

# Microsoft SQL Server

## SQL Server recommendations for Microsoft Endpoint Configuration Manager environment

SQL Server Technical Article

**Summary:** The purpose of this document is to summarize the global recommendations from a SQL Server perspective, applied specifically to a Microsoft Endpoint Configuration Manager (MECM) environment. This document includes general best practices for SQL Server configuration and management, but also a recommendation applied in Microsoft Endpoint Configuration Manager context.

**Writers:**

- Stephane Serero (Snr. Consultant Modern Workplace Microsoft)
- Frederic Michalak (PFE SQL Microsoft)
- Justin Manning (Snr. PFE SQL Microsoft)
- Ryad Ben Salah (PFE SQL Microsoft)

**Technical Reviewer:****Published:**

**Applies to:** SQL Server Versions 2008 to 2017

Date	Reviewer	Comments
<b>04/28/2017</b>	Frederic Michalak Stephane SERERO	First release of the document
<b>12/26/2017</b>	Frederic Michalak Stephane SERERO	<ul style="list-style-type: none"><li>- Read committed snapshot level</li><li>- Windows power saving setting</li><li>- Antivirus Exclusion</li><li>- Disk configuration et sizing</li><li>- Microsoft offering (POP-CMAD, POP-PowerBi)</li></ul>
<b>04/03/2018</b>	Justin Manning	<ul style="list-style-type: none"><li>- readability (grammar)</li><li>- Database Maintenance plans</li><li>- SQL Features</li><li>- SQL Server supported versions</li></ul>
<b>05/05/2018</b>	Stephane SERERO	<ul style="list-style-type: none"><li>- Product Group perf scale doc</li><li>- WSUS recommendations</li></ul>
<b>02/15/2020</b>	Ryad Ben Salah Stephane SERERO	<ul style="list-style-type: none"><li>- TempDB Update</li><li>- Virtualization policies</li><li>- MAXDOP recommendation</li><li>- Always ON Availability Groups</li><li>- SQL Reporting Services</li><li>- Antivirus exclusion update</li><li>- MECM WSUS Cleanup</li><li>- Microsoft Offerings update</li><li>- Database compatibility level</li><li>- Replace SCCM by MECM</li></ul>

<b>05/22/2020</b>	Ryad Ben Salah Stephane SERERO	<ul style="list-style-type: none"><li>- Collation update</li><li>- TSQL queries update</li><li>- CheckDB update</li><li>- Maintenance paragraphs reorganization</li></ul>
-------------------	-----------------------------------	---

## Copyright

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2016 Microsoft. All rights reserved

## Table of Contents

Introduction .....	6
1. SQL Instance settings .....	7
1.1 Maximum Degree of Parallelism (MaxDOP) .....	7
1.2 Memory.....	11
1.2.1 Maximum Server Memory .....	11
1.2.2 Minimum Server Memory.....	13
1.3 Database Collation .....	14
1.4 Additional instance settings .....	16
1.4.1 SQL instance settings .....	16
1.4.2 Trace Flag considerations.....	17
1.4.3 Local Security Policies .....	20
2. Database Settings .....	22
2.1 TempDB and autogrowth settings .....	22
2.2 Database compatibility level.....	27
2.3 Read committed snapshot isolation level.....	28
3. Global Settings .....	30
3.1 Optimize for ad hoc workloads .....	30
Optimize for ad hoc workloads Server Configuration Option .....	31
3.2 Forced Parameterization.....	32
Forced parametrization .....	34
3.3 SQL Maintenance plan .....	34
3.4 Ola.Hallengren SQL Server Maintenance.....	41
3.5 Windows operating system configuration: power saving settings.....	43
3.6 Antivirus exclusion .....	44
3.7 Change Tracking.....	45
3.8 Virtualization.....	46
3.9 Virtual Machines sizing .....	47
3.10 Microsoft Endpoint Configuration Manager (current branch) site sizing guidance .....	49
3.11 SQL Server Reporting Services .....	50
3.12 SCOM .....	54
3.13 MECM Custom Reports.....	55

