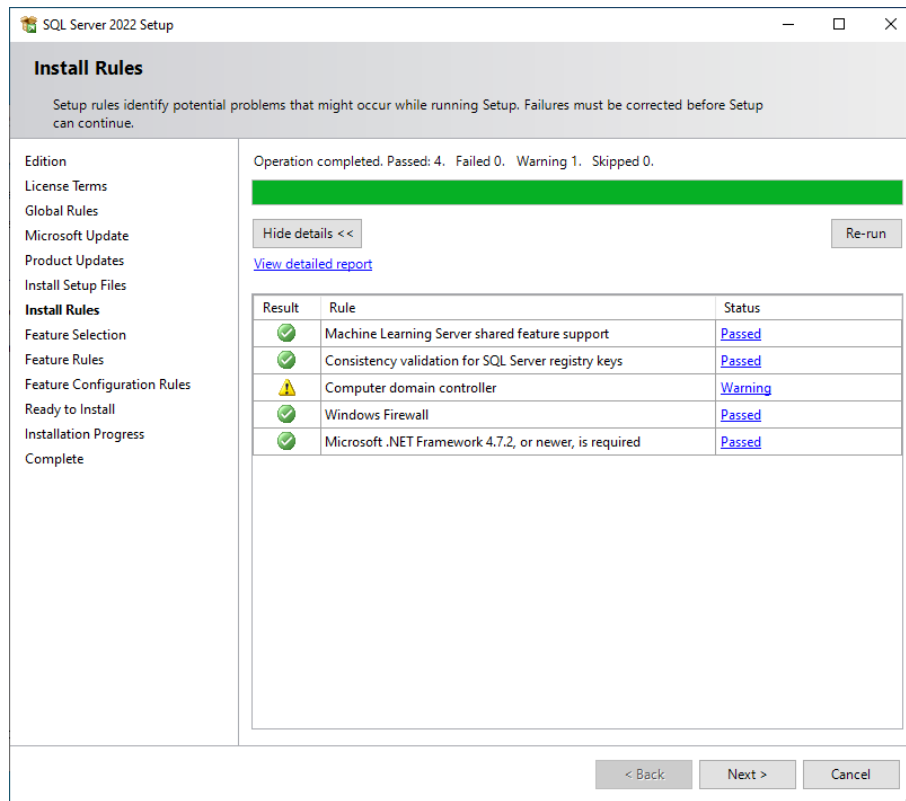


Figure 11: Installing SQL Server 2022 (VII)

The System Configuration Checker will verify the system state of the machine before Setup continues. Warning messages shown by the Configuration Checker (e.g. the one to see in Figure 12 that is shown because the Windows Firewall is active) shall be carefully considered but do not prevent the further installation. For more information, please refer to [AGD, section: “Configure the Windows Firewall to Allow SQL Server Access”].

**Figure 12: Installing SQL Server 2022 (VIII)**

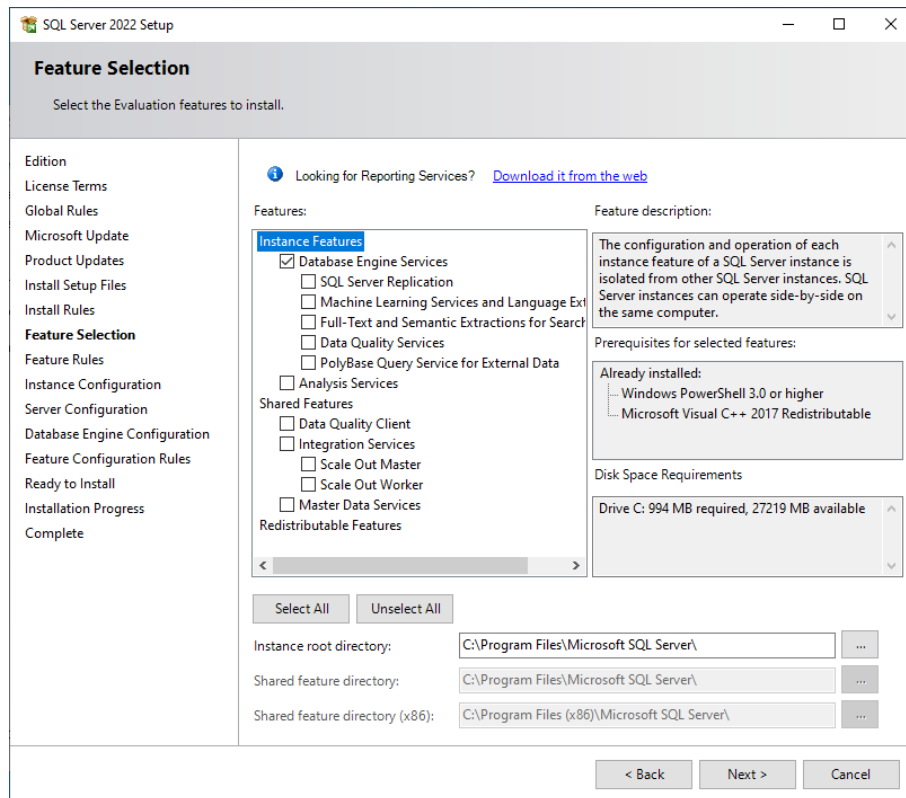
The Azure Extension for SQL Server menu (Figure 13 below) allows to install the Azure Arc-enable SQL extension. To do that, the first checkbox shall be selected and then the Azure account credentials or service principal information need to be provided in order to authenticate the SQL Server instance to Azure. According to [ST,1.3.3] this option shall be enabled to set the TOE in its evaluated configuration.

Figure 13: Installing SQL Server 2022 (IX)

NOTE: Arc-enabled SQL Server is available in the regions detailed here: <https://learn.microsoft.com/en-us/sql/sql-server/azure-arc/overview?view=sql-server-ver16#supported-azure-regions>

On the Feature Selection menu, it can be selected the components for installation. A description for each component group appears in the right-hand pane after selecting the feature name. It can be selected any combination of check boxes. For the certified version of the database engine of SQL Server 2022 the selection of components as shown in Figure 14 is recommended. It will install an instance of the database engine. 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.

A custom directory for shared components can be specified by using the field at the bottom of the Feature Selection menu.

**Figure 14: Installing SQL Server 2022 (X)**

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 (Figure 15).

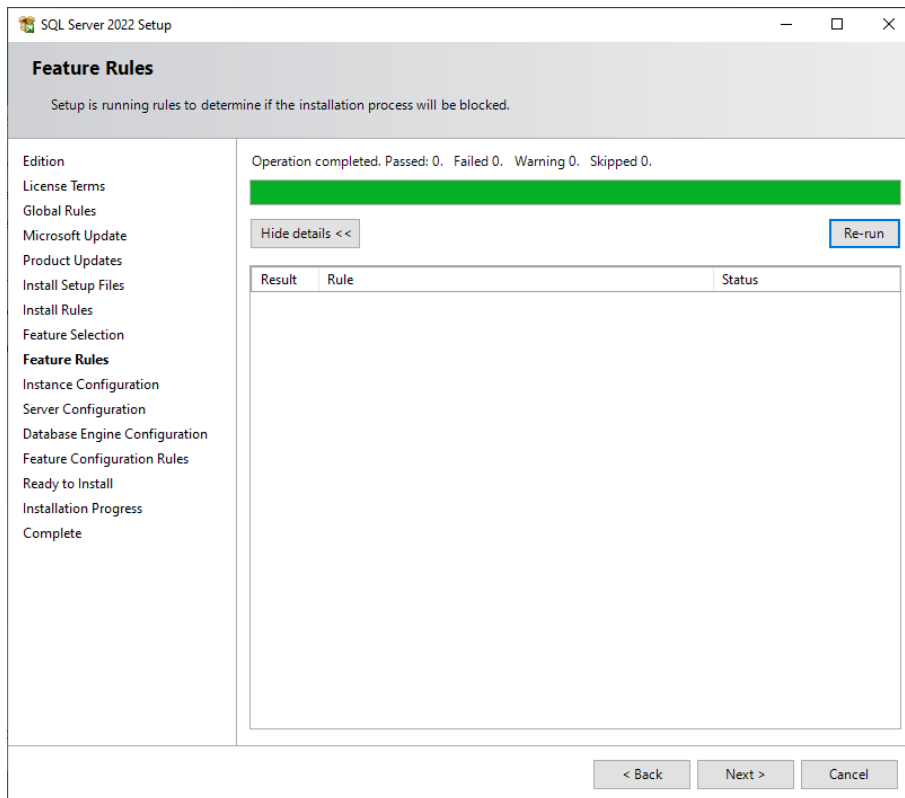


Figure 15: Installing SQL Server 2022 (XI)

On the Instance Configuration page (see Figure 16), can be specified 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 minimize 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: *

Instance ID:

SQL Server directory:

Installed instances:

Instance Name	Instance ID	Features	Edition	Version
---------------	-------------	----------	---------	---------

< Back Next > Cancel

Figure 16: Installing SQL Server 2022 (XII)

On the Server Configuration — Service Accounts page (see Figure 17), it can be specified 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.

It can be assigned the same login account to all SQL Server services, or it is possible to configure each service account individually. Another specification is the start mode of the server: automatic, manual or 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.

Please do not activate the option “Grant Perform Volume Maintenance Task privilege to SQL Server Database Engine Service” because it may result in non CC-compliant behaviour.

The Server Configuration — Collation tab can be used to specify non-default collations for the Database Engine. For more information, see [AGD, section: “Server Configuration Options (SQL Server)”].

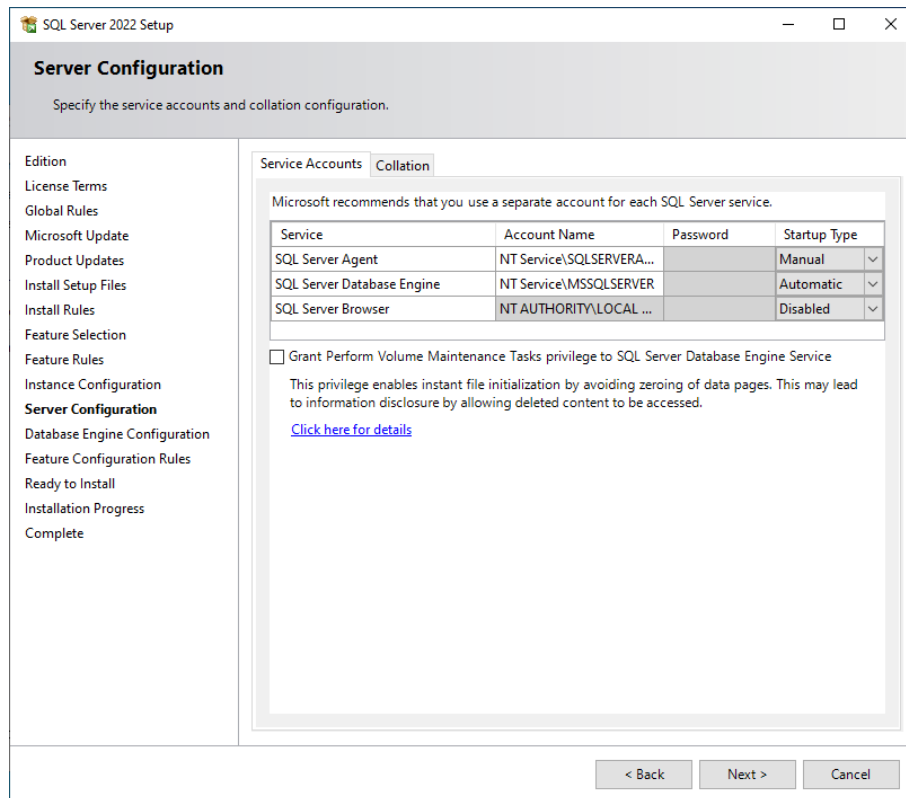


Figure 17: Installing SQL Server 2022 (XIII)

The Database Engine Configuration - Account Provisioning page (see Figure 18) 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, the user 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** — it is compulsory to 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, section: “Configure Windows Service Accounts and Permissions”].
- The Database Engine Configuration - Data Directories page (see Figure 18) can be used to specify non-default installation directories.
- The Database Engine Configuration - FILESTREAM page (see Figure 18) can be used to enable FILESTREAM for your instance of SQL Server.

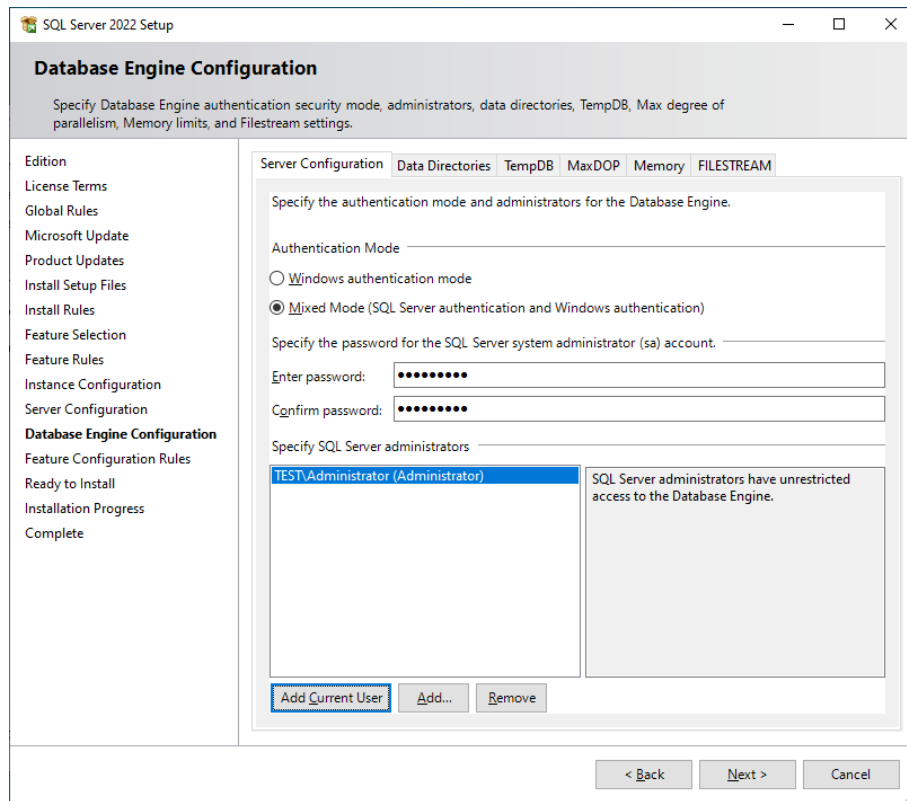


Figure 18: Installing SQL Server 2022 (XIV)

The Ready to Install page (see Figure 19) shows a tree view of installation options that were specified during Setup.

