STORMSHIELD

GUIDE

STORMSHIELD ENDPOINT SECURITY

SQL SERVER RECOMMENDATIONS

Version 7.2

Document last update: April 07, 2022

Reference: ses-en-SQL_server_recommendations_guide-v7.2

Table of contents

In the documentation, Stormshield Endpoint Security is referred to in its short form: SES.

Requirements

This document is intended for system and network administrators familiar with Microsoft SQL Server and its tools.

To configure SQL Server, ensure that you meet the following requirements:

- You must have physical or remote access to the machine that hosts the SQL Server instance for SES 7.2 so that you can perform operations on files or services.
- You must install SQL Management Studio and hold administrator privileges on the instance of the database used for SES 7.2. This may be a *sa* SQL account or a Windows account with administrator privileges on the SQL Server instance.
- To prevent performance issues, install SQL Server on SSD disks or mechanical disks with a rotational speed above 10,000 revolutions/minute.
- If SQL Server is installed in a virtualized environment, you are advised to use the RAID 10 system instead of RAID 5.

Optimizing the configuration of SQL Server

Many settings can be adjusted to improve the performance of SQL Server.

Placing database files on one or several dedicated physical disks

In pools that include many agents, and in which the policy generates a large volume of logs, insertions in the database will greatly increase disk access.

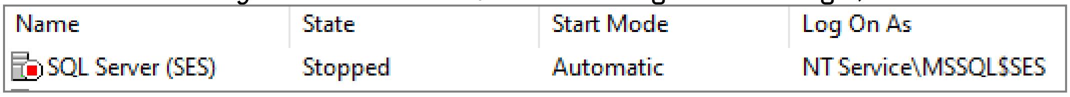
To improve disk-related performance, we recommend placing data files and transaction log files on separate disks for the *urd* and *stormshield* databases, which are the most often used.

> **NOTE**
> Creating two partitions on the same physical disk to spread out the files will not improve performance.

Perform these operations outside business hours.

1. Stop the Stormshield Endpoint Security Server service.
2. In SQL Management Studio, verify the current location of the files to find out whether they are on a dedicated disk:
```
SELECT name, physical_name AS Location, state_desc AS OnlineStatus
FROM sys.master_files
WHERE database_id in (DB_ID(N'stormshield'), DB_ID(N'urd'))
GO
```
3. Create one or several destination folders. In the following example, data files (*.MDF*) will be placed on the disk *d:\SqlData* and transaction log files (*.LDF*) on *e:\SqlLog*.

4. Locate the SID that the SQL Server instance uses. You can:
    - Either look in the *Log On As* column in SQL Server Configuration Manager,



Name	State	Start Mode	Log On As
SQL Server (SES)	Stopped	Automatic	NT Service\MSSQL$SES

    - Or use the following command line in a Windows command window:

```
wmic service where "name like 'MSSQL%'" get Name,StartName
```

    In our example, the SID used in the instance is *NT Service\MSSQL$SES*

5. Grant full access privileges to folders that will contain SQL Server files, by using the following command lines in a command prompt:

```
icacls.exe d:\SqlData /inheritance:e /grant "NT Service\MSSQL$SES:
(OI)(CI)(F)"
icacls.exe e:\SqlLog /inheritance:e /grant "NT Service\MSSQL$SES:
(OI)(CI)(F)"
```

6. In SQL Management Studio, indicate the new file location:

```
ALTER DATABASE stormshield
MODIFY FILE ( NAME = stormshield, FILENAME =
'D:\SqlData\stormshield.mdf');
GO

ALTER DATABASE stormshield
MODIFY FILE ( NAME = stormshield_log, FILENAME =
'E:\SqlLog\stormshield_log.ldf');
GO

ALTER DATABASE urd
MODIFY FILE ( NAME = urd, FILENAME = 'D:\SqlData\urd.mdf');
GO

ALTER DATABASE urd
MODIFY FILE ( NAME = urd_log, FILENAME = 'E:\SqlLog\urd_log.ldf');
GO
```

7. Stop the SQL Server instance:
    - Either via the pop-up menu of the instance in SQL Server Configuration Manager,
    - Or run the following command:
      `sc stop MSSQL$SES` (MSSQL$SES being the name of the SQL instance service).
8. Move the files to their new folders using the Windows file explorer.
9. Ensure that the files have inherited the folder's privileges, i.e., *FullControl* for the SID *NT Service\MSSQL$SES*.
    a. Run the command `icacls`:

```
icacls d:\SqlData\*
icacls e:\SqlLog\*
```

    b. Verify the privileges for each file:

```
NT SERVICE\MSSQL$SES:(I)(F)
```

    c. If the *FullControl* (F) access privilege does not exist for a file, run the following command to force permission to the *NT Service\MSSQL$SES* service:

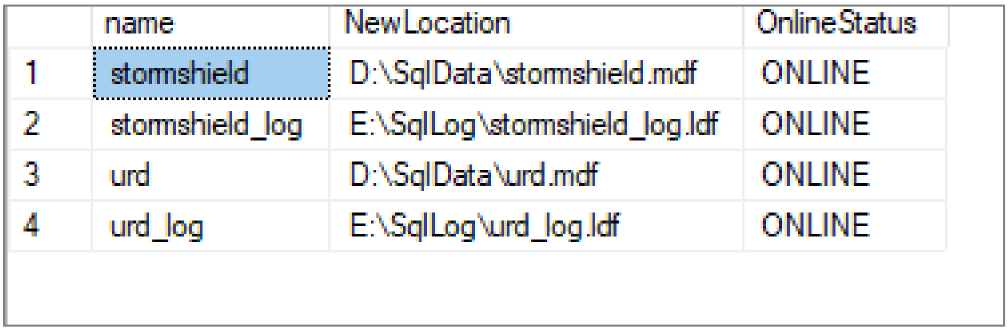
```
icacls d:\SqlData\stormshield.mdf /grant "NT Service\MSSQL$SES:
(F)"
```

    By replacing the path in bold with the path of the file for which access is missing.