3.13.1	Builtin Reports in Microsoft Endpoint Configuration Manager .....	55
3.13.2	Custom Reports in Microsoft Endpoint Configuration Manager .....	55
3.13.3	Microsoft Endpoint Configuration Manager: Advanced Dashboards.....	56
3.13.4	Microsoft Endpoint Manager: PowerBI Dashboard Integration.....	57
3.13.5	SQL Custom Report .....	60
3.14	Management Datawarehouse .....	61
3.14.1	Disk Usage .....	62
3.14.2	Server Activity & Query Statistics .....	62
3.14.3	Installation .....	63
3.14.4	Configuration .....	64
3.15	Custom queries .....	65
3.16	Supported Versions of SQL Server .....	67
3.17	WSUS.....	67
3.17.1	MECM WSUS cleanup .....	67
3.17.2	Rebuilding WSUS.....	68
3.17.3	Shared Database for Configuration Manager Software Update Points.....	68
3.18	Always ON Availability Groups.....	69

## Introduction

The goal of this White Paper is to provide guidance to Microsoft Endpoint Configuration Manager (MECM) administrators and IT professionals on how to configure and administrate the SQL Server instances related to Microsoft Endpoint Configuration Manager. This guide will deal with instance settings, database settings, global settings and maintenance plan considerations. Finally, some tips and guidance are treated at the end of this document for monitoring and performance considerations.

The recommendations exposed in this White Paper are covering mainly SQL Server versions 2008 to SQL Server 2017. Generally, the exposed topics are aimed at all SQL versions. Exceptions to a particular topic, will be mentioned with a note (for some SQL enhancements for example).

# 1. SQL Instance settings

The purpose of this section is to deal with the best practices applied to the global settings of the SQL Instance hosting the Microsoft Endpoint Configuration Manager data. Most of these recommendations are common to any application, but we will highlight recommendations specific to MECM as well.

## 1.1 Maximum Degree of Parallelism (MaxDOP)

The Maximum Degree of Parallelism (MaxDOP) parameter sets the maximum number of processors used to execute a parallelized query. In this context, we mean here processor as a core processor. By default the value MaxDOP is set to 0, implying that the SQL engine will use ALL the available cores.

### **Recommendation**

According MECM Product Group guidance, the recommended value for the MaxDOP setting in most MECM context is **0**.

However, depending on the MECM environment and/or the platform size, this value might be adjusted. To do so, you can refer to the SQL Server guidelines regarding to the MaxDOP setting.