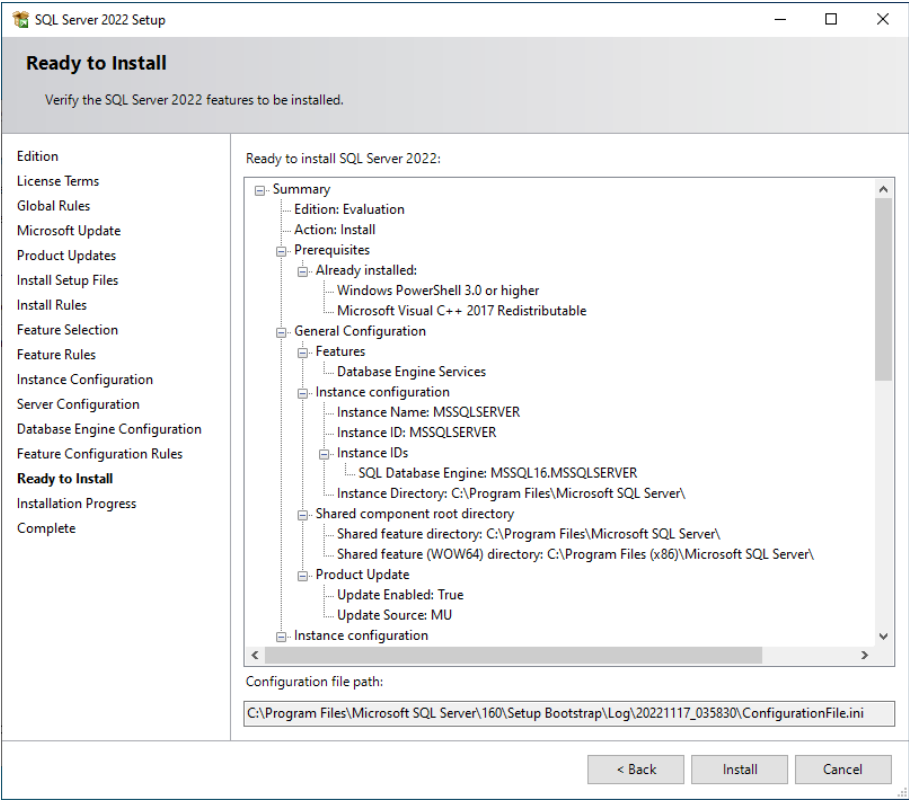


Figure 19: Installing SQL Server 2022 (XV)

During installation, the Installation Progress page provides status in order to control installation progress as Setup proceeds (see Figure 20).

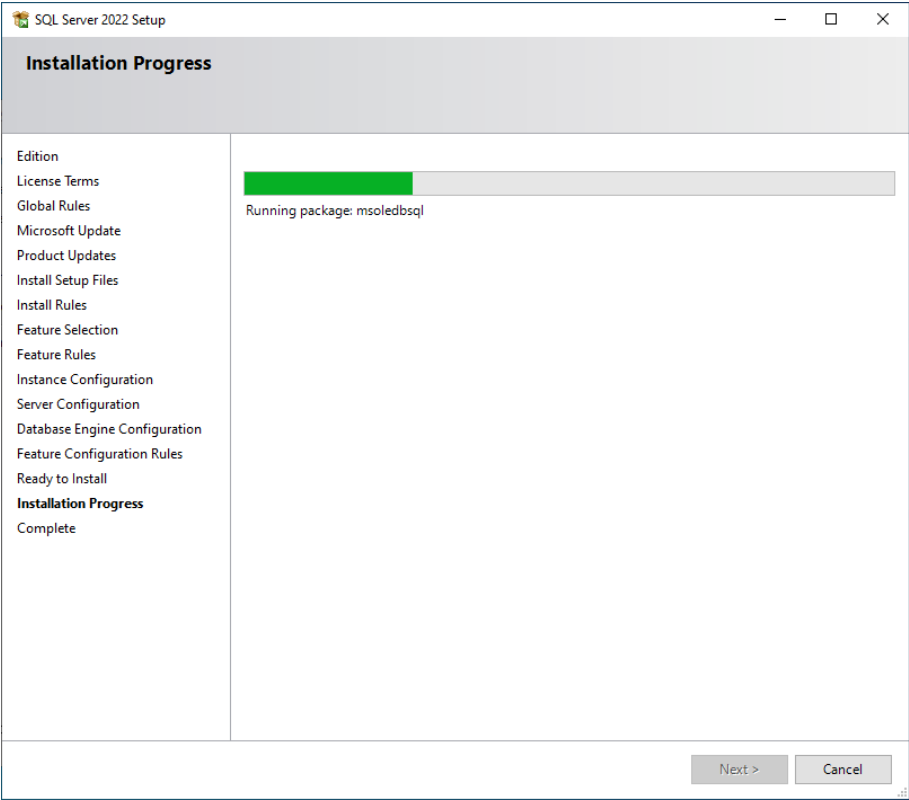


Figure 20: Installing SQL Server 2022 (XVI)

After installation, a pop-up window may inform about a required computer restart. The Complete page (see Figure 21) 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.

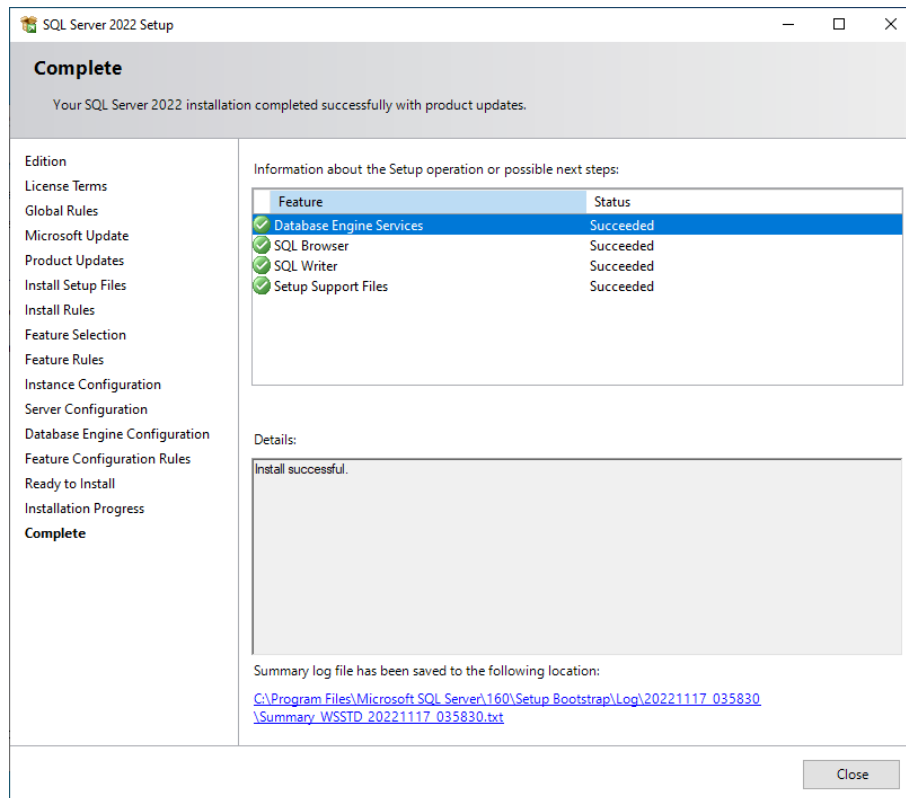


Figure 21: Installing SQL Server 2022 (XVII)

3.2.3 Installing the Documentation Contents

The documentation contents do not come with SQL Server 2022 but have to be downloaded separately from the Common Criteria website [WEB] as a ZIP file [AGD]. The contents can be viewed with Microsoft Help Viewer version 2.3 or later which is installed as part of Microsoft SQL Server Management Studio (SSMS) version 18.0 or later:

<https://docs.microsoft.com/sql/ssms/download-sql-server-management-studio-ssms>

Please note that a suitable version of Help Viewer is also delivered with Visual Studio Community version 2017 or later:

<https://visualstudio.microsoft.com/vs/community/>

To install and view the content, the following steps have to be performed:

1. Once SSMS or VisualStudio has been installed, go to help menu.
2. On help menu, set the help preference to "Launch in Help Viewer" (cf. Figure 22).

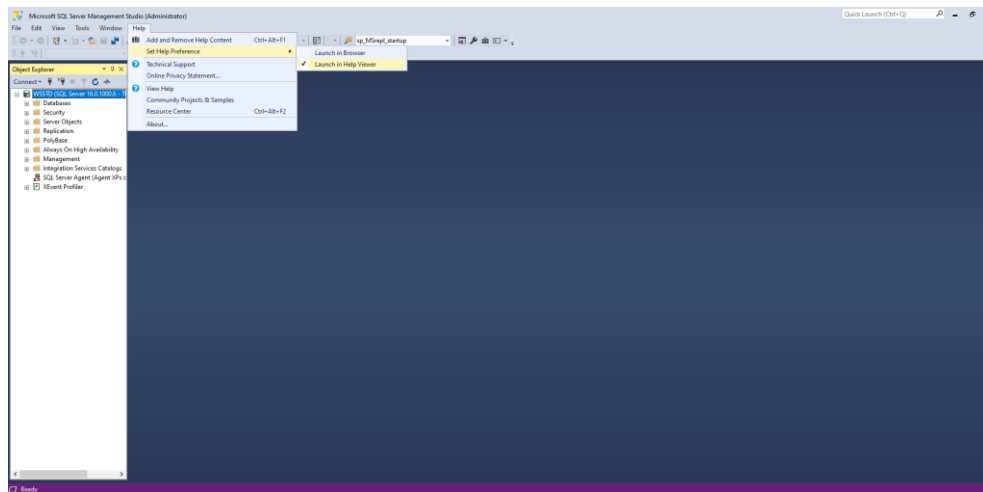


Figure 22: Setting Help Preference in SQL Server Management Studio

3. After that, go to “Help” “Add and Remove Help Content” and set the “Installation source” to “Disk” (cf. Figure 23). Now, click on the “...” and navigate to the directory where the ZIP file “Offline-Book_SQL-Server-2022_1.0_2022-12-23.zip” from [WEB] has been extracted (note that typing in the folder path directly might not work). Choose the file “book.msha” and confirm by clicking on “Open”.
4. “SQL Server 2022 Documentation” should now show up in the content list. Finally, click on “Add” in the “Action” column, and click on “Update” in the lower right corner. This will add SQL Server 2022 Documentation to the Help Viewer contents.

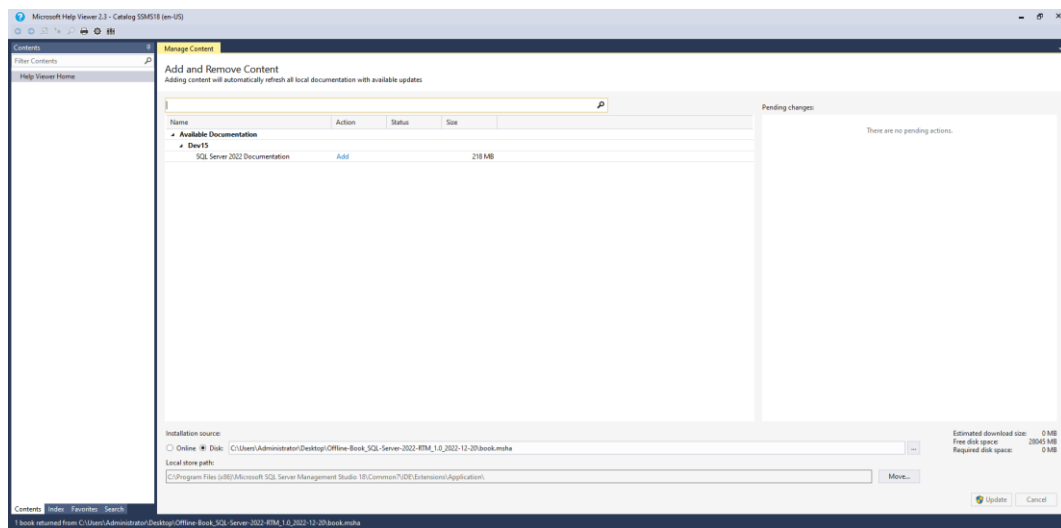


Figure 23: Choosing the installation source for the contents in Help Viewer

In this document all references to the Technical Documentation are given in form of [AGD, section: <Section Title>] and can be easily found in Help Viewer by typing or pasting the section title to the “Filter Contents” search box in the upper left corner of Help Viewer (cf. Figure 24).

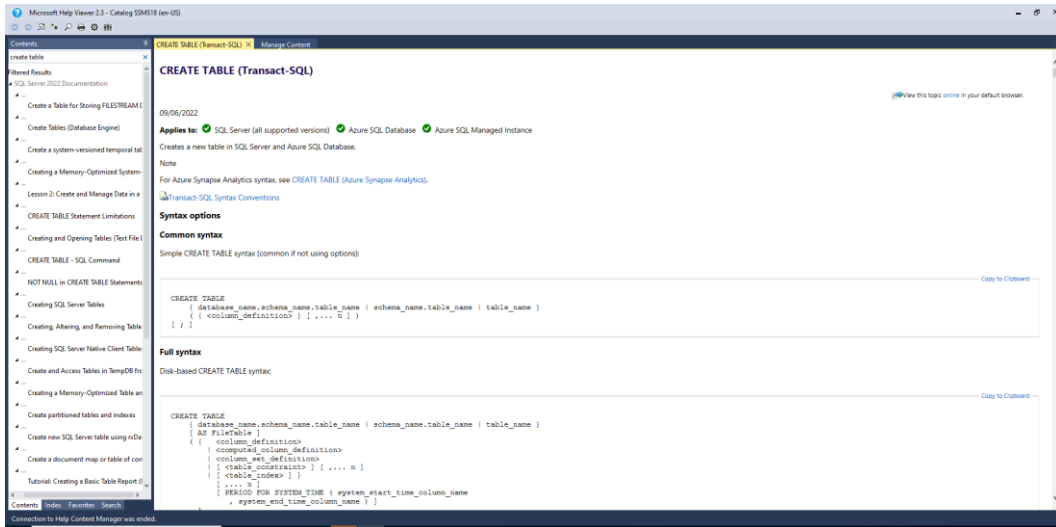


Figure 24: Retrieving relevant content from the Technical Documentation

There exist four graph boxes that can give some information about the search: *Content*, *Index*, *Favorites* and *Search*. It is remarkable that sometimes is necessary to search the name of the item on each of these graph boxes (the most representative and useful box for this task is **Search**, Figure 25) and open every tree graph to precisely find the object that is been looking for on the filter.