10. Restart the SQL Server instance service:
    - Either via the pop-up menu of the instance in SQL Server Configuration Manager,
    - Or run the following command:
      `sc start MSSQL$SES` (MSSQL$SES being the name of the SQL instance service).

11. In SQL Management Studio, verify the access path to the files as well as the status, which must be ONLINE:

```
SELECT name, physical_name AS NewLocation, state_desc AS
OnlineStatus
FROM sys.master_files
WHERE database_id in (DB_ID(N'stormshield'), DB_ID(N'urd'))
GO
```



	name	NewLocation	OnlineStatus
1	stormshield	D:\SqlData\stormshield.mdf	ONLINE
2	stormshield_log	E:\SqlLog\stormshield_log.ldf	ONLINE
3	urd	D:\SqlData\urd.mdf	ONLINE
4	urd_log	E:\SqlLog\urd_log.ldf	ONLINE

12. Restart the Stormshield Endpoint Security Server service.

Setting the size and growth value of data files and transaction logs

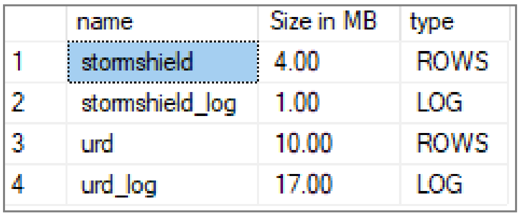
When a database file or transaction log file reaches its full capacity, SQL Server locks access to tables while it increases the size of the file, which will cause slowdowns when requests are processed.

Small but frequent growth of the files will also increase their fragmentation on the disk.

You are strongly advised to adapt the size and growth value of the most frequently used database files - *urd* and *stormshield*. This operation only needs to be performed once, preferably as soon as possible after SQL Server is installed, and outside business hours.

1. Stop the Stormshield Endpoint Security Server service.
2. Verify the current size of the data files and transaction logs for the *urd* and *stormshield* databases. Use the following SQL request:

```
SELECT name, CAST(ROUND((size / 128.00), 0) AS NUMERIC(36, 2)) as
'Size in MB', type_desc as 'type'
FROM sys.master_files
WHERE database_id in (DB_ID(N'stormshield'), DB_ID(N'urd'))
GO
```



	name	Size in MB	type
1	stormshield	4.00	ROWS
2	stormshield_log	1.00	LOG
3	urd	10.00	ROWS
4	urd_log	17.00	LOG

3. Estimate the new size for databases:
   - For data files (type = ROWS), the new size must be rounded up to the closest multiple of 512 MB.

     > **EXAMPLES**
     > If the current size is 42 MB, the new size will be 512 MB.
     > If the current size is 1172 MB, the new size will be 1536 MB.

   - For transaction log files (type = LOG), the new size must be rounded up to the closest multiple of 256 MB.

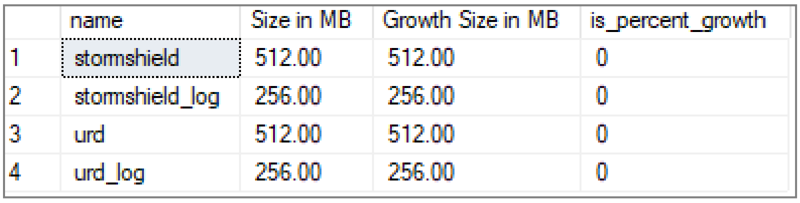
     > **EXAMPLES**
     > If the current size is 42 MB, the new size will be 256 MB.
     > If the current size is 1172 MB, the new size will be 1280 MB.

4. Configure the new size estimated in the previous step, as well as the growth value of the files. This value must be 512 MB for data files and 256 MB for transaction log files:

```
ALTER DATABASE stormshield
MODIFY FILE ( NAME = stormshield, SIZE=512MB, FILEGROWTH=512MB);
GO
ALTER DATABASE stormshield
MODIFY FILE ( NAME = stormshield_log, SIZE=256MB, FILEGROWTH=256MB);
GO
ALTER DATABASE urd
MODIFY FILE ( NAME = urd, SIZE=512MB, FILEGROWTH=512MB);
GO
ALTER DATABASE urd
MODIFY FILE ( NAME = urd_log, SIZE=256MB, FILEGROWTH=256MB);
GO
```

5. Verify the size and growth value:

```
SELECT name, CAST(ROUND((size / 128.00), 0) AS NUMERIC(36, 2)) as
'Size in MB', CAST(ROUND((growth / 128.00), 0) AS NUMERIC(36, 2)) as
'Growth Size in MB', is_percent_growth
FROM sys.master_files
WHERE database_id in (DB_ID(N'stormshield'), DB_ID(N'urd'))
GO
```



	name	Size in MB	Growth Size in MB	is_percent_growth
1	stormshield	512.00	512.00	0
2	stormshield_log	256.00	256.00	0
3	urd	512.00	512.00	0
4	urd_log	256.00	256.00	0

Do note that the **is_percent_growth** column must be set to 0 so that growth is fixed and not a percentage of the previous size.

6. Restart the Stormshield Endpoint Security Server service.

Adapting the number of files in the temporary database

The log insertion requests that SES 7.2 uses frequently make use of the TempDB temporary database. To allow several requests to function under optimal conditions, adjust the number of files linked to this database. This number depends on the number of cores linked to the SQL Server instance.

- If there are fewer than 8 cores, set as many files as the number of cores linked to the instance,
- If there are 8 or more cores, set 8 files.

Adjust the size of each file from the beginning, to avoid unexpected allocations required to increase the size of files and decrease file fragmentation.