<https://support.microsoft.com/en-us/kb/2806535>

Based on this KB article, the following query, kindly provided by Benjamin Reynolds, will help you to determine the recommended value for MaxDop setting.

```

SET NOCOUNT ON;

DECLARE @CurMDOP smallint
        ,@RecMDOP smallint
        ,@SQLMajorVersion tinyint
        ,@NumaNodes int
        ,@ProcsPerNumaNode int
        ,@IsWithinRecommendation bit
        ,@Output varchar(max);

SELECT @CurMDOP = CONVERT(smallint,value_in_use) FROM sys.configurations WHERE name = N'max degree of parallelism';
SELECT @SQLMajorVersion =
CONVERT(tinyint,LEFT(CONVERT(nvarchar(128),SERVERPROPERTY('ProductVersion')),CHARINDEX(N'.',CONVERT(nvarchar(128),SERVERPROPERTY('ProductVersion')))-1));

SELECT @NumaNodes = COUNT(NumaNodeId)
        ,@ProcsPerNumaNode = MIN(LogicalProcsPerNumaNode)
FROM (
    SELECT parent_node_id AS [NumaNodeId]
        ,COUNT(DISTINCT cpu_id) AS [LogicalProcsPerNumaNode]
    FROM sys.dm_os_schedulers
    WHERE status = N'VISIBLE ONLINE'
    GROUP BY parent_node_id
) dta;

IF @SQLMajorVersion >= 13
BEGIN
    SELECT @RecMDOP = CASE WHEN @NumaNodes = 1 THEN CASE WHEN @ProcsPerNumaNode <= 8 THEN @ProcsPerNumaNode ELSE 8 END
        ELSE CASE WHEN @ProcsPerNumaNode <= 16 THEN @ProcsPerNumaNode
        ELSE CASE WHEN @ProcsPerNumaNode*0.5 <= 16 THEN @ProcsPerNumaNode*0.5 ELSE 16 END
        END
    END;
    IF @NumaNodes = 1 AND @ProcsPerNumaNode <= 8 AND @CurMDOP <= @ProcsPerNumaNode SET @IsWithinRecommendation = 1;
    IF @NumaNodes > 1 AND @ProcsPerNumaNode <= 16 AND @CurMDOP <= @ProcsPerNumaNode SET @IsWithinRecommendation = 1;
END;
ELSE
BEGIN
    SELECT @RecMDOP = CASE WHEN @ProcsPerNumaNode < 8 THEN @ProcsPerNumaNode ELSE 8 END;
    IF @ProcsPerNumaNode <= 8 AND @CurMDOP <= @ProcsPerNumaNode SET @IsWithinRecommendation = 1;
END;

IF @CurMDOP != @RecMDOP
BEGIN
    SELECT @Output = CASE WHEN @IsWithinRecommendation = 1 THEN '-- MDOP is within the recommended setting and MAY benefit from
setting it differently.
-- The Max Recommended MDOP setting is '+CONVERT(varchar(2),@RecMDOP)+' and the server is currently configured to
'+CONVERT(varchar(2),@CurMDOP)+''.
-- Consider running the following to set the server to the max recommended setting:
'
        ELSE '-- MDOP NOT AT RECOMMENDED SETTING!
-- Recommended MDOP setting is '+CONVERT(varchar(2),@RecMDOP)+' and the server is currently configured to
'+CONVERT(varchar(2),@CurMDOP)+''.
-- Run the following to set the server to the recommended setting:
'
        END;
    SELECT @Output += 'DECLARE @AdvancedOptions bit;
SELECT @AdvancedOptions = CONVERT(bit,value_in_use)
FROM sys.configurations
WHERE name = N''show advanced options'';
IF @AdvancedOptions = 0
BEGIN
    EXECUTE sp_configure ''show advanced options'', 1;
    RECONFIGURE WITH OVERRIDE;
END;
EXECUTE sp_configure ''max degree of parallelism'', '+CONVERT(varchar(2),@RecMDOP)+';
RECONFIGURE WITH OVERRIDE;
IF @AdvancedOptions = 0
BEGIN
    EXECUTE sp_configure ''show advanced options'', 0;
    RECONFIGURE WITH OVERRIDE;
END;
GO';

    PRINT @Output;
END;
ELSE
BEGIN
    PRINT '-- MDOP configured at the recommended setting!
-- Recommended MDOP setting is '+CONVERT(varchar(2),@RecMDOP)+' and the server is currently configured to
'+CONVERT(varchar(2),@CurMDOP)+''.
END;
GO

```



### How to check

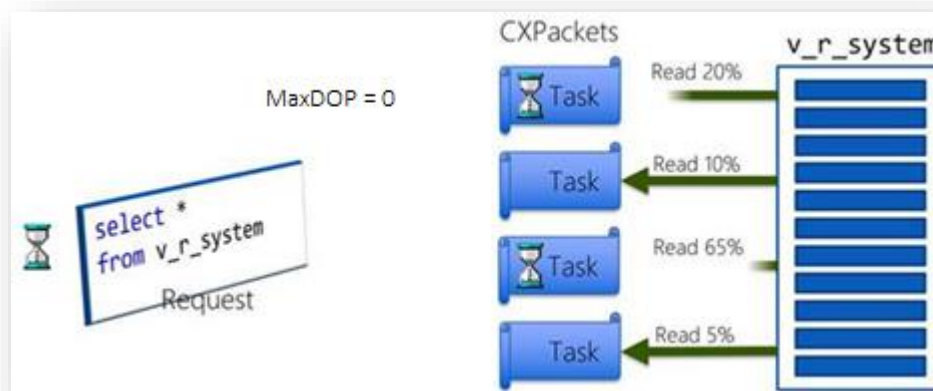
You can use the system view sys.configurations to check if your MaxDOP parameter is properly set:

```
SELECT
    name,
    value
FROM
    sys.configurations
WHERE
    name = 'max degree of parallelism'
```

### Why is this important?

If we consider an example of a parallelized query, the request is divided into several tasks, and each of them will execute a piece of the required work. As described below, because some tasks will complete sooner than others, the mechanism will imply some wait statistics called CxPacket for thread synchronization.

*Example given on 4 cores system*



Yet, the more thread you will use the more CxPacket you will get, and then the more waits you will encounter.