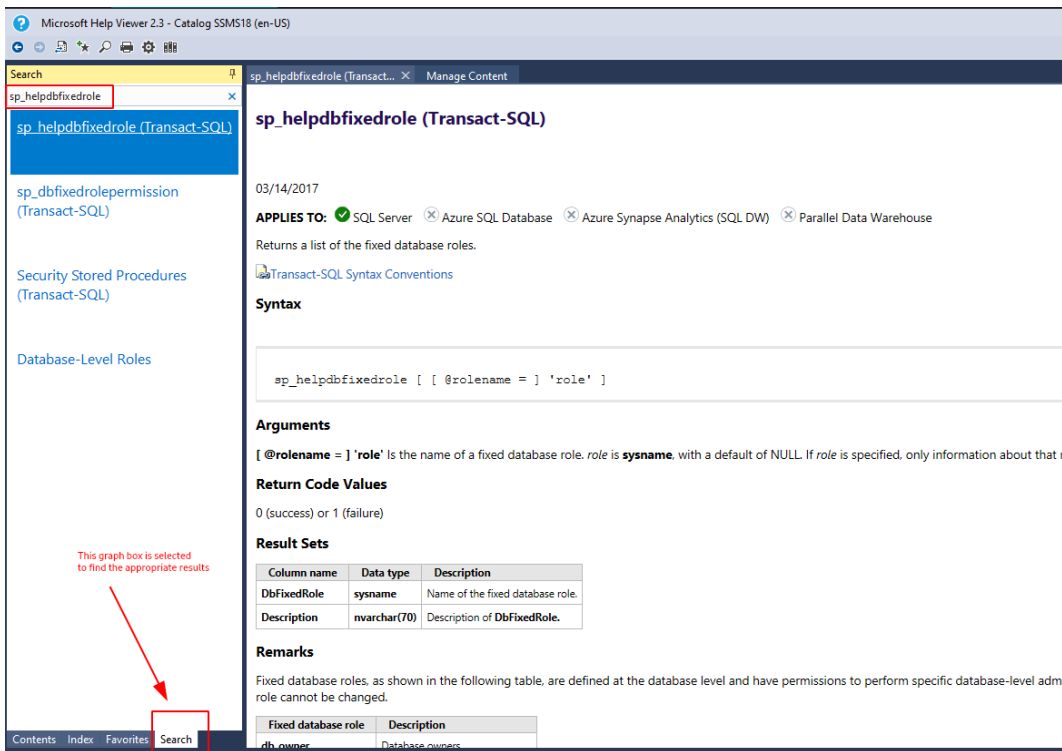


Figure 25: Search tab

An overview of the Technical Documentation contents is given in chapter 4.

Warning: To ensure that HelpViewer works properly, some extra steps are required.

Open Server Manager application. In the left menu window, go to the option *Local Server* as it is shown in picture. In the properties window, select *IE Enhanced Security Configuration* and a new window will open (Figure 26).

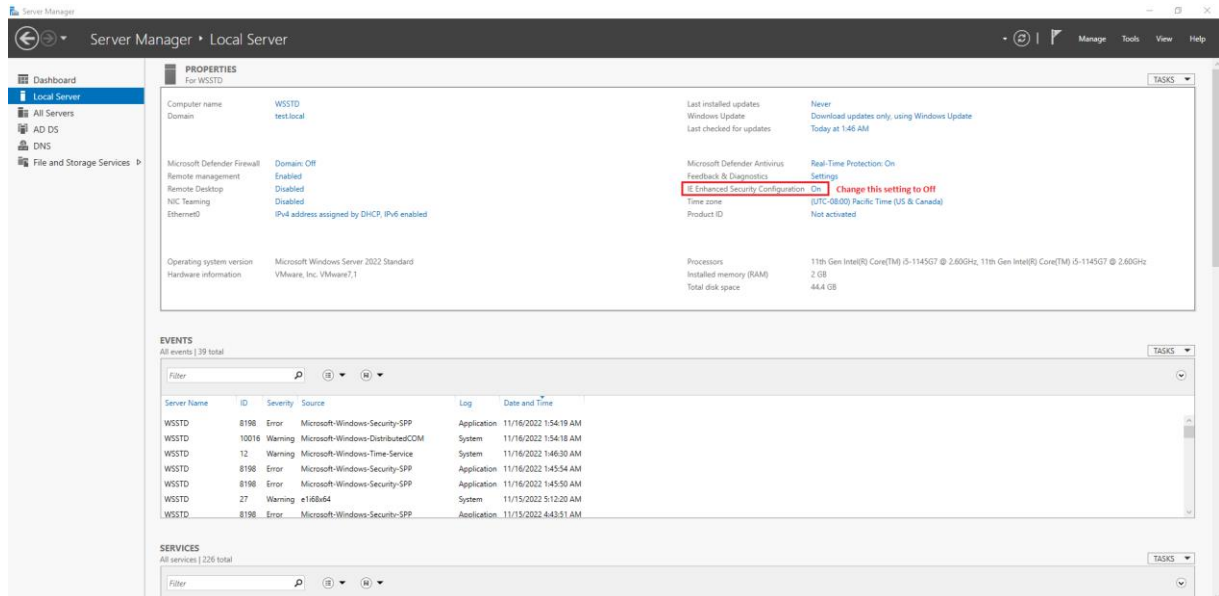


Figure 26: Local Server Settings

In this new window, set both options (users and administrators) to *OFF* and push OK button to apply the configuration changes (Figure 27).

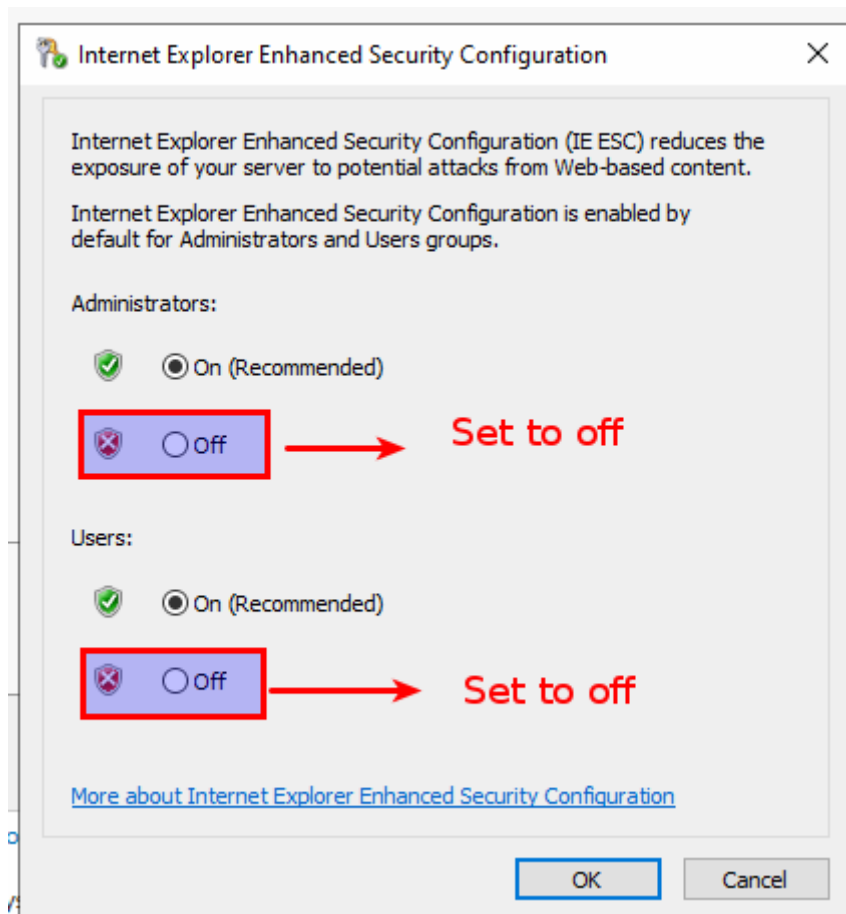


Figure 27: IE Enhanced Security Configuration

After perform these steps, HelpViewer tool should work correctly and the help menu can be consulted.

3.2.4 Installing the Cumulative Update

In order to install the Cumulative Update 3 please execute the CU3 update file (see section 3.2.1 for delivery details) on the machine on which the RTM version has been installed. In order to proceed, one needs to accept the license terms (Figure 28).

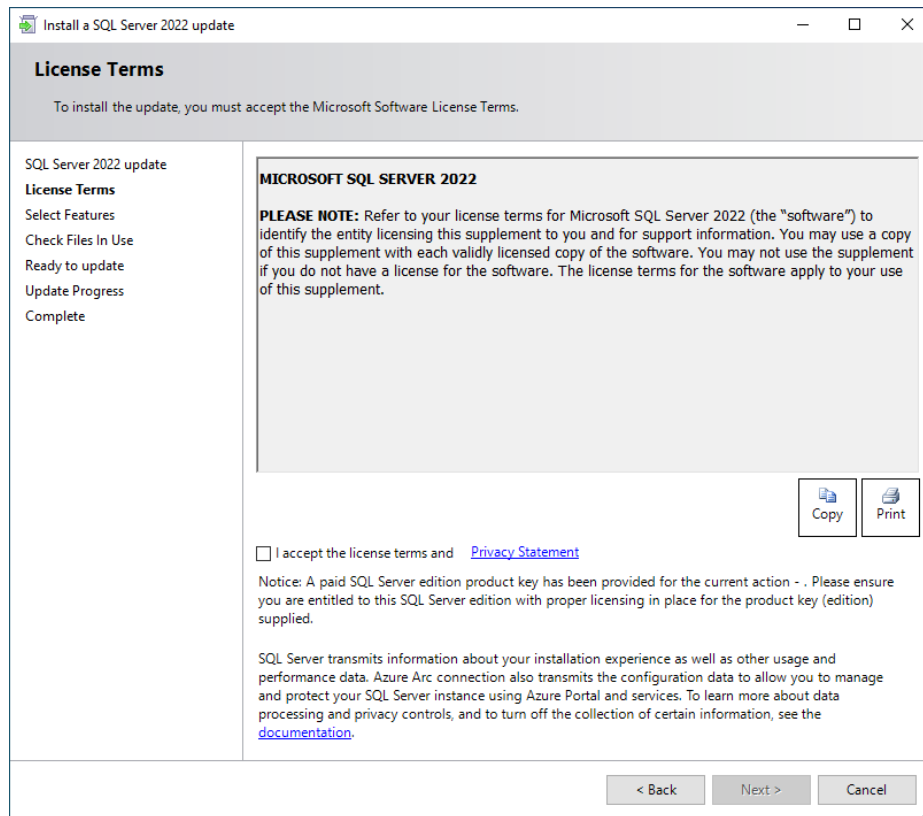


Figure 28: Updating SQL Server 2022 (I)

After that, the instances to update can be selected as can be seen in (Figure 29).

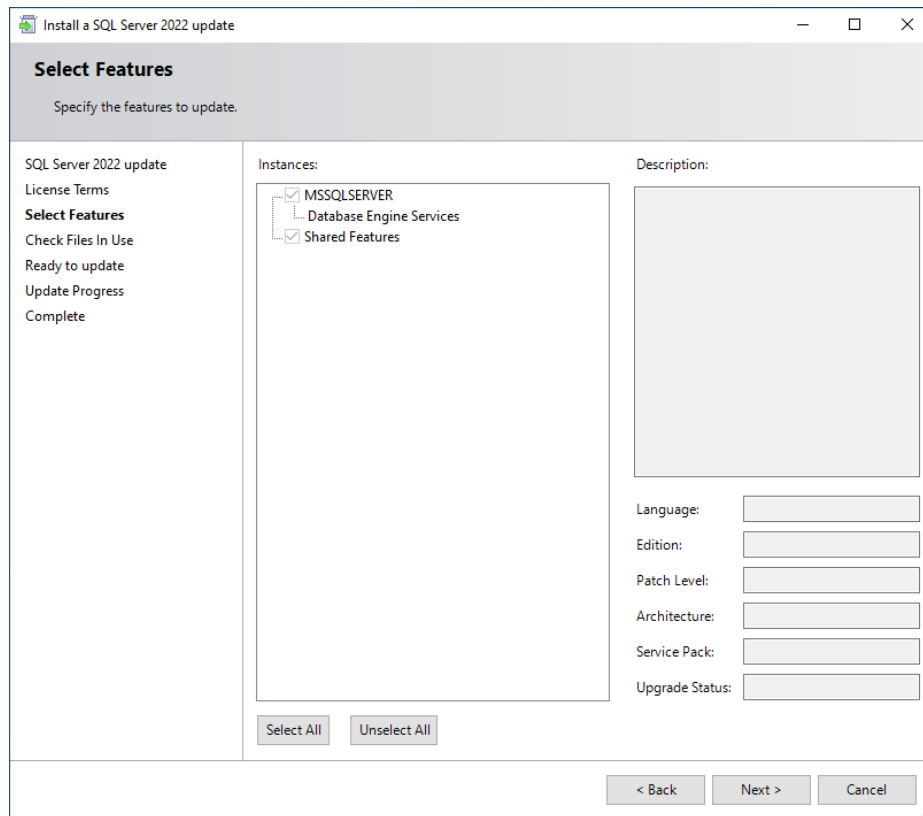


Figure 29: Updating SQL Server 2022 (II)

After the files in use are checked (Figure 30), the installer is ready to update (Figure 31). Please note that the patch level displayed in the “Ready to update” confirmation dialogue may differ from the one shown in the screenshot.