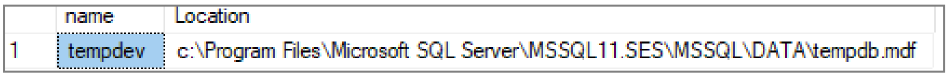
Each file can be set to 512 MB with fixed growth of 512 MB.

Perform these operations outside business hours.

To adapt the number of files in the TempDB database:

1. Determine the number of cores linked to the SQL Server instance.
2. Stop the Stormshield Endpoint Security Server service.
3. Verify the number of files associated with the TempDB database:

```
SELECT name, physical_name AS Location
FROM sys.master_files
WHERE database_id = DB_ID(N'tempDB') AND type = 0
GO
```



	name	Location
1	tempdev	c:\Program Files\Microsoft SQL Server\MSSQL11.SES\MSSQL\DATA\tempdb.mdf

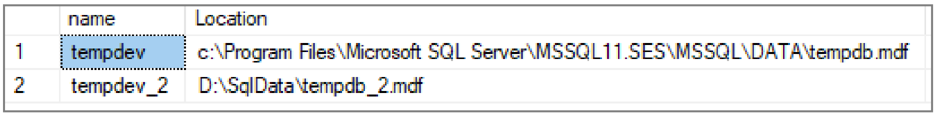
4. Add files until there are as many as the number of cores:

```
ALTER DATABASE tempDB
ADD FILE
( NAME = tempdev_2,
FILENAME = 'D:\SqlData\tempdb_2.mdf',
SIZE = 512MB,
FILEGROWTH = 512MB
)
```

We recommend placing files on disks other than the ones on which the *urd* and *stormshield* database files are stored.

5. Verify the number of files:

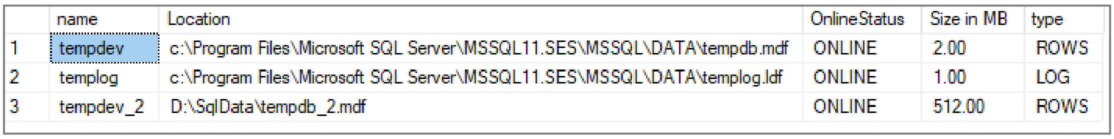
```
SELECT name, physical_name AS Location
FROM sys.master_files
WHERE database_id = DB_ID(N'tempDB') AND type = 0
GO
```



	name	Location
1	tempdev	c:\Program Files\Microsoft SQL Server\MSSQL11.SES\MSSQL\DATA\tempdb.mdf
2	tempdev_2	D:\SqlData\tempdb_2.mdf

6. Apply the recommended file location and size to the TempDB database files via the following SQL Server requests:

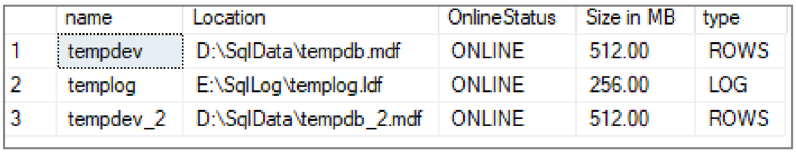
   a.
   ```
   SELECT name, physical_name AS Location, state_desc AS
   OnlineStatus, CAST(ROUND((size / 128.00), 0) AS NUMERIC(36, 2))
   as 'Size in MB', type_desc as 'type'
   FROM sys.master_files
   WHERE database_id in (DB_ID(N'tempDB'))
   GO
   ```



	name	Location	OnlineStatus	Size in MB	type
1	tempdev	c:\Program Files\Microsoft SQL Server\MSSQL11.SES\MSSQL\DATA\tempdb.mdf	ONLINE	2.00	ROWS
2	templog	c:\Program Files\Microsoft SQL Server\MSSQL11.SES\MSSQL\DATA\templog.ldf	ONLINE	1.00	LOG
3	tempdev_2	D:\SqlData\tempdb_2.mdf	ONLINE	512.00	ROWS

   For more information, refer to Placing database files on one or several dedicated physical disks and Setting the size and growth value of data files and transaction logs.

   b.
   ```
   SELECT name, physical_name AS Location, state_desc AS
   OnlineStatus, CAST(ROUND((size / 128.00), 0) AS NUMERIC(36, 2))
   as 'Size in MB', type_desc as 'type'
   FROM sys.master_files
   WHERE database_id in (DB_ID(N'tempDB'))
   GO
   ```



	name	Location	OnlineStatus	Size in MB	type
1	tempdev	D:\SqlData\tempdb.mdf	ONLINE	512.00	ROWS
2	templog	E:\SqlLog\templog.ldf	ONLINE	256.00	LOG
3	tempdev_2	D:\SqlData\tempdb_2.mdf	ONLINE	512.00	ROWS

7. Restart the Stormshield Endpoint Security Server service.

Configuring the memory of the SQL Server instance

You must allocate enough RAM to the SQL Server instance to restrict disk access in read-only mode as much as possible. However, you still need to keep some memory to allow other services to run, especially if the SES server is installed on the same machine as the SQL Server instance.

You must adjust the amount of memory allocated to the SQL Server instance according to the other services used on the machine:

1. Determine the amount of memory available on the machine that hosts the SQL Server instance . You can use the Windows Task Manager or run the following request.
   ```
   SELECT total_physical_memory_kb / 1024 as 'Total physical Memory in
   MB' FROM sys.dm_os_sys_memory
   ```

2. Determine the amount of memory to allocate to the SQL Server instance.
   - If the SQL Server instance is on a dedicated machine, allocate almost all the available memory to it. Keep only 1 GB if the machine has less than 8 GB of RAM, keep 2 GB for 8 GB of RAM and higher.

     > **EXAMPLE**
     > If the machine has 4 GB of RAM, then allocate 3 GB to the SQL Server instance. For a RAM of 16 GB, allocate 14 GB to the SQL Server instance.