As there is no value which can fit to all situations, it is strongly recommended to closely monitor your SQL Server platform performance and to adjust the MaxDOP value for your environment according to your monitoring analysis and by performing tests on your platform.

If you leave the MaxDOP value to zero (default) on the server with a high number of cores, you risk having a huge CxPacket waits, impacting your MECM platform performances.

### Advanced considerations - Optional

CxPacket is not evil but setting a correct value for the MaxDOP parameter is a good way to mitigate the parallelism excess. To measure the CxPacket, you can check the global wait stats with this query:

```
select TOP 10 * from sys.dm_os_wait_stats
order by wait_time_ms DESC
```

If the CxPacket is the first wait type with a huge wait\_time\_ms value comparing to the other wait types, you can decrease again the MaxDOP value to 4, then 2 or 1. It can be especially the case if you have a very large Microsoft Endpoint Configuration Manager environment.

### Important SQL Server 2016/2017 consideration

Beginning with SQL Server 2016 SP2 and SQL Server 2017 CU3, the CXPACKET wait has been separated in two parts, CXPACKET as an actionable wait and CXCONSUMER as a negligible wait. CXCONSUMER represents now the normal part of parallel execution, and CXPACKET can be easily considered to detect parallelism issue.

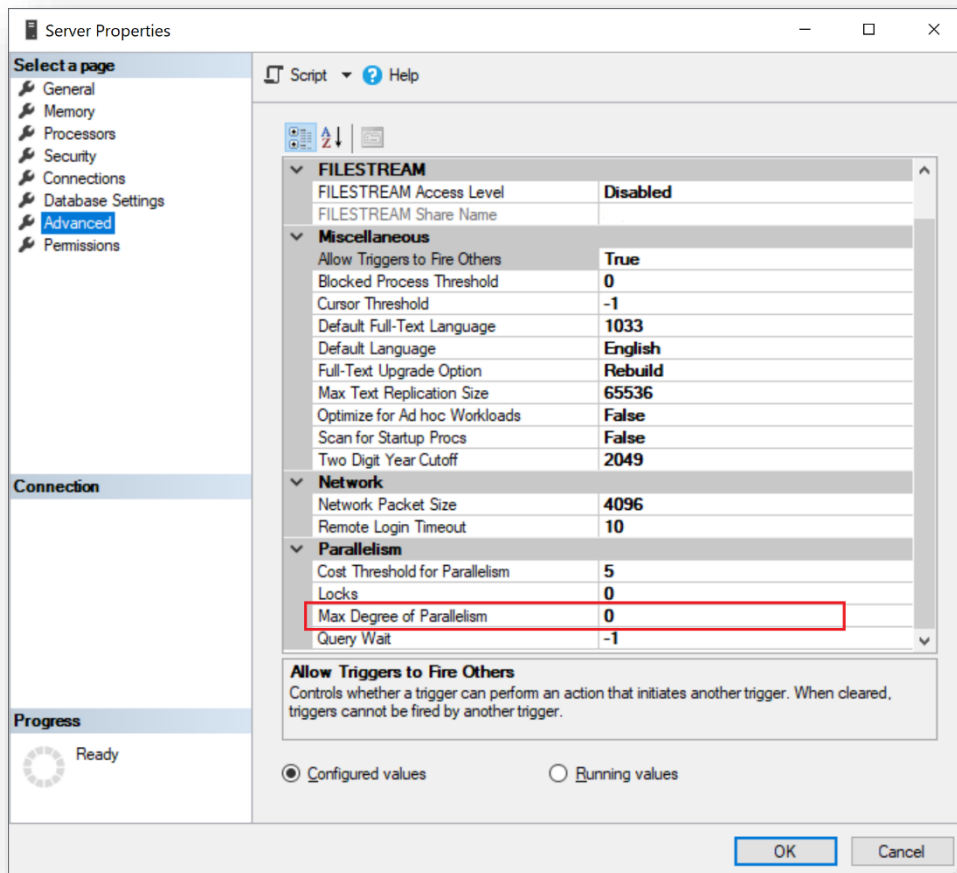
### How to configure it

To set the MaxDOP value to 0 for example by script:

```
EXEC sp_configure 'max degree of parallelism', 0
GO
RECONFIGURE
GO
```

To set the MaxDOP value to 0 by the management studio:

- Right Click on the SQL instance
- Property
- “Advanced” TAB
- “Parallelism” section



## Technical Reference

Should I change MaxDOP settings on my database?