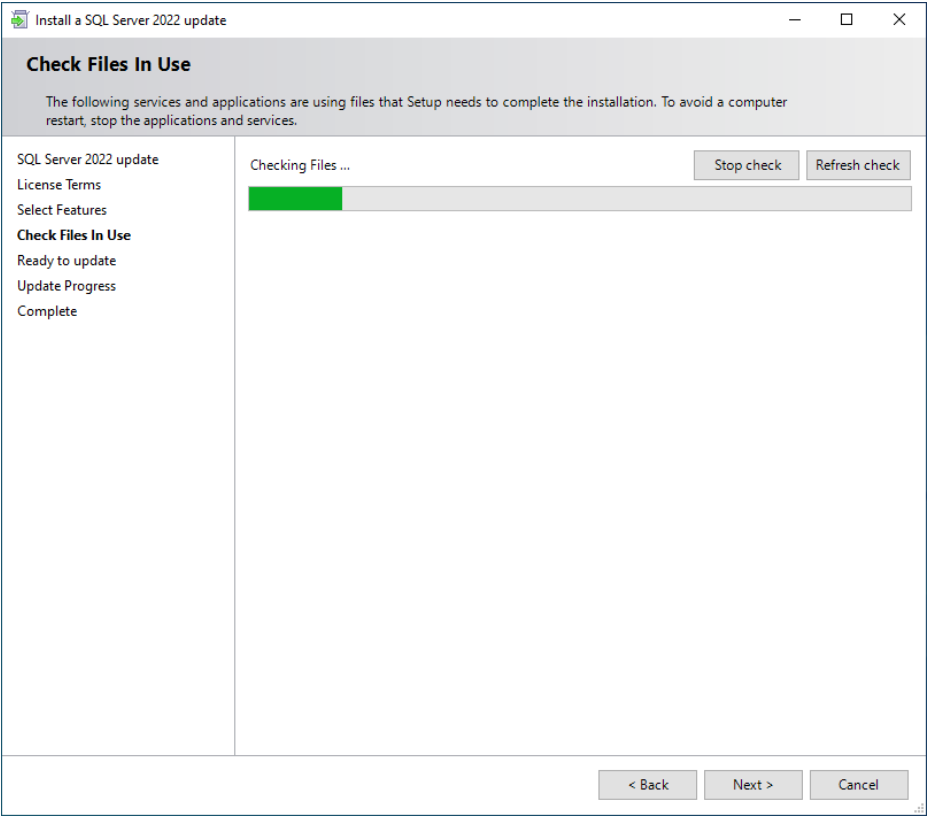


Figure 30: Updating SQL Server 2022 (III)

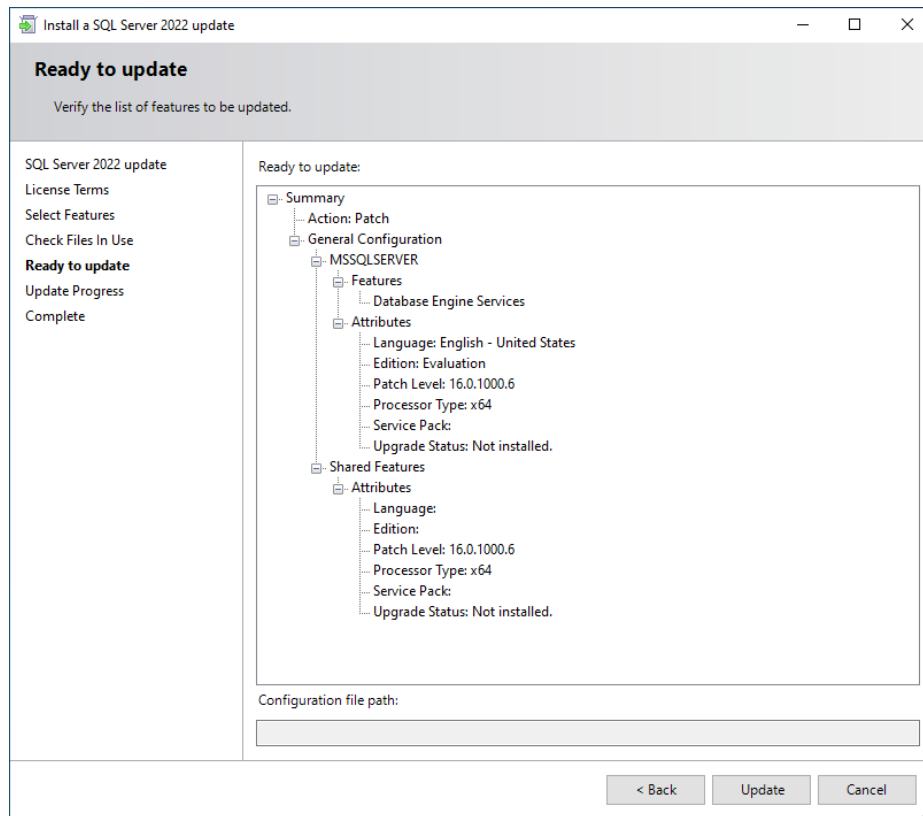


Figure 31: Updating SQL Server 2022 (IV)

The update progress is shown on the screen (Figure 32).

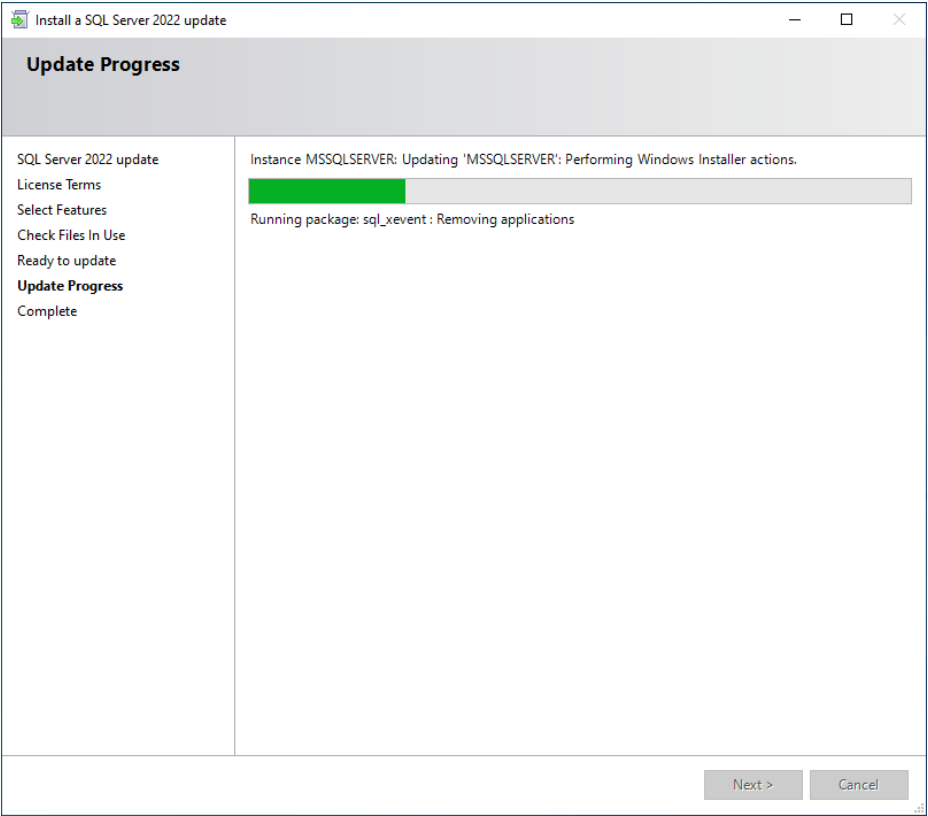


Figure 32: Updating SQL Server 2022 (V)

After the message, update has been completed, a status overview is shown (Figure 33).

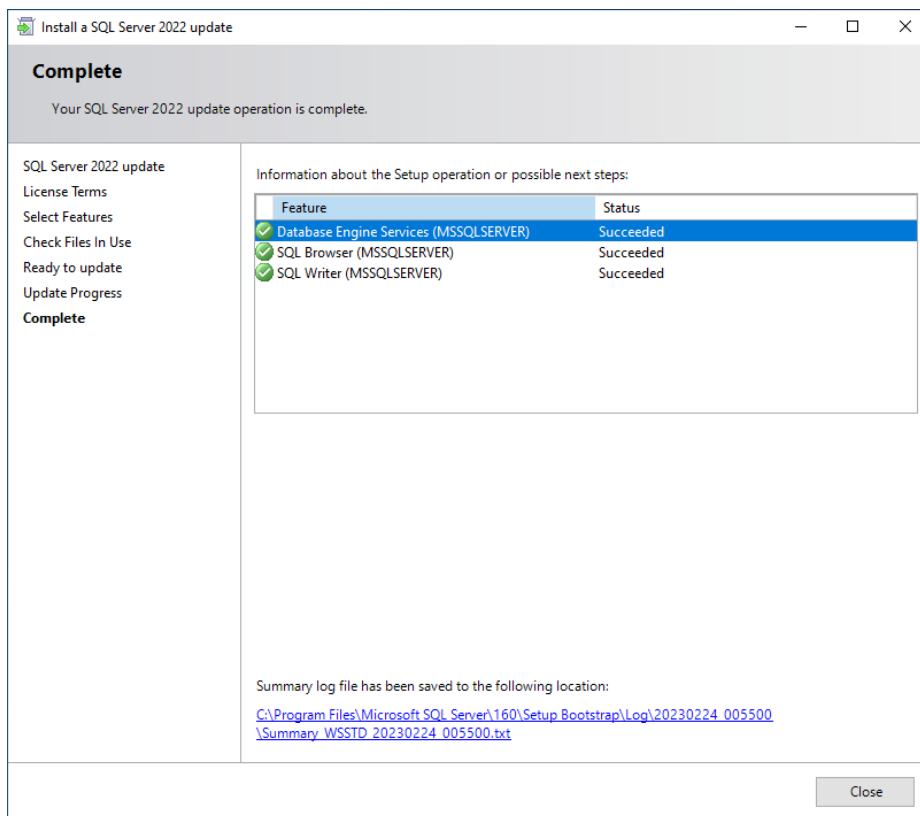


Figure 33: Complete installation of Cumulative Update

3.2.5 Querying the installed version

After the installation process has been finished the admin shall finally determine whether the correct version of SQL Server 2022 is installed. To do this the administrator of SQL Server shall 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 the string that is returned in response to this command shall start with

Microsoft SQL Server 2022 (RTM-CU3) - 16.0.4025.1 (X64)

These strings include information on the actual version of the SQL Server that has been installed (16.0.4025.1) and also show that the x64 edition has been installed.

3.2.6 Enabling the certified version

In the default installation of SQL Server 2022 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, section: “common criteria compliance enabled Server Configuration Option”].

For more information on “sp_configure” please see [AGD, section: “sp_configure (Transact-SQL)”].

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

3.2.7 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, section: “Logon Triggers”])

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 "SQL22_W_Install_cc_triggers_1.0_2022-12-20.sql" that can be obtained via [WEB] (click on "View our Common Criteria certification" and a PDF document will be downloaded. Within this PDF, click in the trigger link).

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 allowing 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 9.1.12)
- `dbo.sp_revoke_logon_denies` (see chapter 9.1.10)
- `dbo.sp_set_maximum_number_of_connections_per_login` (see chapter 9.1.11)
- `dbo.sp_remove_maximum_number_of_connections_limit` (see chapter 9.1.9)

3.2.8 Hardening execution permissions of xp_dirtree

Xp_dirtree is a stored procedure described in section 9.1.2. To restrict the execution of this stored procedure to administrators only, the following command shall be executed:

```
REVOKE Execute ON xp_dirtree FROM PUBLIC;
```

4 SQL Server Technical Documentation

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 utilizes a T-SQL client for interaction with the database engine of SQL Server 2022.
- An application using the database engine of SQL Server 2022.

SQL Server Technical Documentation ([AGD]) provides all kinds of users with the necessary information, how the database engine of SQL Server 2022 can be used.

The following links can be used as entry points into SQL Server Technical Documentation:

Topic	Reference
What's New (Database Engine)	[AGD, section: "What's new in SQL Server 2022"]
SQL Server Business Continuity	[AGD, section: "Business continuity and database recovery"]
Stored Procedure Reference	[AGD, section: "System Stored Procedures (Transact-SQL)"]
Transact-SQL Reference	[AGD, section: "Transact-SQL Reference (Database Engine)"]
XQuery Reference	[AGD, section: "XQuery Language Reference (SQL Server)"]

Table 3: Entry Points into SQL Server Technical Documentation

The following chapters are going to introduce the aspects for the secure administration and usage of SQL Server 2022, which are specific to the certified version. In order to ensure that the TOE is successfully installed and is working in a fully operational mode is mandatory to follow the instructions in section 7 and the requirements of the section 8 of this document.

5 Guidance Addendum

This chapter contains the guidance addendum for the secure administration and operation of the TOE. This document covers the aspects of guidance and operation, which are specific for the certified version of the database engine of SQL Server 2022. This document is a supplement of [AGD] documentation.

5.1 SQL Server start-up flags

The default configuration of the process of SQL Server database engine is running as a service under Windows Server 2022 and it is automatically started after the start of the Operating System. This configuration will be the used in the evaluated version.

Nonetheless in several situations it can be useful to start this program using the application "sqlservr.exe" directly employing a command prompt. To perform this action, it is important to start the SQL Server Configuration Manager and stops the automatic instance of the engine that is running.

Next, it is needed to start a command prompt process or PowerShell process in the main directory of SQL Server Engine that, by default, is "C:\Program files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Binn" and execute the "sqlserver.exe" application with an appropriate option, selecting a concrete mode of operation.

The subsequent table lists the available options that can be used with "sqlservr.exe" application that result in a certain mode of operation, cf. [AGD, section: Database Engine Service Startup Options]:

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.
-k DecimalNumber	This startup parameter limits the number of checkpoint I/O requests per second, where the DecimalNumber represents the checkpoint speed in MB per second. Changing this value can impact the speed of taking backups or going through the recovery process so proceed with caution.
-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 this feature shall not be used as a security feature because the provided client application name can be easily spoofed.
-m Client Application	Limits the connections to a specified client application, when you use the -m

Option	Description
Name	option with SQLCMD or SQL Server Management Studio. For example, -m SQLCMD limits connections to a single connection and that connection must identify itself as the SQLCMD client program. Use this option when you are starting SQL Server in single-user mode and an unknown client application is taking the only available connection. To connect through the Query Editor in Management Studio, use -m Microsoft SQL Server Management Studio - Query. Client Application Name is case sensitive. ** Security Note ** Do not use this option as a security feature. The client application provides the client application name, and can provide a false name as part of the connection string.
-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 2022. 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\bin 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, section: "DBCC TRACEON - Trace Flags (Transact-SQL)"]
-x	Disables monitoring features such as the keeping of CPU time and cache-hit ratio statistics. Allows maximum performance.
-d <i>master_path</i>	Indicates the fully qualified path for the master database. If you do not provide this option, the existing registry parameters are used. There are no spaces between -d and <i>master_path</i> .
-e <i>error_log_path</i>	Indicates the fully path for the error log file. If not specified, the default location is *(<Drive>)*\Program Files\Microsoft SQL Server\MSSQL\Log\Errorlog for the default instance and *(<Drive>)*\Program Files\Microsoft SQL Server\MSSQL\$*instance_name*\Log\Errorlog for a named instance. There are no spaces between -e and <i>error_log_path</i> .
-l <i>master_log_path</i>	Indicates the fully qualified path for the master database transaction log file. There are no spaces between -l and <i>master_log_path</i> .
-v	Displays the server version number.
--help	Shows the help application menu of sqlservr.exe application.

Table 4: Start-up 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 / -m Client Application Name: 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) this mode is not used within the certified configuration.
- -k: This parameter can modify the speed of taking backups or going through the recovery process. This can result in a security issue, so this parameter is not used on productive environment. Nevertheless, this parameter can solve bottlenecks caused by I/O requests by the checkpoint process.
- -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, section: "DBCC TRACEON - Trace Flags (Transact-SQL)"].

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
- -s: Simply starts a further instance of the engine. The instances will work independently and enforce all Security Functions.
- -x / -E: These modes 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].
- -l, -e and -d are log path configuration options that allow to modify the default paths of logs.

The application "sqlservr.exe" provides more options than listed before. These options do not represent different modes of operation. A complete overview of the options for "sqlservr.exe" can be found in [AGD, section: "sqlservr Application"].