   - If the SQL Server instance is on the same machine as the SES server, leave 512 MB more memory available to allow both applications to run properly.

     > **EXAMPLE**
     > If the machine has 16 GB of RAM, then allocate 14 GB-512 MB (=13.5 GB) to the SQL Server instance.

     This type of configuration is not recommended. It works well for small pools, in which the SES server and the Apache server (to generate certificates) are not often contacted.
3. Display the Memory properties of the SQL Server instance and change the **Max server memory (in MB)** value.

Configuring the maximum degree of parallelism and the cost threshold for parallelism

Maximum degree of parallelism

The maximum degree of parallelism is the number of cores that the SQL Server instance can use simultaneously to process a single request. It must be equivalent to about 25% of the number of cores linked to the SQL Server instance.

If the maximum degree of parallelism is too high, SQL requests may be blocked. As the SQL engine will not detect these blocks as deadlocks, they will be hard to detect.

A value of 0 indicates to the SQL Server instance that it can use all the cores to parallelize a costly request.

A value of 1 disables parallelism and allows only one core to be used to process the same request.

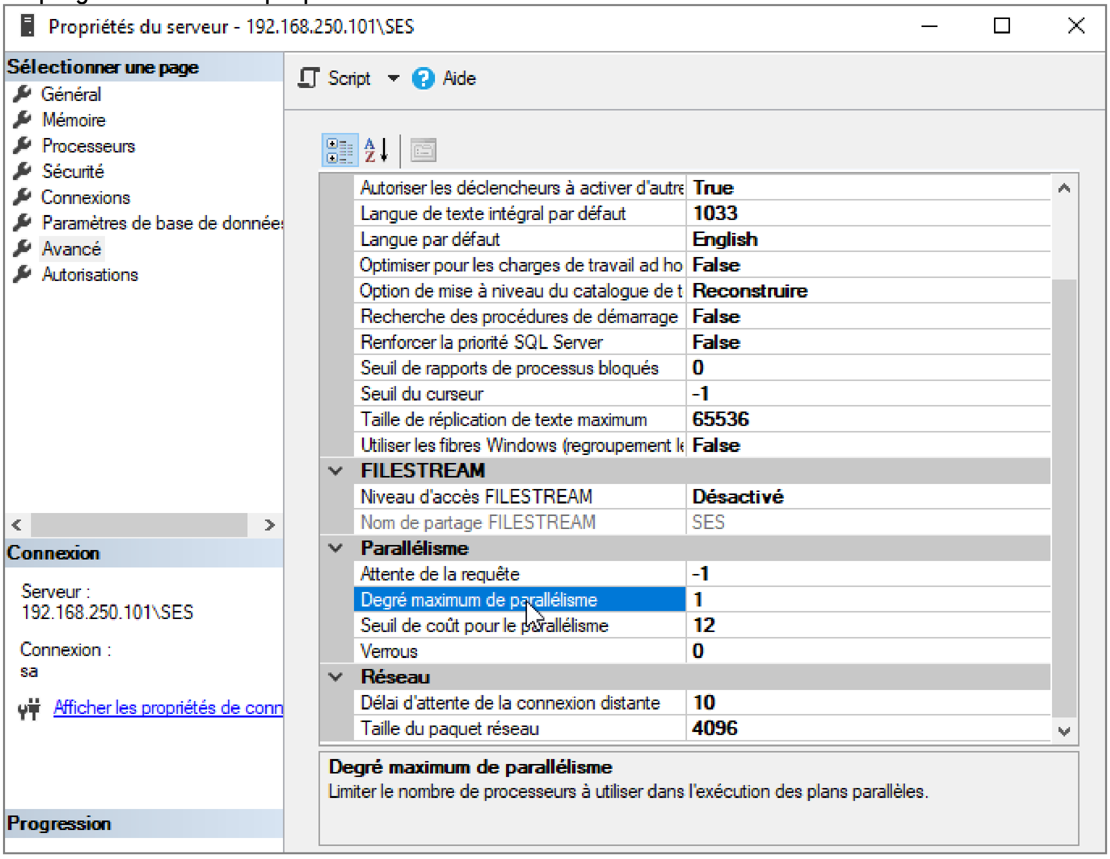
Cost threshold for parallelism

The cost threshold is the value above which SQL Server can parallelize the processing of requests. The cost of a request is evaluated with the execution plan.

This depends on the complexity of the request as well as the state of the tables in which the request will extract data. A table with statistics that are not up to date or a high-volume table will have a higher cost than a table with few entries.

Configuring the maximum degree and cost threshold for parallelism

1. Determine the number of cores linked to the SQL Server instance.
2. Display the advanced properties of the SQL instance.



Propriétés du serveur - 192.168.250.101\SES

Autoriser les déclencheurs à activer d'autre	True
Langue de texte intégral par défaut	1033
Langue par défaut	English
Optimiser pour les charges de travail ad ho	False
Option de mise à niveau du catalogue de t	Reconstruire
Recherche des procédures de démarrage	False
Renforcer la priorité SQL Server	False
Seuil de rapports de processus bloqués	0
Seuil du curseur	-1
Taille de réplication de texte maximum	65536
Utiliser les fibres Windows (regroupement l	False
FILESTREAM	
Niveau d'accès FILESTREAM	Désactivé
Nom de partage FILESTREAM	SES
Parallélisme	
Attente de la requête	-1
Degré maximum de parallélisme	1
Seuil de coût pour le parallélisme	12
Verrous	0
Réseau	
Délai d'attente de la connexion distante	10
Taille du paquet réseau	4096

Degré maximum de parallélisme
Limiter le nombre de processeurs à utiliser dans l'exécution des plans parallèles.