<https://docs.microsoft.com/en-us/configmgr/core/understand/site-size-performance-faq#should-i-change-maxdop-settings-on-my-database>

Recommendations and guidelines for the "max degree of parallelism" configuration option in SQL Server: <https://support.microsoft.com/en-us/kb/2806535>

Introducing CXCONSUMER

[https://blogs.msdn.microsoft.com/sql\\_server\\_team/making-parallelism-waits-actionable/](https://blogs.msdn.microsoft.com/sql_server_team/making-parallelism-waits-actionable/)

## 1.2 Memory

### 1.2.1 Maximum Server Memory

The Maximum Server Memory parameter set the maximum Buffer Pool size of the SQL instance.

By default, the value is set to 2147483647, implying that the SQL engine will consume all the memory it can. The recommendation is to leave an amount of memory for the operating system and the other applications hosted on the same server. Note: it is not recommended to have other applications hosted on the same server.

### **How to check**

You can use the system view sys.configurations to check if your Maximum Server Memory parameter is properly set:

```
SELECT
    name,
    value
FROM
    sys.configurations
WHERE
    name = 'max server memory (MB)'
```

### **Why is that important**

By “system”, we mean here the Kernel system but also all kinds of agents like antivirus agents, monitoring agents, specific drivers, and so on... If SQL is allowed to consume the whole quantity of memory, and if it does, you may encounter some external memory pressure issues. Consequently, it may impact indirectly the performance of your MECM platform and the global health of your system.

It is recommended for a dedicated server to leave 2GB-4GB to the system, granting it enough memory to work with, reducing the risk of performance issues on the system later. For example, if your server has 32 GB physical memory, set the Maximum Server Memory to 28 GB.

### **Advanced considerations - Optional**

Additionally, note that there’s no specific advantage to apply the rule “keep 10% of memory for the system”. If your server has a huge amount of physical memory (let’s say 256GB), there’s no benefit to leave 25 GB for the system. That is too much, and does not make sense for a dedicated SQL server.

### **Recommended configuration for MECM**

	Total Memory <= 16 GB	Total memory > 16 GB & < 64? GB	Total Memory >= 64 GB?
SQL collocated	Leave at least 4 GB - 8 GB for the system/MECM	50% - 60% of total memory	65% - 75% of total memory

SQL not collocated	Leave at least 2 GB – 4 GB for the system	70% - 80% of total memory	80% - 95% of total memory
--------------------	---	---------------------------	---------------------------

### 1.2.2 Minimum Server Memory

The Minimum Server Memory sets the minimum amount of memory the buffer pool will be on the SQL instance. When SQL starts it doesn't automatically consume the minimum size, it just won't free any of the memory until the minimum amount has been reached.

#### **How to Check**

You can use the system view sys.configurations to determine if your Minimum Server Memory parameter is set:

```
SELECT
    name,
    value
FROM
    sys.configurations
WHERE
    name = 'min server memory (MB)'
```

#### **Recommended configuration**

While not as critical as ensuring max memory has been properly set, it is recommended to set this to something greater than 0. On servers with a lot of memory the recommendation is to set this to 8 GB. For servers with 16 GB or less a value of 4 GB to 8 GB is more appropriate.

#### **How to configure it**

Via script:

If configuring the minimum server memory value to 8 GB:

```
EXECUTE sp_configure 'min server memory (MB)', 8192;
GO
RECONFIGURE WITH OVERRIDE;
GO
```