Additional info: if the parameter --help is used, parameters (-a<L2, -B, -g number, -y number and -K), that are not defined in the AGD or the table, can be shown. These parameters are Microsoft Internal Purpose or Legacy parameters and are out of the certified evaluation.

As it is specified at the beginning of this chapter, the TOE runs, by default, as a service with the startup option "-s" and the instance name "MSSQLSERVER". Path options "-l", "-e" and "-d" are defined, by default, as Windows Registry Variable values with the following paths and no other modes will be allowed in the evaluated version:

- "C:\Program files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA" for options -l and -d.
- "C:\Program files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Log" for option -e.

Furthermore, errors of the TOE are managed via the error log (that is in the default path C:\Program files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Log\ERRORLOG). The only user(s)

that can manage these errors are the user(s) that belong(s) to the *sysadmin* role. A token is sent to the error log to identify the different type of the error, the severity and the actions to carry out when these errors occur. For further detail, please refer to [TDS-SPEC, section 2.2.7.10 Error].

In order to manage the TOE, it should be possible to use a friendly interface instead of a command prompt. For example, in order to manage the configuration part, software SQL 2022 Server Configuration Manager could be used and to manage the database SQL Server Management Studio could be used also. All these software programs and interfaces are out of scope of the evaluation and shall not be taken into account as TOE parts.

5.2 Administrator server management roles

The ST document defines different roles for the management of the TOE. These roles have different capabilities to manage the permissions of the server. SQL server provides nine different fixed roles that are defined in the table below:

Fixed server-level role	Description
sysadmin	Members of the sysadmin fixed server role can perform any activity in the server.
serveradmin	Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server.
securityadmin	Members of the securityadmin fixed server role manage login and their properties. They can GRANT , DENY , REVOKE server-level permissions. They can also GRANT , DENY , REVOKE database-level permissions if they have access to the database. Additionally, they can reset passwords for SQL Server logins. IMPORTANT: The ability grant access to the Database Engine and to configure user permissions allows the security admin to assign most server permissions. The securityadmin role should be treated as equivalent to the sysadmin role. As an alternative, consider using the new fixed server role ##MS_LoginManager## defined below.
processadmin	Members of the processadmin fixed server role can end processes that are running in an instance of SQL server.
setupadmin	Members of the setupadmin fixed server role can add and remove linked servers by using Transact-SQL statements. (sysadmin membership is needed when using Management Studio).
bulkadmin	Members of bulkadmin fixed server role can run the BULK INSERT statement.
diskadmin	The diskadmin fixed server role is used for managing disk files.
dbcreator	Members of the dbcreator fixed server role can create, alter, drop and restore any database
public	Every SQL Server login belongs to the public server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to

Fixed server-level role	Description
	public on that object. Only assign public permissions on any object when you want the object to be available to all users. You cannot change membership in public. Note: public is implemented differently than other role, and permissions can be granted, denied or revoked from the public fixed server roles.

Table 5: Administrator server roles

The permissions that are granted to the fixed server roles (except public role) cannot be changed. It is possible to create user-defined server roles and add server-level permissions to the user-defined server roles.

SQL Server 2022 introduces ten additional server roles that have been designed specifically with the *Principle of Least Privileges* in mind, which have the prefix **##MS_** and the suffix **##** to distinguish them from other regular user-created principals and custom server roles. These new roles contain privileges that apply on server scope but also can inherit down to individual databases. Descriptions for each of these roles are provided in the table below:

Fixed server-level role	Description
##MS_DatabaseConnector##	Members of the ##MS_DatabaseConnector## fixed server role can connect to any database without requiring a User-account in the database to connect to. To deny the CONNECT permission to a specific database, users can create a matching user account for this login in the database and then DENY the CONNECT permission to the database-user. This DENY permission will overrule the GRANT CONNECT permission coming from this role.
##MS_LoginManager##	Members of the ##MS_LoginManager## fixed server role can create, delete and modify logins. Contrary to the fixed server role securityadmin, this role does not allow members to GRANT privileges. It is a more limited role that helps to comply with the <i>Principle of least Privilege</i>.
##MS_DatabaseManager##	Members of the ##MS_DatabaseManager## fixed server role can create and delete databases. A member of the ##MS_DatabaseManager## role that creates a database, becomes the owner of that database, which allows that user to connect to that database as the dbo user. The dbo user has all database permissions in the database. Members of the ##MS_DatabaseManager## role don't necessarily have permission to access databases that they don't own.
##MS_ServerStateManager##	Members of the ##MS_ServerStateManager## fixed server role have the same permissions as

Fixed server-level role	Description
	the ##MS_ServerStateReader## role. Also, it holds the ALTER SERVER STATE permission, which allows access to several management operations, such as: DBCC FREEPROCCACHE, DBCC FREESYSTEMCACHE ('ALL'), DBCC SQLPERF()
##MS_ServerStateReader##	Members of the ##MS_ServerStateReader## fixed server role can read all dynamic management views (DMVs) and functions that are covered by VIEW SERVER STATE , and respectively has VIEW DATABASE STATE permission on any database on which the member of this role has a user account.
##MS_ServerPerformanceStateReader##	Members of the ##MS_ServerPerformanceStateReader## fixed server role can read all dynamic management views (DMVs) and functions that are covered by VIEW SERVER PERFORMANCE STATE , and respectively has VIEW DATABASE PERFORMANCE STATE permission on any database on which the member of this role has a user account. This is a subset of what the ##MS_ServerStateReader## server role has access to which helps to comply with the <i>Principle of least Privilege</i> .
##MS_ServerSecurityStateReader##	Members of the ##MS_ServerSecurityStateReader## fixed server role can read all dynamic management views (DMVs) and functions that are covered by VIEW SERVER SECURITY STATE , and respectively has VIEW DATABASE SECURITY STATE permission on any database on which the member of this role has a user account. This is a small subset of what the ##MS_ServerStateReader## server role has access to, which helps to comply with the <i>Principle of least Privilege</i> .
##MS_DefinitionReader##	Members of the ##MS_DefinitionReader## fixed server role can read all catalog views that are covered by VIEW ANY DEFINITION , and respectively has VIEW DEFINITION permission on any database on which the member of this role has a user account.
##MS_PerformanceDefinitionReader##	Members of the ##MS_PerformanceDefinitionReader## fixed server role can read all catalog views that are covered by VIEW ANY PERFORMANCE DEFINITION , and respectively has VIEW PERFORMANCE DEFINITION permission on any database on which the member of this role has a user account. This is a subset of what the ##MS_DefinitionReader## server role has access to.
##MS_SecurityDefinitionReader##	Members of the ##MS_SecurityDefinitionReader## fixed server role can read all catalog views that are covered by VIEW ANY SECURITY DEFINITION , and respectively has VIEW SECURITY DEFINITION permission on any database on which the member of this role has a user account. This is a small subset of what

Fixed server-level role	Description
	the ##MS_DefinitionReader## server role has access to which helps to comply with the <i>Principle of least Privilege</i> .

Table 6: Fixed server-roles introduced in SQL Server 2022

A more detailed description about roles can be found in document [AGD, section: Server-Level Roles]. Further information regarding the functionality of the server roles can be found in the section 5.4 and 5.6.

In order to check user server-level permissions can be used the query:

SELECT * FROM sys.fn_builtin_permissions('SERVER') ORDER BY permission_name;

More information about manage server permissions can be found in document [AGD, section: Permissions (Database Engine)] and referring to the function to check the permissions in document [AGD, section: "sys.fn_builtin_permissions (Transact-SQL)"].

The management functionality of the TOE is defined in the ST document and it can be executed via T-SQL commands or Stored procedures, which are called using a T-SQL command.

A T-SQL client could be used to provide some help with this procedure. As an example SQL Server Management tools, that comprise a T-SQL client with a comfortable GUI that can be used for administration but any other client could be useful for this task.

However, it is important to remark that this GUI is not a part of the configured evaluation, so it is out of scope.

To control the services associated with SQL Server, such as configure network protocols used by SQL Server and manage network connectivity configuration from SQL Server clients, SQL Server Configuration Manager tool can be used. The settings are stored and changed into 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 2022, to view service properties, or to change service properties.
- SQL Server 2022 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, section: "SQL Server Configuration Manager"].

More detailed information about the functionality which is provided by the SQL Server Configuration Manager can be found in [AGD, section: "SQL Server Configuration Manager"].

5.3 User database management roles

The ST document defines different roles for the management functions of the database. These roles have different capabilities to manage the permissions in the database. These roles are comparable to groups in Windows OS. Database-level roles are database-wide in their permissions scope. There exist two types of permissions: *fixed-database roles* and *user-defined roles* that can be created by the administrator of the TOE. The first ones are defined in the following table:

Fixed-database role name	Description
db_owner	Members of the db_owner fixed database role can perform all configuration and maintenance activities on the database and can also drop the database in SQL Server.
db_securityadmin	Members of the db_securityadmin fixed database role can modify role membership for custom roles only and manage permissions. Members of this role can potentially elevate their privileges and their actions should be monitored.
db_accessadmin	Members of the db_accessadmin fixed database role can add or remove access to the database for Windows logins, Windows groups and SQL Server logins.
db_backupoperator	Members of the db_backupoperator fixed database role can back up the database.
db_ddladmin	Members of the db_ddladmin fixed database role can run any Data Definition Language (DDL) command in a database.
db_datawriter	Members of the db_datawriter fixed database role can add, delete or change data in all user tables.
db_datareader	Members of the db_datareader fixed database role can read all data from all user tables and views.
db_denydatawriter	Members of the db_denydatawriter fixed database role cannot add, modify or delete any data in the user tables within a database.
db_denydatareader	Members of the db_denydatareader fixed database role cannot read any data from the user tables and views within a database.

Table 7: Database roles

It is not possible to change the permissions on fixed-database roles. The permissions of user-defined database roles can be customized by using GRANT, DENY and REVOKE statements. For more information about permissions on database user-defined roles, see document [AGD: section Permissions (Database Engine)].

Every user can connect to the TOE via the external interface EV-CLIENT using a T-SQL client.

The SQL Server Management Studio which ships separately from the TOE comprises a T-SQL client, which can be used. 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, section: "Transact-SQL Reference"].

5.4 Relevant functions for administrator server roles

The following chapters (from 5.4.1 to 5.4.9) describe the different administrator server roles, their associated permissions and examples with commands that can be used. The permissions associated to each user can be configured with the commands of the following list. Also, the commands of this list can

be used to create new roles with their associated permissions. Please refer to [AGD, section: “Server-Level Roles”] for a detailed explanation:

Feature	[AGD] Reference
sp_helpsrvrole (Transact-SQL)	[AGD, section: “sp_helpsrvrole (Transact-SQL)”]
sp_helpsrvrolemember (Transact-SQL)	[AGD, section: “sp_helpsrvrolemember (Transact-SQL)”]
sp_srvrolepermission (Transact-SQL)	[AGD, section: “sp_srvrolepermission (Transact-SQL)”]
IS_SRVROLEMEMBER (Transact-SQL)	[AGD, section: “IS_SRVROLEMEMBER (Transact-SQL)”]
sys.server_role_members (Transact-SQL)	[AGD, section: “sys.server_role_members (Transact-SQL)”]
sp_addsrvrolemembers (Transact-SQL)	[AGD, section: “sp_addsrvrolemembers (Transact-SQL)”]
CREATE SERVER ROLE (Transact-SQL)	[AGD, section: “CREATE SERVER ROLE (Transact-SQL)”]
ALTER SERVER ROLE (Transact-SQL)	[AGD, section: “ALTER SERVER ROLE (Transact-SQL)”]
DROP SERVER ROLE (Transact-SQL)	[AGD, section: “DROP SERVER ROLE (Transact-SQL)”]
sp_dropsrvrolemember (Transact-SQL)	[AGD, section: “sp_dropsrvrolemember (Transact-SQL)”]

Table 8: Relevant functions for Administrator server roles

5.4.1 Sysadmin role

The Sysadmin server fixed role has all permissions in the server. This role can create users, groups or databases, change user’s permissions, manage security functions like audit trails, define modes of authentication, control different databases; among other things. This role can manage all the security events of the TOE, including audit trails.

Feature/permission	Purpose	Command Example
CONTROL SERVER	Control any server option.	Any command of the server.

Table 9: Sysadmin permissions

5.4.2 Serveradmin role

The Serveradmin fixed role has top level permissions that can allow to control servers, such as alter resources or shutdown the server. This role can check and fix all the security events related to integrity of the data of the server or the resources. Functions that can perform are described in the following tasks:

Feature/permission	Purpose	Command Example
ALTER ANY ENDPOINT	Alter any endpoint.	ALTER/CREATE/DROP ENDPOINT,...
ALTER SETTINGS	Alter server settings.	sp_configure, RECONFIGURE,...
CREATE ENDPOINT	Create any endpoint	CREATE ENDPOINT
SHUTDOWN	Shutdown the server.	SHUTDOWN
ALTER SERVER STATE	Alter server state.	DBCC FREE, CACHE, SQLPERF,...

Feature/permission	Purpose	Command Example
VIEW SERVER STATE	View server state.	SELECT on server-level DMV's

Table 10: Serveradmin permissions

5.4.3 Securityadmin role

The securityadmin fixed role can create users or manage users that exist and manage permissions of different users. This role can check and fix all the security events related to users (drop old users, create new users, assign specific roles). Functions of this role are defined on the next table:

Feature/permission	Purpose	Command Example
ALTER ANY LOGIN	Alter any parameter of a login.	ALTER_LOGIN, sp_addlinkedserverlogin, DROP LOGIN, CREATE LOGIN

Table 11: Securityadmin permissions

5.4.4 Processadmin role

The Processadmin fixed role can manage the connections and the state of the server. This role can check and fix all the security events related to server resources (out of memory, limit in management of resources, etc). Functions of this role are described in the following table:

Feature/permission	Purpose	Command Example
ALTER ANY CONNECTION	Modify/suppress server connections.	KILL
ALTER SERVER STATE	Alter server state.	DBCC FREE, CACHE, SQLPERF,...
VIEW SERVER STATE	View server state.	SELECT on server-level DMV's

Table 12: Processadmin permissions

5.4.5 Setupadmin role

The Setupadmin fixed role can add or eliminate linked servers with T-SQL commands. This role has no specific security events on its charge.

Warning: In the case that the Management Studio use, this feature is only available for sysadmin fixed role.

Feature/permission	Purpose	Command Example
ALTER ANY LINKED SERVER	Add/suppress linked servers.	sp_addlinkedserver()

Table 13: Setupadmin permissions

5.4.6 Bulkadmin role

The Bulkadmin fixed role can perform bulk operations. This role has no specific security events on its charge.

Feature/permission	Purpose	Command Example
ADMINISTER BULK OPERATIONS	Copy data in different parts.	BULK INSERT, OPENROWSET(BULK,..)