Connexion

Serveur :
192.168.250.101\SES

Connexion :
sa

3. Set the **Cost threshold for parallelism** to 12.
4. Set the **Maximum degree of parallelism:**
   - If the number of cores is 8, set the maximum degree of parallelism to 2,
   - If the number of cores is 4, set the maximum degree of parallelism to 1 (which disables parallelism),
   - The maximum degree of parallelism must be higher than 0.
   - The maximum degree of parallelism must always be lower than half the number of cores.

Excluding SQL Server from antivirus analyses

Antiviruses slow down processes, especially when files are opened and closed, and this may affect performance. Stormshield recommends excluding SQL Server from antivirus analyses and real-time protections.

The following are several articles written by Microsoft to assist you on the topic:

- Performance and consistency issues when certain modules are loaded into SQL Server address space
- Antivirus software that isn't cluster-aware may cause problems with Cluster Services
- How to choose antivirus software to run on computers that are running SQL Server
- Microsoft Defender Antivirus compatibility with other security products

Configuring power management on High performance

To optimize SQL Server's performance, Stormshield recommends checking and reconfiguring the machine's power management mode where necessary.

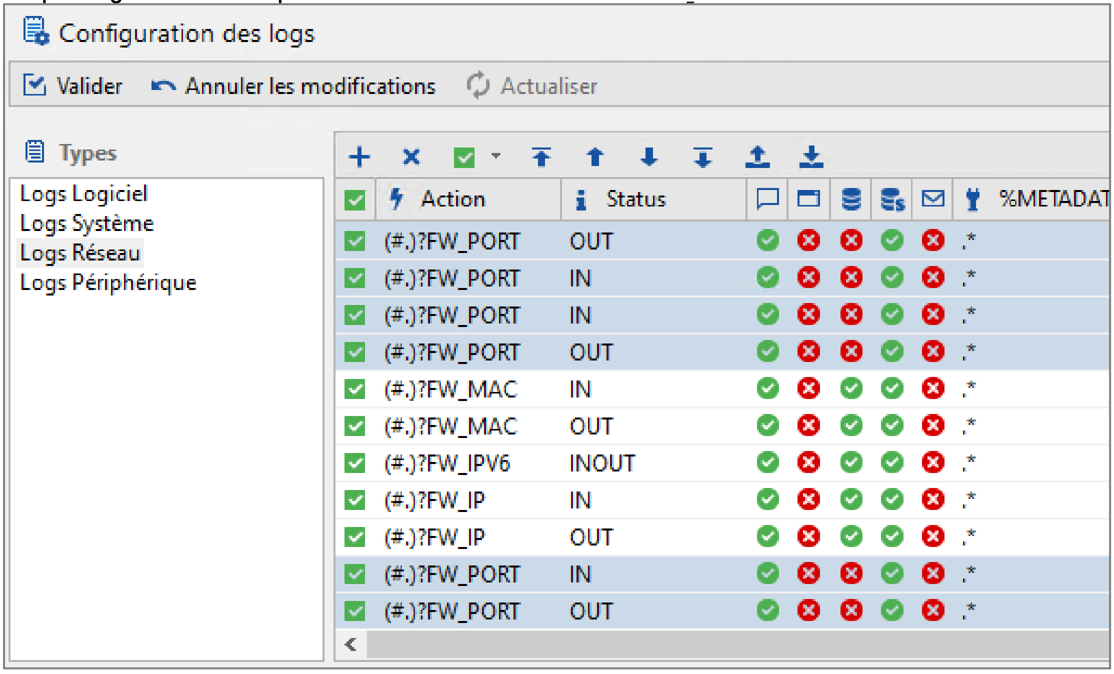
1. Open the Windows Control Panel, then select **Hardware and Sound** > **Power Options** or open a command prompt window and enter `powercfg.cpl`.
2. Check whether the selected power plan is **High performance**.
3. If it is not, select it.

Sending non-critical agent logs to a Syslog server

Depending on the policy set by the administrator, SES agents may generate a large volume of logs. If you have many agents, you are advised to use a Syslog server to which you can send logs that are not related to the operation of agents. For example, network audit or file access logs on removable devices are never analyzed in the SES 7.2 console.

This will save bandwidth between SES servers and the SQL Server instance, and save disk space as well as processing time for the SQL Server instance.

1. From the SES administration console, select **Environment Manager** > **Log configuration**.
2. Put a red cross on the column that represents the database for audit logs that appear frequently. In the example below, this would be *Action: FW PORT, Status: IN and OUT.*



Logs Logiciel
Logs Système
Logs Réseau
Logs Périphérique

☑	Action	Status	💬	▭	🛢	🛢S	✉	%METADAT
☑	(#.)?FW_PORT	OUT	✓	✗	✗	✓	✗	.*
☑	(#.)?FW_PORT	IN	✓	✗	✗	✓	✗	.*
☑	(#.)?FW_PORT	IN	✓	✗	✗	✓	✗	.*
☑	(#.)?FW_PORT	OUT	✓	✗	✗	✓	✗	.*
☑	(#.)?FW_MAC	IN	✓	✗	✓	✓	✗	.*
☑	(#.)?FW_MAC	OUT	✓	✗	✓	✓	✗	.*
☑	(#.)?FW_IPV6	INOUT	✓	✗	✓	✓	✗	.*
☑	(#.)?FW_IP	IN	✓	✗	✓	✓	✗	.*
☑	(#.)?FW_IP	OUT	✓	✗	✓	✓	✗	.*
☑	(#.)?FW_PORT	IN	✓	✗	✗	✓	✗	.*
☑	(#.)?FW_PORT	OUT	✓	✗	✗	✓	✗	.*

3. Click **Deploy to the environment**.

Always-On replication option

SQL Server's Always-On option, which enables data replication and load balancing, requires special attention.

If you use Always-On replication on *Stormshield*, *Stormshield3*, *Urd* or *SrKey* databases, they must have a full recovery mode, unlike the simple mode selected by default during the installation of SES 7.2.