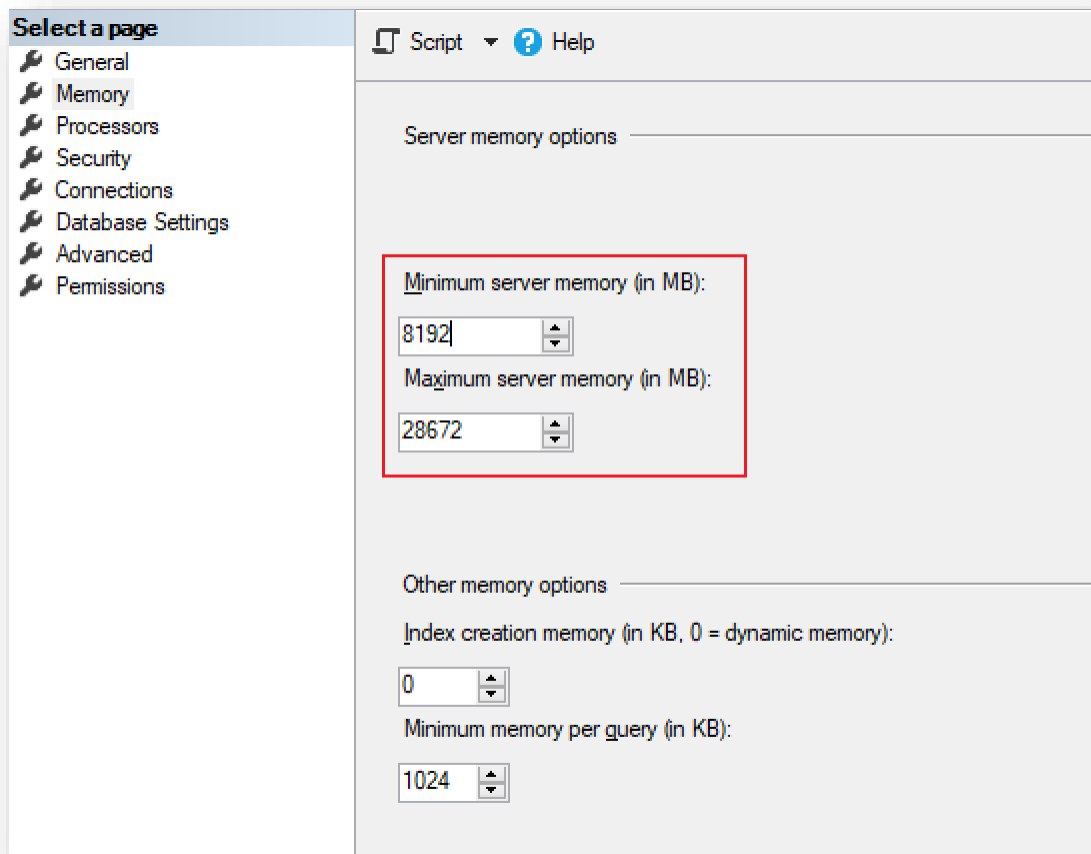
If configuring the minimum server memory value to 8 GB:

```
EXECUTE sp_configure 'max server memory (MB)', 28672;
GO
RECONFIGURE WITH OVERRIDE;
GO
```

Via SQL Server Management Studio:

To set the Maximum Server Memory value by the management studio:

- Right Click on the SQL instance
- Property
- “Memory” TAB
- “Server memory options” section



## **Technical Reference**

Effects of min and max server memory:

[https://technet.microsoft.com/en-us/library/ms180797\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms180797(v=sql.105).aspx)

## **1.3 Database Collation**

The database collation defined a combination of character set and sort order.

The collation will impact the alphabetical order, accent sensitivity, case sensitivity, width sensitivity, codepage and so on.

The SQL server-level collation is defined **during the instance installation phase**, that's why it's important to keep it in mind at the very beginning.

### **How to check it**

You can use the following statement to check what collation is defined once the installation completed:

```
SELECT SERVERPROPERTY ('Collation')
```

### **Why is that important**

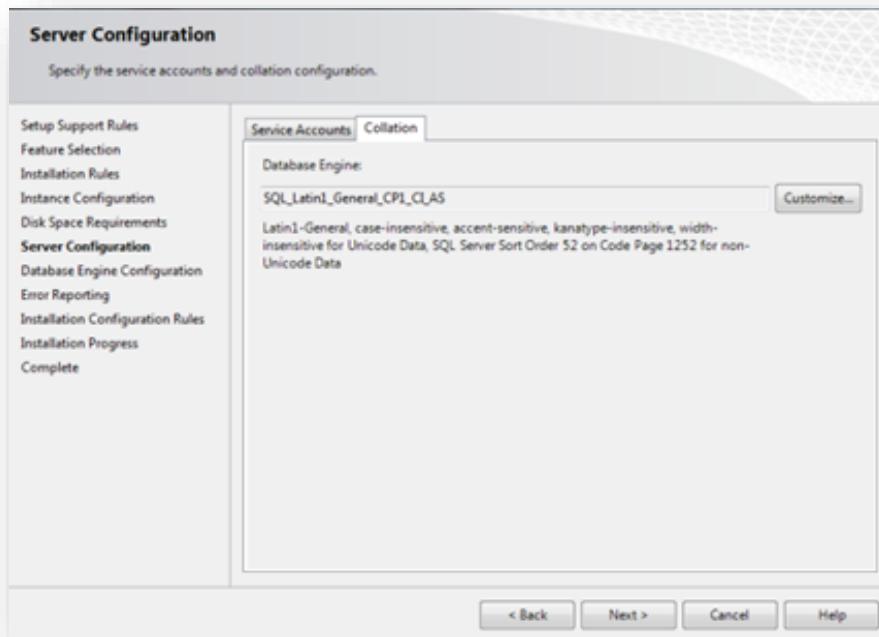
At each site, both the instances of SQL Server that is used for the site and the site database must use the following collation: **SQL\_Latin1\_General\_CP1\_CI\_AS**.