Table 14: Bulkadmin permissions

5.4.7 Diskadmin role

The Diskadmin fixed role can perform operations related to disk files of the TOE. This role can check and fix all the security events related to the disk files.

Feature/permission	Purpose	Command Example
ALTER RESOURCES	Alter any server disk file resource.	ADD FILE...

Table 15: Diskadmin permissions

5.4.8 Dbcreator role

The Dbcreator fixed role can perform operations related to the management of databases: create, drop, modify or access. . This role has no specific security events on its charge.

Feature/permission	Purpose	Command Example
ALTER ANY DATABASE	Alter parameters and data from the database.	ALTER/DROP DATABASE
CREATE ANY DATABASE	Create any database.	CREATE DATABASE

Table 16: Dbcreator permissions

5.4.9 Public role

The public fixed role is the default role in a new handling SQL session. Unless a server security entity has not been granted or denied specific security permissions, the user inherits the permissions granted to public role (The default permissions of this class can be fixed by the administrator of the TOE). This class do not have inherent permissions but there are some server permissions by default (These default permissions can be revoked by an administrator). This role is not an administrator, so this role has no access to any security relevant event of the TOE.

Feature/permission	Purpose	Command Example
VIEW ANY DATABASE	View parameters and data from database.	SELECT * ON DATABASE
CONNECT ON ENDPOINT	Connect on an endpoint.	CONNECT ON ENDPOINT

Table 17: Public role permissions

5.5 Relevant functions for database management roles

The following chapters (from 5.5.1 to 5.5.9) describe the different database user roles, their associated permissions and examples with commands that can be used. The permissions associated to each user can be configured with the commands of the following list. Also, the commands of this list can be used to create new roles with their associated permissions. These database roles has no security events on its charge. Please refer to [AGD, section: "Database-Level Roles"] for a detailed explanation:

Feature	[AGD] Reference
sp_dbfixedrolepermission(Transact-SQL)	[AGD, section: "ssp_dbfixedrolepermission(Transact-SQL)"]
sp_helprole(Transact-SQL)	[AGD, section: "sp_helprole(Transact-SQL)"]
sp_helprolemember(Transact-SQL)	[AGD, section: "sp_helprolemember(Transact-SQL)"]
sys.database_role_members (Transact-SQL)	[AGD, section: "sys.database_role_members (Transact-SQL)"]
IS_MEMBER (Transact-SQL)	[AGD, section: "IS_MEMBER (Transact-SQL)"]
CREATE ROLE (Transact-SQL)	[AGD, section: "CREATE ROLE (Transact-SQL)"]
ALTER ROLE (Transact-SQL)	[AGD, section: "ALTER ROLE (Transact-SQL)"]
DROP ROLE (Transact-SQL)	[AGD, section: "DROP ROLE (Transact-SQL)"]
GRANT	[AGD, section: "GRANT"]
DENY	[AGD, section: "DENY"]
REVOKE	[AGD, section: "REVOKE"]

Table 18: Relevant functions for database roles

5.5.1 Db_owner role

The fixed database role Db_owner is the owner of the database and can perform all configuration and maintenance activity related to the database.

Feature/permission	Purpose	Command Example
CONTROL ON DATABASE	Control any operation on database	Any command related to a database

Warning: some operations and activities require different server permissions, so a db_owner cannot perform them.

5.5.2 Db_securityadmin role

The fixed database role Db_securityadmin can modify role membership for custom roles and manage permissions. The functions of this role are described in the following table:

Feature/permission	Purpose	Command Example
ALTER ANY SCHEMA	Alter any schema on a database.	ALTER/DROP SCHEMA

Feature/permission	Purpose	Command Example
CREATE SCHEMA	Create schema on a database. Also create tables and views within this schema.	CREATE SCHEMA
ALTER ANY ROLE, CREATE ROLE	Modify parameters related to roles.	ALTER ON ROLE:: <name>, CREATE ROLE, DROP ROLE</name>
ALTER ANY APPLICATION ROLE	Modify parameters on application roles.	ALTER ON APPLICATION ROLE<name>,ALTER/DROP/CREA TE APPLICATION ROLE
VIEW DEFINITION	View any definition.	VIEW DEFINITION ON DATABASE:: <name>,view </name>,view DEFINITION ON APPLICATION ROLE:: <name>< td=""></name><>

Table 19: Db_securityadmin permissions

Warning: members of this role can potentially elevate their privileges and their actions should be monitored.

5.5.3 Db_accessadmin role

The fixed database role Db_accessadmin can control the access of the database of Windows logins, Windows groups and SQL server logins. The functions of this role are described in the following table:

Feature/permission	Purpose	Command Example
CREATE SCHEMA	Create schema on a database. Also create tables and views within this schema.	CREATE SCHEMA
ALTER ANY USER	Modify some user parameters.	ALTER USER:: <name>< td=""></name><>
CONNECT ON DATABASE	Control the connections to the database.	GRANT/DENY/REVOKE CONNECT

Table 20: Db_accessadmin permissions

5.5.4 Db_backupoperator role

The fixed database role Db_backupoperator can manage backup processes of the database. The following table contains the functions that it can manage:

Feature/permission	Purpose	Command Example
BACKUP DATABASE	Control backup copies of the database.	BACKUP DATABASE
BACKUP LOG	Control backup copies of the log of the database.	BACKUP LOG
CHECKPOINT	Generate a manual checkpoint in a connected database.	CHECKPOINT

Table 21: *Db_backupoperator* permissions**5.5.5 Db_ddladmin role**

The fixed database role Db_ddladmin can run any command related to Data Definition Language (DDL) in a database. The following table contains the functions that it can manage:

Feature/permission	Purpose	Command Example
ALTER ANY ASSEMBLY	Alter an assembly modifying SQL Server catalog properties.	ALTER/DROP/CREATE ASSEMBLY
ALTER ANY ASYMMETRIC KEY	Change the properties of an asymmetric key.	ALTER/DROP/CREATE ASYMMETRIC KEY
ALTER ANY CERTIFICATE	Change the password used to encrypt private key of a certificate or import a new one.	ALTER/DROP/CREATE CERTIFICATE
ALTER ANY CONTRACT	Create/Drop contract, that is a message type used in Service Broker conversation.	CREATE/DROP CONTRACT
ALTER ANY DATABASE DDL TRIGGER	Modify the definition of a DDL trigger.	ALTER/DROP/CREATE database triggers
ALTER ANY DATABASE EVENT NOTIFICATION	Alter any database scoped notification.	CREATE EVENT NOTIFICATION
ALTER ANY DATASPACE	Alter partition and guide statements.	ALTER ANY DATASPACE TO
ALTER ANY FULLTEXT CATALOG	Change the properties of a fulltext catalog.	ALTER/DROP/CREATE FULLTEXT CATALOG
ALTER ANY MESSAGE TYPE	Change the properties of a message type.	ALTER/DROP/CREATE MESSAGE TYPE
ALTER ANY REMOTE SERVICE BINDING	Change the properties of a remote service binding.	ALTER/DROP/CREATE SERVICE BINDING
ALTER ANY ROUTE	Change the information of an existing route in SQL Server.	ALTER/DROP/CREATE ROUTE
ALTER ANY SCHEMA	Transfer a securable between schemas.	CREATE SCHEMA, ALTER ON SCHEMA::<name>
ALTER ANY SERVICE	Change an existing service.	ALTER/DROP/CREATE SERVICE
ALTER ANY SYMMETRIC KEY	Changes the properties of a symmetric key	ALTER/DROP/CREATE SYMMETRIC KEY
ALTER ANY EXTERNAL LIBRARY	Changes an existing external library.	ALTER/DROP/CREATE EXTERNAL LIBRARY

Feature/permission	Purpose	Command Example
CHECKPOINT	Generate a manual checkpoint in a connected database.	CHECKPOINT
CREATE AGGREGATE	Create a user-defined aggregate function whose implementation is defined in a class of assembly.	CREATE AGGREGATE
CREATE DEFAULT	Create an object called a default.	CREATE DEFAULT
CREATE FUNCTION	Create a user-defined function on SQL Server.	CREATE FUNCTION
CREATE PROCEDURE	Create a T-SQL or CLR stored procedure in SQL Server.	CREATE PROCEDURE
CREATE QUEUE	Create a new queue in a database.	CREATE QUEUE
CREATE RULE	Create an object called rule.	CREATE RULE
CREATE SYNONYM	Create a new synonym.	CREATE SYNONYM
CREATE TABLE	Create a new table in SQL Server.	CREATE TABLE
CREATE TYPE	Create an alias data type or a user-defined type in the current database in SQL-Server.	CREATE TYPE
CREATE VIEW	Create a virtual table whose contents are defined in a query.	CREATE/ALTER/DROP VIEW
CREATE XML SCHEMA COLLECTION	Import the schema components into a database.	CREATE XML SCHEMA COLLECTION
REFERENCES	Apply to subordinate objects in database	sys.dm_sql_referencing_entities, sys.dm_sql_referenced_entities

Table 22: Db_ddladmin permissions

Warning: some commands require server level permissions and only 'sysaqdmin' members can perform them. Some of these commands are CREATE/ALTER/ACCESS/UNSAFE ASSEMBLY, CREATE/ALTER EVENT NOTIFICATION, CREATE/ALTER EXTERNAL LIBRARY and CREATE/ALTER AGGREGATE. Other statements in the table may require different fixed server roles membership. For further information, go to document [AGD, section: "Get started with Database Engine permissions"].

5.5.6 Db_datawriter role

The fixed database role Db_datawriter can add, delete or modify data in the user tables within a database. The following table contains the functions that it can manage:

Feature/permission	Purpose	Command Example
GRANT INSERT ON DATABASE	Allow insert information in a database.	GRANT INSERT ON DATABASE:: <name>< td=""></name><>
GRANT UPDATE ON DATABASE	Allow update information on a database.	GRANT UPDATE ON DATABASE:: <name>< td=""></name><>
GRANT DELETE ON DATABASE	Allow delete information on a database.	GRANT DELETE ON DATABASE:: <name>< td=""></name><>

Table 23: Db_datawriter permissions

5.5.7 Db_datareader role

The fixed database role Db_datareader can read all data from all user tables. The following table contains the functions that it can manage:

Feature/permission	Purpose	Command Example
GRANT SELECT ON DATABASE	Allow select information within a database.	GRANT SELECT ON DATABASE:: <name>< td=""></name><>

Table 24: Db_datareader permissions

5.5.8 Db_denydatawriter role

The fixed database role Db_denydatawriter cannot add, delete or modify any data in the user tables within a database. The following table contains the functions that it can manage:

Feature/permission	Purpose	Command Example
DENY INSERT ON DATABASE	Deny insert information in a database.	DENY INSERT ON DATABASE:: <name>< td=""></name><>
DENY UPDATE ON DATABASE	Deny update information in a database.	DENY UPDATE ON DATABASE:: <name>< td=""></name><>
DENY DELETE ON DATABASE	Deny delete information in a database.	DENY DELETE ON DATABASE:: <name>< td=""></name><>

Table 25: Db_denydatawriter permissions

5.5.9 Db_denydatareader role

The fixed database role Db_denydatareader cannot read any data in the user tables within a database. The following table contains the functions that it can manage:

Feature/permission	Purpose	Command Example
DENY SELECT ON DATABASE	Deny select information in a database.	DENY SELECT ON DATABASE:: <name>< td=""></name><>

Table 26: Db_denydatareader permissions

5.6 Relevant functions for fixed server-roles (##MS_) introduced in SQL Server 2022

As described in section 5.2, SQL Server 2022 introduces ten additional server roles that have been designed specifically with the *Principle of Least Privileges* in mind, which have the prefix ##MS_ and the suffix ## to distinguish them from other regular user-created principals and custom server roles.

The relationship between these new roles and the server-level and database-level permissions assigned is as follows.

Fixed server-level role	Server-level permission	Database-level permissions
##MS_DatabaseConnector##	CONNECT ANY DATABASE	CONNECT
##MS_LoginManager##	CREATE LOGIN ALTER ANY LOGIN	N/A
##MS_DatabaseManager##	CREATE ANY DATABASE ALTER ANY DATABASE	ALTER
##MS_ServerStateManager##	ALTER SERVER STATE VIEW SERVER STATE VIEW SERVER PERFORMANCE STATE VIEW SERVER SECURITY STATE	VIEW DATABASE STATE VIEW DATABASE PERFORMANCE STATE VIEW DATABASE SECURITY STATE
##MS_ServerStateReader##	VIEW SERVER STATE VIEW SERVER PERFORMANCE STATE VIEW SERVER SECURITY STATE	VIEW DATABASE STATE VIEW DATABASE PERFORMANCE STATE VIEW DATABASE SECURITY STATE
##MS_ServerPerformanceStateReader##	VIEW SERVER PERFORMANCE STATE	VIEW DATABASE PERFORMANCE STATE
##MS_ServerSecurityStateReader##	VIEW SERVER SECURITY STATE	VIEW DATABASE SECURITY STATE
##MS_DefinitionReader##	VIEW ANY DATABASE VIEW ANY DEFINITION VIEW ANY PERFORMANCE DEFINITION	VIEW DEFINITION VIEW PERFORMANCE DEFINITION VIEW SECURITY DEFINITION

Fixed server-level role	Server-level permission	Database-level permissions
	VIEW ANY SECURITY DEFINITION	
##MS_PerformanceDefinitionReader##	VIEW ANY PERFORMANCE DEFINITION	VIEW PERFORMANCE DEFINITION
##MS_SecurityDefinitionReader##	VIEW ANY SECURITY DEFINITION	VIEW SECURITY DEFINITION

Table 27: ##MS_ roles permissions

Some of the permissions assigned to the above defined roles are newly introduced in SQL Server 2022. These permissions consist of more granular permissions based on permissions already existing in SQL Server (documented in [PERM]) and they have been defined based on *the Principle of Least Privileges*. Therefore, these new permissions are a subset of existing coarse-grained permissions.

Ten new permission are introduced in SQL Server 2022, five at server level plus five at database level:

Server level:

1. VIEW ANY SECURITY DEFINITION
2. VIEW ANY PERFORMANCE DEFINITION
3. VIEW SERVER SECURITY STATE
4. VIEW SERVER PERFORMANCE STATE
5. VIEW ANY CRYPTOGRAPHICALLY SECURED DEFINITION

Database level:

1. VIEW DATABASE SECURITY STATE
2. VIEW DATABASE PERFORMANCE STATE
3. VIEW SECURITY DEFINITION
4. VIEW PERFORMANCE DEFINITION
5. VIEW CRYPTOGRAPHICALLY SECURED DEFINITION

The purpose of these permissions can be summarized as follows:

- SECURITY STATE and SECURITY DEFINITION: These permissions are required to access view which contains security-wise sensitive information.
- PERFORMANCE STATE and PERFORMANCE DEFINITION: These permissions covers all other views which do not contain security-wise sensitive information.
- CRYPTOGRAPHICALLY SECURED DEFINITION: These permissions cover a very specific subset of information within certain objects. They do not protect any specific view but rather specific columns in some objects. The background is that there are a few data fields that contain information that is much more sensitive than a username, algorithms strengths or granted permissions are: Hashed passwords and encrypted keys.