SES 7.2 does not support full recovery mode as it forces transaction logs to be backed up regularly, making it possible to delete their contents. Without such backups, transaction log files accumulate until disks are saturated.

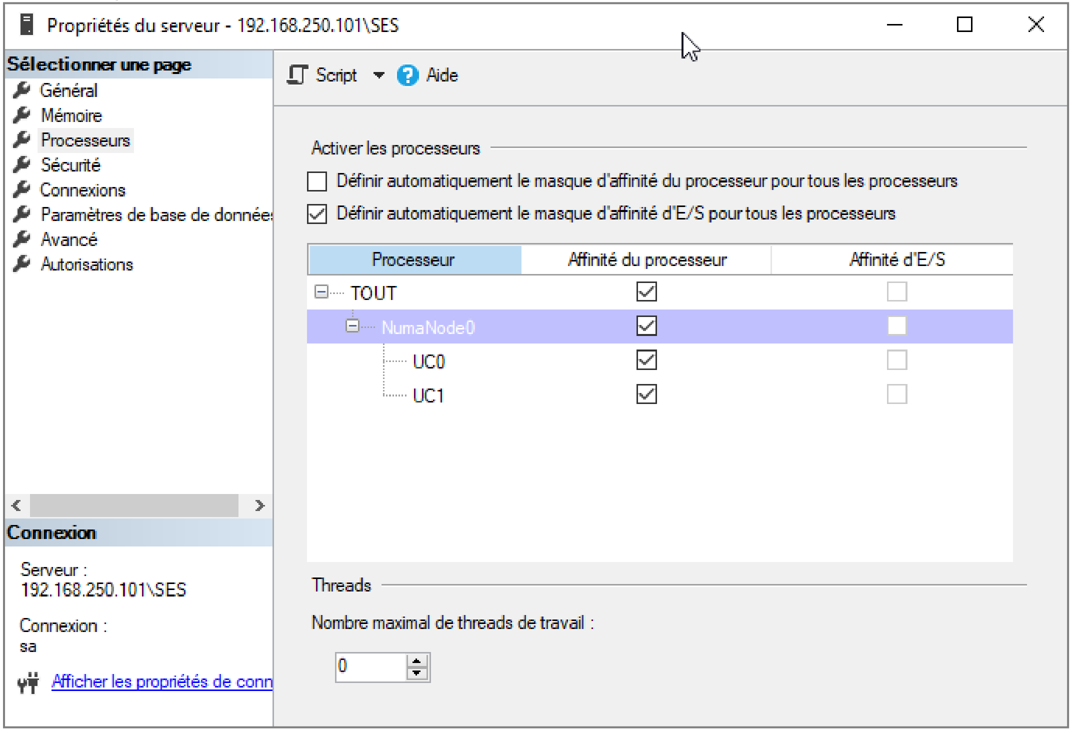
SES 7.2 is not equipped with any internal mechanism that makes it possible to regularly back up transaction logs, and therefore purge files. This is why Always-On replication is not recommended. To use this option, you must configure a regular backup of the database files in question, at least once a week.

For further information regarding backups, refer to the section Backing up databases in full recovery mode.

Determining the number of cores linked to the SQL Server instance

For the optimal configuration of the SQL Server instance for SES, you must know the number of cores on the processor linked to the instance.

- In SQL Management Studio, open the page showing the properties of the SQL Server instance and count the number of cores dedicated to the instance. There are two cores in the example below.



Propriétés du serveur - 192.168.250.101\SES

Sélectionner une page

- Général
- Mémoire
- Processeurs
- Sécurité
- Connexions
- Paramètres de base de données
- Avancé
- Autorisations

Connexion

Serveur :
192.168.250.101\SES

Connexion :
sa

Afficher les propriétés de conn

Activer les processeurs

☐ Définir automatiquement le masque d'affinité du processeur pour tous les processeurs

☑ Définir automatiquement le masque d'affinité d'E/S pour tous les processeurs

Processeur	Affinité du processeur	Affinité d'E/S
TOUT	☑	☐
NumaNode0	☑	☐
UC0	☑	☐
UC1	☑	☐

Threads

Nombre maximal de threads de travail : 0

- Or you can use the following SQL request:

```
SELECT cpu_count as 'cores' FROM master.sys.dm_os_sys_info
```

Checking the operation of the database

Several checks are required to ensure that the database runs properly.