**Note:** Deviating from the require collation will result in Microsoft Endpoint Configuration Manager being in an unsupported state.

Microsoft Endpoint Configuration Manager supports two exceptions to this collation to meet standards that are defined in GB18030 for use in China. Chinese exceptions are described here:  
<https://technet.microsoft.com/library/hh738544.aspx>

### **How to configure it**

As mentioned, the collation name is defined during the installation setup:



## Note

It is also possible to change the server-level collation. This operation is complex and could be done to reconfigure a SQL Server instance before deploying MECM databases. It is important to keep in mind that the system databases (master, model, msdb, resource) will be rebuilt and results to a loss of every modifications (Settings, user objects, server-level objects)

To do so, please refer to the following document.

<https://docs.microsoft.com/en-us/sql/relational-databases/collations/set-or-change-the-server-collation?view=sql-server-ver15>

## 1.4 Additional instance settings

A number of SQL instance settings have to be enabled as well in a Microsoft Endpoint Configuration Manager context.

Normally they are enabled by default with the installation of MECM. You can find their description below.

### 1.4.1 SQL instance settings

A number of SQL instance settings need to be enabled as per Microsoft Endpoint Configuration Manager requirements.



Generally, they are enabled by default with the installation of MECM. You can find their description below.

### **CLR Enabled**

The Management Point Control Manager uses CLR to connect to the site database.

So it has to be turned on.

### **Max text repl size**

The max text repl size is especially required in a “replica Management Point” context.

If it is implemented, you can set the Max text repl size to 2 GB on both the site database server and replica server.

### **How to check it**

You can use the system view sys.configurations to check if these settings are properly configured :

```
SELECT
    name,
    value
FROM
    sys.configurations
WHERE
    name IN ('CLR Enabled', 'max text repl size (B)')
```

## **1.4.2 Trace Flag considerations**

A trace flag is a SQL Server startup parameter. Generally, its role is to bring a specific modification on the SQL engine. Even if the two following trace flags are especially significant for the Tempdb database, please note that when you enable a trace flag, it will always impact the whole SQL instance and then all the databases.

### **Recommendation**

Referring to the paragraph below dealing with the TempDB specifications ([2.1](#)), it is recommended to apply the Trace Flags 1117 and 1118 on the SQL Server instance.

#### **Important SQL Server 2016 consideration**

Beginning with SQL Server 2016, the behaviour of the trace flags 1117 and 1118 are overridden by the AUTOGROW\_ALL\_FILES and MIXED\_PAGE\_ALLOCATION options of the Alter Database statement, and are no longer required.

### **How to check**

You can use the following DBCC statement to check if these two trace flags are enabled.

```
DBCC TRACESTATUS (-1)
```

The expected result should return two records if these trace flags are enabled.

TraceFlag	Status	Global	Session
1117	1	1	0
1118	1	1	0

Note that you may have another additional trace flag enabled, as MECM requires the Trace flag 8295 by design for change tracking reason.

If you do not see this kind of results, and if you read "" instead, that means there's no trace flag enabled on your SQL environment.

### **Why is that important**

**Trace Flag 1117:** When a file in the filegroup meets the autogrow threshold, all files in the filegroup grow. This is especially useful for the Tempdb Database in order to apply the recommendation to keep all the files on the same size.