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, section: SQL Server Audit (Database Engine)]) 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, section: "CREATE SERVER AUDIT (Transact-SQL)"]).

6.1.1 Examples of Use

The following example creates a CC-compliant SERVER AUDIT object²:

```
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);
```

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

² The directory to which the file path points to has to be created before.

as a varchar(4) as retrieved via sys.fn_get_audit_file but needs to be specified as an integer. To translate 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 Reviewing the Audit Log and Audit Record Contents

To review the audit log an authorized administrator can use the system function sys.fn_get_audit_file (requires Control Server permission). This function takes three arguments:

- Specification of a set of audit files to be read
- Specification of an audit file from the set to start reading from (initial file name)
- Specification of an audit record from the initial audit file to start reading from

The last two arguments can be set to the default value to read all audit files and records.

The complete specification of sys.fn_get_audit_file can be found in [AGD, section: "sys.fn_get_audit_file (Transact-SQL)"].

In order to provide the audit records to the user the function is used as part of a SELECT statement. Thereby a selection and ordering of the audit records can be performed through the WHERE and ORDER BY clauses.

For each logged event the following information is recorded as required by the Security Target (please see [AGD, section: "SQL Server Audit Records"] 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 28: 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 groups³ 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
Use of special permissions (e.g., those often used by authorized administrators to circumvent access control policies)	IMP (IMPERSONATE)	DATABASE_PRINCIPAL_IMPERSONATION_GROUP SERVER_PRINCIPAL_IMPERSONATION_GROUP
All modifications to the audit configuration that occur while the audit collection functions are operating.	AL (ALTER)	AUDIT_CHANGE_GROUP

³ An audit action group is a set of actions which are functionally related.

Event	Action_id (name)	Containing Action Group Name
Successful requests to perform an operation on an object covered by the SFP.	CR (CREATE) DR (DROP) AL (ALTER) ⁴	DATABASE_OBJECT_ACCESS_GROUP DATABASE_CHANGE_GROUP DATABASE_OBJECT_CHANGE_GROUP SCHEMA_OBJECT_ACCESS_GROUP SCHEMA_OBJECT_CHANGE_GROUP SERVER_OBJECT_CHANGE_GROUP
Unsuccessful use of the authentication mechanism	LGIF (LOGIN FAILED)	FAILED_LOGIN_GROUP FAILED_DATABASE_AUTHENTICATION_GROUP
Unsuccessful use of the user identification mechanism, including the user identity provided	DBAF (DATABASE AUTHENTICATION FAILED)	
Unsuccessful binding of user security attributes to a subject (e.g. creation of a subject)	CR (CREATE) AL (ALTER) APRL (ADD MEMBER) DPRL (DROP MEMBER)	DATABASE_PRINCIPAL_CHANGE_GROUP SERVER_PRINCIPAL_CHANGE_GROUP DATABASE_ROLE_MEMBER_CHANGE_GROUP SERVER_ROLE_MEMBER_CHANGE_GROUP
Unsuccessful revocation of security attributes ⁵	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

⁴ 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.

⁵ This applies to revocation of database role memberships and permissions.

Event	Action_id (name)	Containing Action Group Name
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 membership for Server scoped roles	APRL (ADD MEMBER) DPRL (DROP MEMBER)	SERVER_ROLE_MEMBER_CHANGE_GROUP
Create and destroy database scoped groups	CR (CREATE) DR (DROP)	DATABASE_PRINCIPAL_CHANGE_GROUP
Create, Start and Stop Audit	CR (CREATE) AUSE (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
Manage Attributes for Session Establishment	EX (EXECUTE)	SCHEMA_OBJECT_ACCESS_GROUP
Define the action to take in case the audit file is full	CR (CREATE) AL (ALTER)	AUDIT_CHANGE_GROUP

Event	Action_id (name)	Containing Action Group Name
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
Rejection of a new session based on the limitation of multiple concurrent sessions	LGIF (LOGIN FAILED) UDAU (USER DEFINED AUDIT)	FAILED_LOGIN_GROUP USER_DEFINED_AUDIT_GROUP
Denial of a session establishment due to the session establishment mechanism	LGIF (LOGIN FAILED) UDAU (USER DEFINED AUDIT)	FAILED_LOGIN_GROUP USER_DEFINED_AUDIT_GROUP

Table 29: Audit Events

6.2.1 Configuration

The following configuration creates a CC-compliant SERVER AUDIT SPECIFICATION based on the server-level events listed in Table 29. 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 (FAILED_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_PRINCIPAL_IMPERSONATION_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_PRINCIPAL_IMPERSONATION_GROUP),
ADD (SERVER_ROLE_MEMBER_CHANGE_GROUP),
```

```
ADD (SERVER_STATE_CHANGE_GROUP),  
ADD (USER_DEFINED_AUDIT_GROUP),  
ADD (TRANSACTION_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, section: "SQL Server Audit Action Groups and Actions"].

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 specific 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. The user can select the database engine shut down or overwrite old audit files if a certain size is reached. In those cases, administrators shall consider to back-up the audit files and start over the audit process with a new set of files. An example of use is shown in this document, in section 6.1.1. A proper configuration in case of running out of disc space is provided in section 8.1.
- 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.

All the security events of the TOE can be, by default, managed by the user or users with the role **sysadmin**. This role is in charge of the logs revision in search of all the security events that are described on previous paragraphs and take the appropriate measures in order to fix them or find a security breach.

7 Security Server Policies Configuration

In it is required to configure the server in order to enable the account policies configuration. In the following lines the procedure to enable this Windows Configuration is going to be explained.

First, in the computer that is installed the TOE, open the Server Manager application and then select the option Tools (Figure 34):

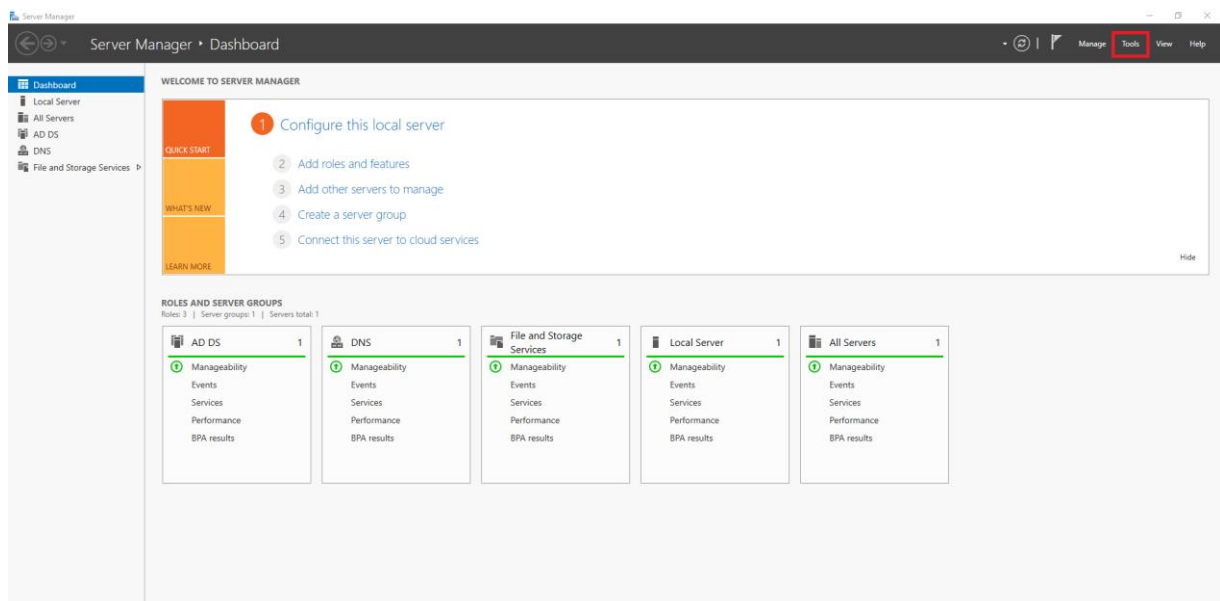


Figure 34: Tools option selection

In the Tools menu, select the option Local Security Policy (Figure 35):

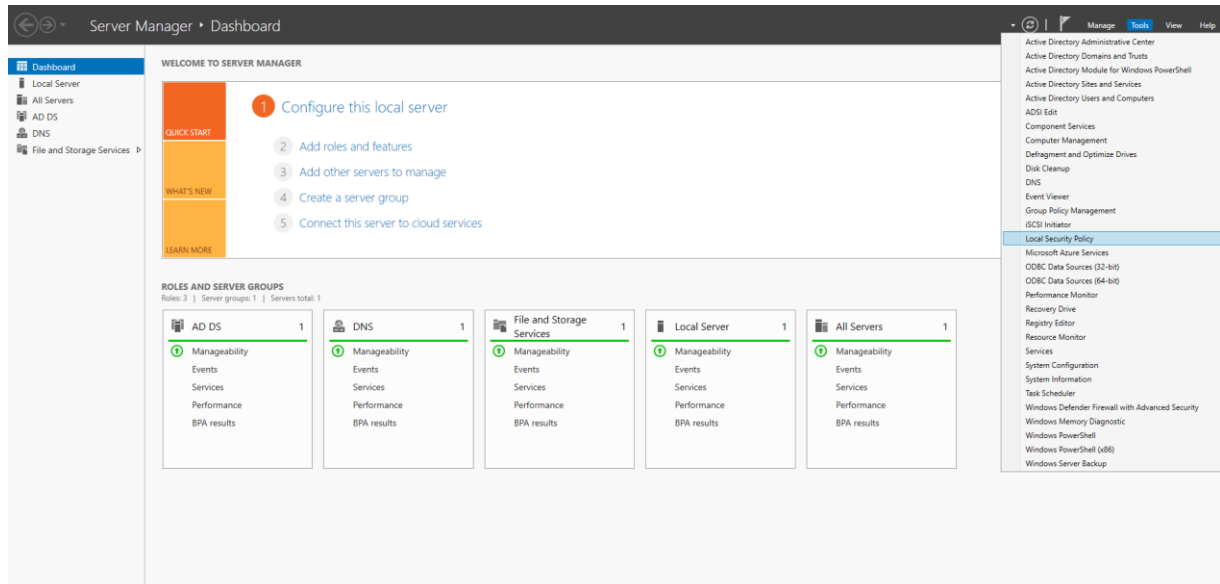


Figure 35: Local Security Policy option selection

This option opens a menu with some Account Policies options for configuration, select the option Account Policies, and click to pen the dropdown menu (Figure 36):

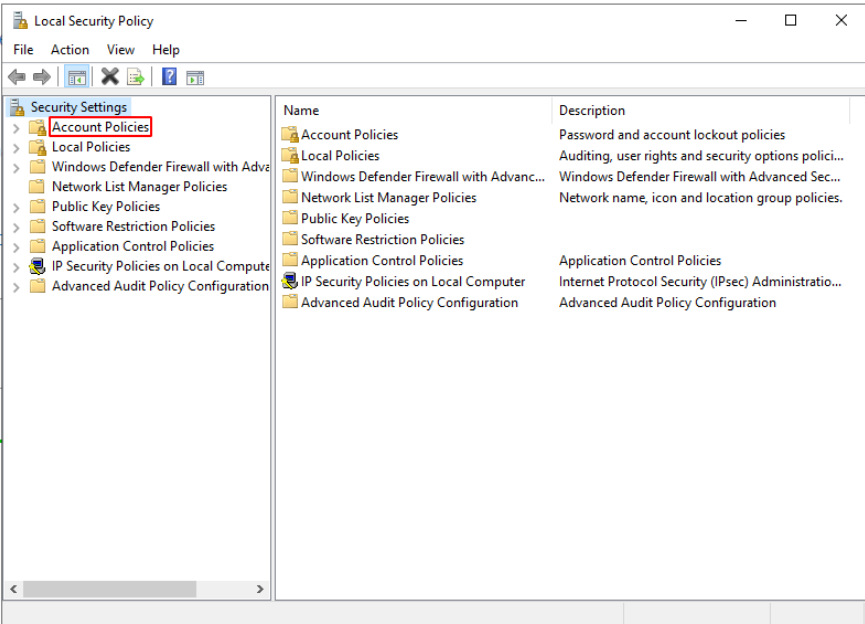


Figure 36: Account Policies modification

Inside of the dropdown menu, select the option Account Lockout Policy, press mouse right button over Account lockout threshold and select edit properties (Figure 37):

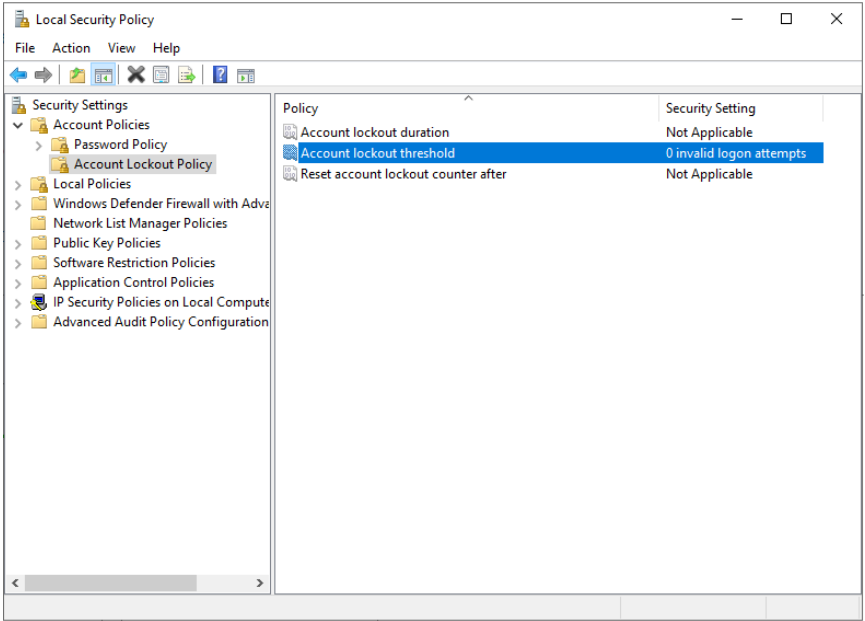
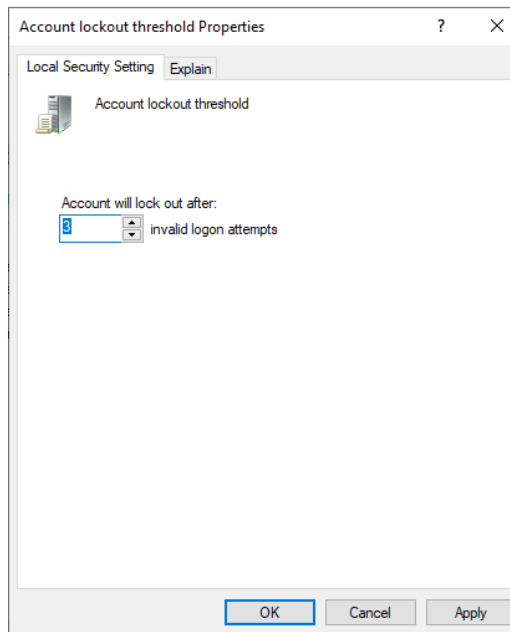
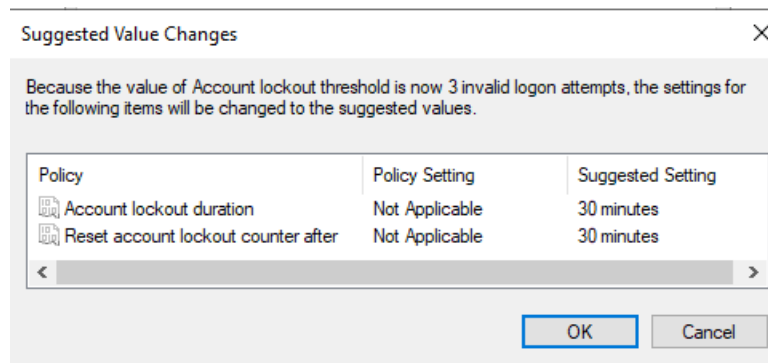


Figure 37: Number of attempts modification

A new menu should be opened. In this menu change the threshold properties from 0 (default value) to 3 as in the image below (Figure 38):

**Figure 38: Selection of attempts**

Press button ok and a new pop-up window will be opened with some suggested values for other security policy options. Press ok on this pop-up window and close the Server Manager (Figure 39).