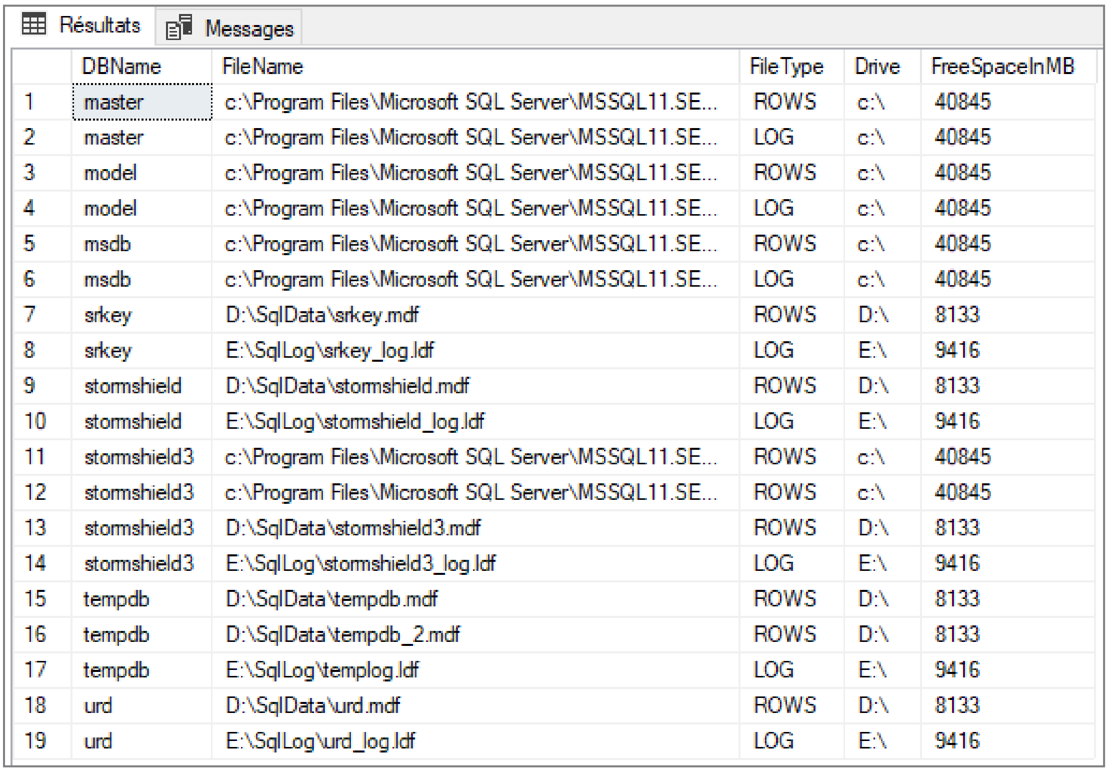
Checking available space on disks

You must regularly check how much disk space is available on the disks that hold SQL Server data files and transaction logs.

This step will detect any disk saturation and allow you to perform preventive operations before the SQL Server instance stops running properly.

- Use the file explorer on the machine that hosts the SQL Server instance or run the following SQL request:

```
 SELECT DISTINCT DB_NAME(dovs.database_id) DBName, mf.physical_name
AS FileName, mf.type_desc AS FileType, dovs.volume_mount_point AS
Drive, CONVERT(INT,dovs.available_bytes/1048576.0) AS FreeSpaceInMB
FROM sys.master_files mf
CROSS APPLY sys.dm_os_volume_stats(mf.database_id, mf.FILE_ID) dovs
ORDER BY DBName ASC, Drive ASC, FreeSpaceInMB ASC
GO
```



	DBName	FileName	FileType	Drive	FreeSpaceInMB
1	master	c:\Program Files\Microsoft SQL Server\MSSQL11.SE...	ROWS	c:\	40845
2	master	c:\Program Files\Microsoft SQL Server\MSSQL11.SE...	LOG	c:\	40845
3	model	c:\Program Files\Microsoft SQL Server\MSSQL11.SE...	ROWS	c:\	40845
4	model	c:\Program Files\Microsoft SQL Server\MSSQL11.SE...	LOG	c:\	40845
5	msdb	c:\Program Files\Microsoft SQL Server\MSSQL11.SE...	ROWS	c:\	40845
6	msdb	c:\Program Files\Microsoft SQL Server\MSSQL11.SE...	LOG	c:\	40845
7	srkey	D:\SqlData\srkey.mdf	ROWS	D:\	8133
8	srkey	E:\SqlLog\srkey_log.ldf	LOG	E:\	9416
9	stormshield	D:\SqlData\stormshield.mdf	ROWS	D:\	8133
10	stormshield	E:\SqlLog\stormshield_log.ldf	LOG	E:\	9416
11	stormshield3	c:\Program Files\Microsoft SQL Server\MSSQL11.SE...	ROWS	c:\	40845
12	stormshield3	c:\Program Files\Microsoft SQL Server\MSSQL11.SE...	ROWS	c:\	40845
13	stormshield3	D:\SqlData\stormshield3.mdf	ROWS	D:\	8133
14	stormshield3	E:\SqlLog\stormshield3_log.ldf	LOG	E:\	9416
15	tempdb	D:\SqlData\tempdb.mdf	ROWS	D:\	8133
16	tempdb	D:\SqlData\tempdb_2.mdf	ROWS	D:\	8133
17	tempdb	E:\SqlLog\templog.ldf	LOG	E:\	9416
18	urd	D:\SqlData\urd.mdf	ROWS	D:\	8133
19	urd	E:\SqlLog\urd_log.ldf	LOG	E:\	9416

In this example, the data files and transaction logs for the *stormshield*, *srkey*, *urd* and *tempdb* databases have sufficient disk space (8 GB and 9 GB of available disk space on D:\ and E:\).

Since the *master*, *model*, *msdb* and *stormshield3* databases do not grow at the same rate, they do not need to be monitored as often as the other databases.

You can therefore filter the previous request so only databases that need to be monitored are displayed:

```
SELECT DISTINCT DB_NAME(dovs.database_id) DBName, mf.physical_name AS
FileName, mf.type_desc AS FileType, dovs.volume_mount_point AS Drive,
CONVERT(INT,dovs.available_bytes/1048576.0) AS FreeSpaceInMB
FROM sys.master_files mf
CROSS APPLY sys.dm_os_volume_stats(mf.database_id, mf.FILE_ID) dovs
```

```
WHERE mf.database_id in (DB_ID(N'stormshield'), DB_ID(N'urd'), DB_ID
(N'srkey'), DB_ID(N'tempDB'))
ORDER BY DBName ASC, Drive ASC, FreeSpaceInMB ASC
GO
```

Checking processor usage of the SQL Server instance

You must monitor the SQL Server instance's processor usage to ensure that it runs properly. While it is normal for the SQL Server process to occasionally use more than 90% of the CPU, if such excessive use lasts for several hours, it may be a sign that performance is decreasing.