**Figure 39: Suggested values change**

Furthermore, it is necessary to configure the minimum character length of password, complexity and duration. These facts can be configured in tab Password Policy on Local Security Policy tool. (Figure 40)

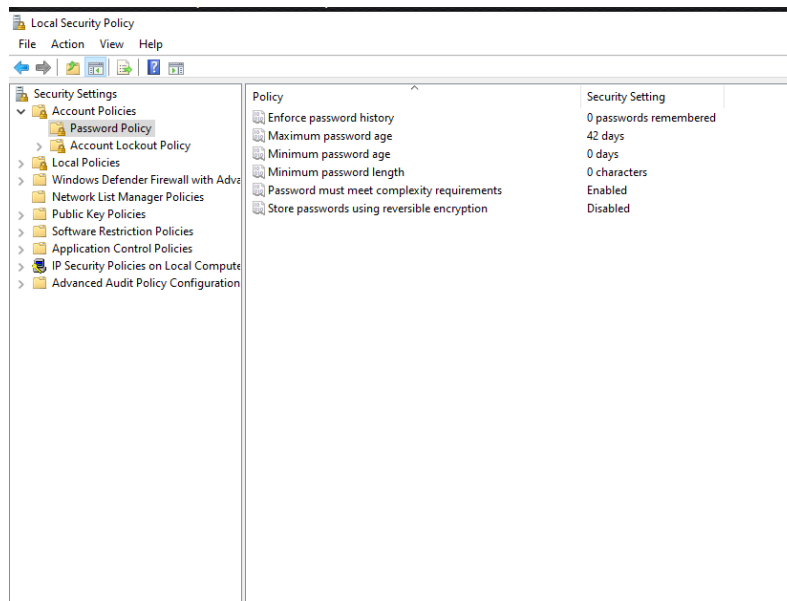


Figure 40: Password Policy Tab

The first modification is the setting Maximum password age. By default, is 42 days, it is necessary to change from a lower value (For example, 30 days can be a realistic and secure value). This can be done with right click mouse button and select edit properties. A pop-up window should be appeared (Figure 41), on this window the value can be adjusted with two arrows.

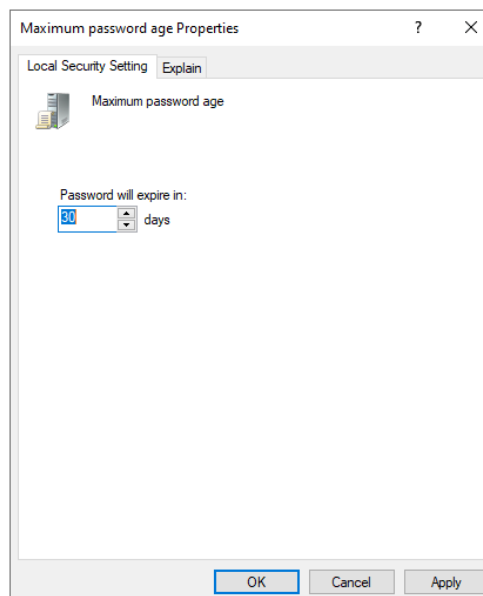


Figure 41: Change password duration to 30 days

The next characteristic to be modified is the Minimum Password Length. By default, is 0, it is necessary to change from a more secure value, this must be at least 12. This value can be changed with right click mouse button and select edit properties. A pop-up window should be appeared (), on this window the value can be adjusted with two arrows, as same on previous characteristic.

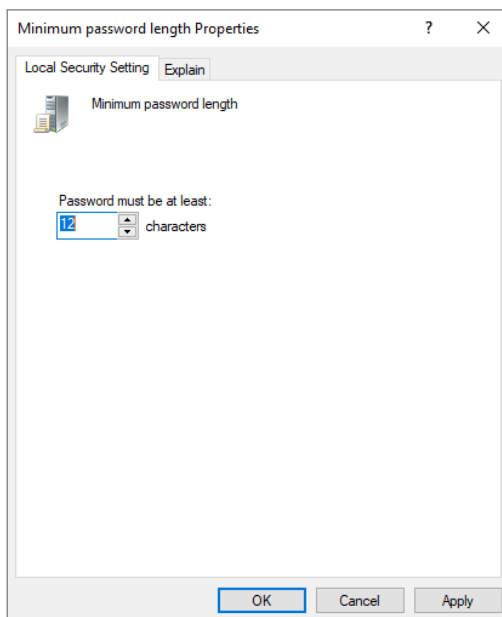


Figure 42: Change password length to 12 characters

This configuration could help to avoid some distinct attacks, such as brute-force, dictionaries or rainbow table attacks.

8 Requirements for secure administration, configuration and usage

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

8.1 Requirements about Security Audit

It is required 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.

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.

The most adequate and secure option is SHUTDOWN, that avoids the attacker to manipulate the logs of the TOE, because the instance shutdowns when a certain size of log files is reached. This size can be configured by the user. Taking into account, that this option can affect the availability of the server instance.

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.

Natively compiled stored procedures do not provide statement-level auditing. Only the execution of the SP as a whole can be audited. Therefore, they do not fulfil the CC requirements and may not be used in the certified configuration.

Please note that some Audit Action Groups do not log the outcome of the contained actions, but only the outcome of the permission check for the statement. This is especially problematic if an operation is un-done ("rolled back"). In order to determine if a statement has been executed successfully, the administrator can use the following Action IDs:

- TXBG: BEGIN TRANSACTION
- TXRB: ROLLBACK TRANSACTION
- TXCM: COMMIT TRANSACTION
- UNDO: UNDO STATEMENT

To determine whether a statement succeeded or failed the administrator should first group the audit records by Session ID, Transaction ID and Sequence Group ID.

Then the administrator can identify whether an UNDO for a specific statement has been performed and that this statement has therefore not succeeded. In addition to this only statements within a TXBG and TXCM action are finally executed, statements within a TXBG and TXRB (or AUSE – Audit Session Changed) are rolled back.

Every user with the appropriate permissions can manage the security audit. By default, only the role *sysadmin* is the role that is in charge of this sensitive task. It is one of the security pillars of this TOE. As it is described in section **2.1.1 Trusted and competent administrator**, the role *public* has no permissions to perform any action related to this requirement.

8.2 Requirements 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 [PERM] that can be downloaded from [WEB]. This file contains an overview of the server and database level permissions in SQL Server. It also contains information on the permission syntax and instructions on how to read the permission charts.

According to the concept for Access Control in SQL Server 2022 it is possible (if not likely) that two users/administrators have the same permission for one object. This could lead to 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 2022 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 must 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 these Stored Procedures from *public*.

The description of the *sp_droprsvrolemember* in [AGD, section: "*sp_droprsvrolemember* (Transact-SQL)"] 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 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, section: "*CREATE LOGIN* (Transact-SQL)"] 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 by default visible to all users of the *public* role, cf. [AGD, section: "*DB_NAME* (Transact-SQL)"]. Therefore, care should be taken that the database name does not leak information about its contents.

Setting the *TRUSTWORTHY* flag on a database may allow database users to escalate their privileges to the database owner's privileges. Therefore, the *TRUSTWORTHY* flag shall be used with special care and shall only be used if an escalation of privileges can be excluded.

Every user with the appropriate permissions can manage the access control. By default, the role *securityadmin* is the role that should be in charge of this sensitive task, because this role can manage any login of the TOE. As it is described in section **2.1.1 Trusted and competent administrator**, the role public has no permissions to perform any action related to this requirement.

8.3 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://docs.microsoft.com/en-us/sql/relational-databases/security/strong-passwords>

The specific settings for the enforcement of minimum password requirements on the underlying Operating System depend on the actual installation. To allow the secure operation of the TOE the administrator must ensure that the OS enforces strong password using not less than the following settings:

- Password must be at least 12 characters in length
- Option "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'.

'CHECK_POLICY=on' enables "Account lockout duration", "account lockout threshold", and "reset account lockout counter after". The security policy can be set in Windows (Local Security Policy) and should define that an account is locked out after three invalid logon attempts. The minimum lockout duration is one minute (For further detail, see section 7). This task depends on the configuration of the OS. Refer to **section 7 Security Server Policies Configuration** for further details.

8.4 Other requirements

Furthermore, it is required 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, section:

“KILL (Transact-SQL) [”]) 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 2022 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, section: “ntext, text, and image (Transact-SQL)”]) 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 Security Update Guide (<https://portal.msrc.microsoft.com/en-us/security-guidance>) from Microsoft Security Response Center website (<https://www.microsoft.com/en-us/msrc>) 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 Security Response Center 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://msrc.microsoft.com/create-report>). Moreover, users who want to report any security issue should follow the guidance published at: <https://www.microsoft.com/en-us/msrc/faqs-report-an-issue>.

The permission “Perform volume maintenance tasks” must not be granted to the account which runs the TOE (by default: NT SERVICE\MSSQLSERVER). If this permission is granted the TOE may not protect residual information in a resource when the resource is re-allocated.

The backup functionality of SQL Server 2022 (cf. [AGD, section: “BACKUP (Transact-SQL)”]) writes database contents directly to disk, tape, Azure or a Virtual Device.

These requirements must be ensured by the person who is in charge of the TOE when they are related to the Operating System and the TOE's environment.

Regarding of KILL command, any user with appropriate permissions can perform it. By default, the *processadmin* role is in charge of this task although any competent and trusted administrator can perform this action as it is defined in section **2.1.1 Trusted and competent Administrator**.

The role sysadmin, by default, is the only role that can manage permission administration and perform tasks of internal database configuration however any role with appropriate permissions can perform this action when this role is a trusted and competent administrator as it is defined on section **2.1.1 Trusted and competent Administrator**.

At last but not least, the backup functionality is reserved by default, to the database role *db_backupoperator* that shall ensure that any critical database or log on its charge is conveniently saved on a disk; although any competent and trusted administrator can perform this action as it is defined in section **2.1.1 Trusted and competent Administrator**.

9 Appendix

9.1 Stored Procedures

The following chapters contain information on Stored Procedures that are contained in SQL Server 2022 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.

9.1.1 sp_Msgetversion

This Stored Procedure can be used to get the current version of SQL Server 2022.

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

9.1.2 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:

9.1.3 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'

9.1.4 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

9.1.5 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

9.1.6 xp_qv

This Stored Procedure wraps SQLBOOT's QueryProductValue function.

Input: <setting>, <instancename>

Output: 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

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_qv '<setting>' [, '<instancename>']

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

Example: declare @sqlbootvalue int
 exec @sqlbootvalue = xp_qv '2745196162'
 select @sqlbootvalue 'VALUE_REPLICATION'

9.1.7 xp_instance_regread

See xp_regread for details

9.1.8 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
```

9.1.9 Sp_remove_maximum_number_of_connections_limit

Functionality: 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.

Input: None

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

Permission: Requires the CONTROL SERVER permission.

Return: 0 (Success) or >0 (Failure)

Syntax: dbo.sp_remove_maximum_number_of_connections_limit

9.1.10 Sp_revoke_logon_denies

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

Input: `[@login_name=] 'login'`

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

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

Permission: Requires the CONTROL SERVER permission.

Return: 0 (Success) or >0 (Failure)

Syntax: `sp_revoke_logon_denies [@login_name=]'login'`

9.1.11 Sp_set_maximum_number_of_connections_per_login

Functionality: 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.

Input: `[@max_connections=] max_connections`

New value for the maximum number of allowed connection per login.

Max_connections is of data type INT.

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

Permission: Requires the CONTROL SERVER permission.

Return: 0 (Success) or >0 (Failure)

Syntax: `dbo.sp_set_maximum_number_of_connections_per_login`

`[@max_connections=] max_connections`

9.1.12 Sp_deny_logon

Functionality: 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.

Input: `[@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).

Comments: 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, this Stored Procedure is stored in the dbo-schema of the master database.

Permission: Requires the CONTROL SERVER permission.

Return: 0 (Success) or >0 (Failure)

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

9.1.13 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_ACCESSDENIED 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

10 References

All documentation references (abbreviations, glossary, and document references) have been moved to an external reference document, which will be updated with every new document version:

[AGD]	Microsoft SQL Server 2022 Technical Documentation (file: Offline-Book_SQL-Server-2022_1.0_2022-12-23.zip)
[PERM]	Poster containing information on the permissions model of the TOE poster (file: Microsoft_SQL_Server_2017_and_Azure_SQL_Database_permissions_infographic_1.0.pdf)
[Ref]	Reference List - CC Evaluation of SQL Server 2022, Microsoft Corp.
[ST]	Microsoft SQL Server 2022 Database Engine - Common Criteria Evaluation (EAL4+) - Security Target, last version listed in [Ref]
[TDS-SPEC]	https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-tds/893fcc7e-8a39-4b3c-815a-773b7b982c50
[WEB]	https://www.microsoft.com/en-us/sql-server/data-security (click on "View our Common Criteria certification" and a PDF document will be downloaded)