There are two ways to monitor its usage:

- By using a SQL request that makes it possible to retrieve the history of the past several hours:

  ```
  DECLARE @ms_ticks_now BIGINT

  SELECT @ms_ticks_now = ms_ticks
  FROM sys.dm_os_sys_info;

  SELECT record_id
  ,dateadd(ms, - 1 * (@ms_ticks_now - [timestamp]), GetDate()) AS
  EventTime ,SQLProcessUtilization
  ,SystemIdle
  ,100 - SystemIdle - SQLProcessUtilization AS
  OtherProcessUtilizationFROM (
  SELECT record.value('(./Record/@id)[1]', 'int') AS record_id
  ,record.value('
  (./Record/SchedulerMonitorEvent/SystemHealth/SystemIdle)[1]', 'int')
  AS SystemIdle
  ,record.value('
  (./Record/SchedulerMonitorEvent/SystemHealth/ProcessUtilization)
  [1]', 'int') AS SQLProcessUtilization
  ,TIMESTAMP
  FROM (
  SELECT TIMESTAMP
  ,convert(XML, record) AS record
  FROM sys.dm_os_ring_buffers
  WHERE ring_buffer_type = N'RING_BUFFER_SCHEDULER_MONITOR'
  AND record LIKE '%<SystemHealth>%'
  ) AS x
  ) AS y
  ORDER BY record_id DESC
  ```

  In the result, look in the *SQLProcessUtilization* column, which shows how much of the processor SQL Server is using.

- With the help of the task manager, look up the percentage of processor usage for the SQL Server instance.

Checking memory usage of the SQL Server instance

When SQL Server runs normally, it uses as much memory as it has available. If SQL Server's performance appears to decrease when the amount of memory is changed, ensure that the memory is correctly used. For more information, see the section Configuring the memory of the SQL Server instance.

To optimize memory usage, we recommend limiting the volume of the database by regularly running a script to purge tables. For further information, refer to the section Running the script.

Performing regular maintenance operations

Some maintenance operations must be performed regularly on SQL Server databases.

Backing up databases in full recovery mode

If you have switched to full recovery mode on SES databases, you must then perform regular backups. The backup file that is generated after a backup must not be left on the same disk because it takes up space, whereas the aim of a backup is to clear the transaction log file.

1. Check whether you are in full recovery mode by running the following request:
   ```
   SELECT name, recovery_model_desc
   FROM sys.databases
   WHERE database_id in (DB_ID(N'stormshield'), DB_ID(N'urd'), DB_ID
   (N'srkey'), DB_ID(N'stormshield3'))
   ```

   The *SIMPLE* value means that the database is in simplified recovery mode, while *FULL* means it is in full recovery mode.
2. Back up the data files.
   The backup file must not be left on the same disk as the database’s *.mdf* file.
3. If you wish to keep the data backup, move the backup file. Otherwise, delete it.
4. Back up the transaction logs.
   This operation will clear the transaction file. The generated backup file must not be left on the same disk as the database’s *.ldf* file. The file contains the transactions that took place since the last backup of transaction logs. If there were no such backups, it will contain the transactions since the last time the database was backed up.
5. If you wish to keep the transaction log backup, include the earlier backups of transaction logs since the last time the database was backed up.
   - or -
   If you do not wish to keep transaction log backups, delete the backup file.

You must create and move these backup files regularly depending on the volume of the logs.

In full recovery mode, back up transaction logs or databases at least once a week.

This backup must be performed after tables are purged, because the purge script may generate many insertions in transaction logs.

Perform backup operations outside business hours.

Below are a few resources on backing up databases.

- https://www.mssqltips.com/sqlservertutorial/7/sql-server-full-backups/
- https://www.mssqltips.com/sqlservertutorial/8/sql-server-transaction-log-backups/
- https://docs.microsoft.com/fr-fr/sql/relational-databases/backup-restore/create-a-full-database-backup-sql-server?view=sql-server-ver15
- https://docs.microsoft.com/fr-fr/sql/relational-databases/backup-restore/back-up-a-transaction-log-sql-server?view=sql-server-ver15

Maintaining performance

SES 7.2 is not equipped with any internal mechanism that makes it possible to limit the amount of data found in the databases.

In order for the SQL Server instance to maintain a high level of performance, you must delete obsolete data by using the purge script that Stormshield provides. This script deletes older logs and agent tracking data that have not been consulted in a long time.

Contact Stormshield technical support to obtain this script in different formats.

Configuring the script

The following values in the *PurgeSESDB*script can be customized:

- *@RetentionDays*: number of days' worth of logs to retain,
- *@RecordsPerIteration*: number of lines to delete for each iteration,
- *@RemoveAgentsOlderThan*: number of days agent statuses will be kept for agent tracking.

Change these values in the lines:

```
"SET @RetentionDays = "
"SET @RecordsPerIteration = "
"SET @RemoveAgentsOlderThan = "
```

Running the script

Run this script once a week outside business hours and before a backup if you intend to make one. The more often you run it, the faster it will run.

It should ideally be run from the dedicated *SqlAgent* tool provided with SQL Server, which is meant to run such tasks.

If you do not have *SqlAgent*, run it as a task from the Windows Task Planner.

There are several ways to run the script.

From SQL Server Management Studio

1. Connect to the SQL Server instance.
2. Open the *PurgeSESDB.sql* request file.
3. Run the SQL request.

With PurgeSESDB.bat

1. Open the *PurgeSESDB.*bat request file.
2. Modify the SQL Server instance:
   `Set SqlInstance=<Votre instance SQL Server>`
3. Run:
   - *PurgeSESDB.bat* if the current user has administrator privileges on the database,
   - *PurgeSESDB.bat* <login> <password> if this is not the case.

With PurgeSESDB.ps1, from a powershell session

1. Run `PurgeSESDB.ps1 -SqlServerInstance <Votre instance Sql>`.
2. Enter the login and password to connect to the SQL Server instance.

STORMSHIELD

documentation@stormshield.eu

All images in this document are for representational purposes only, actual products may differ.

Copyright © Stormshield 2022. All rights reserved. All other company and product names contained in this document are trademarks or registered trademarks of their respective companies.