**Trace Flag 1118:** When enabled, this trace flag allocates all eight pages from the same extent when creating new objects, avoiding mixed extents, and then PFS and SGAM contention in a TempDB context. See "Advanced considerations - Optional" part below for further information.

### **Advanced considerations - Optional**

When a new object is created, by default, the first eight pages are allocated from different extents (mixed extents). Afterwards, when more pages are needed, those are allocated from that same extent (uniform extent). The SGAM page is used to track these mixed extents, so can quickly become a bottleneck when numerous mixed page allocations are occurring.

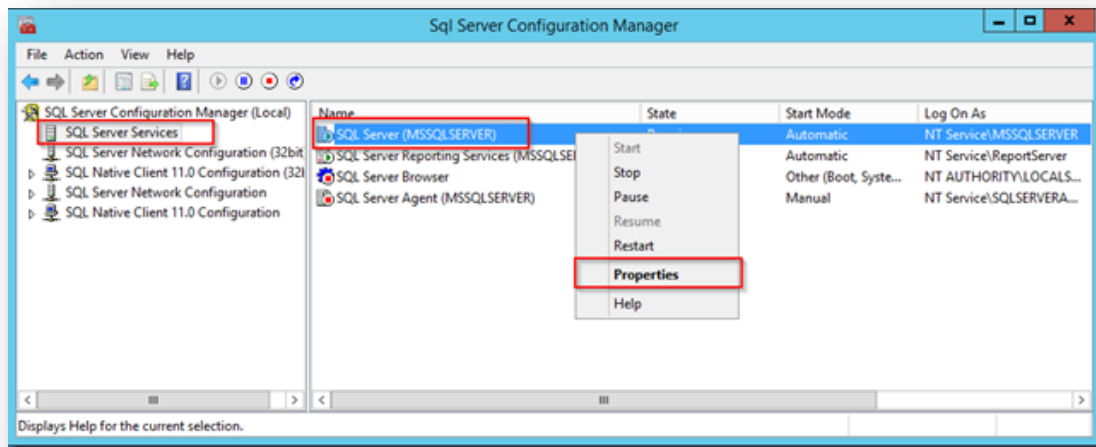
As part of allocating a page from the mixed extent, SQL Server must scan the Page Free Space (PFS) page to find out which mixed page is free to be allocated. The PFS page keeps track of free space available on every page, and each PFS page tracks about 8000 pages. Appropriate synchronization is maintained to make changes to the PFS and SGAM pages; and that can stall other modifiers for short periods.

When SQL Server searches for a mixed page to allocate, it always starts the scan on the same file and SGAM page. This results in intense contention on the SGAM page when several mixed page allocations are underway, which can cause some PFS and SGAM contentions, globally identified as "TempDB bottleneck".

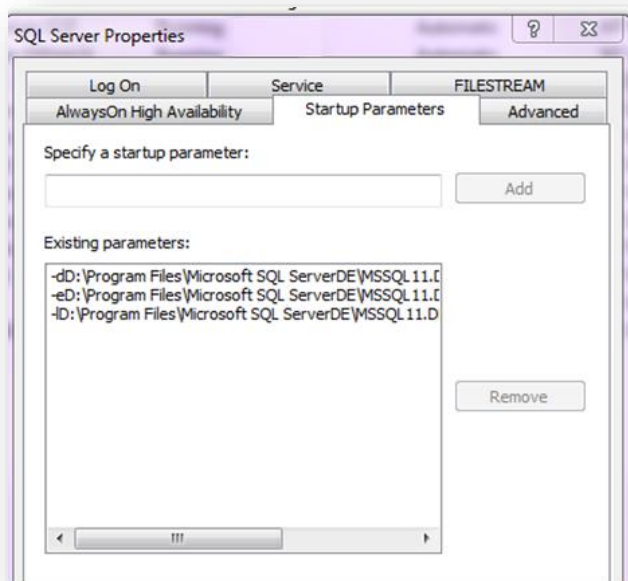
### **How to configure**

The trace flags are defined thru the SQL Microsoft Endpoint Configuration Manager.

The setting is done in the properties of the SQL Server engine service, as shown below:



In the property window, select the "Startup Parameters" tab, and add both -T1117 and -T1118 parameters in the top section to add them.



This action needs the SQL service to restart.

## Technical Reference

List of the existing trace flags: <https://docs.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-traceon-trace-flags-transact-sql>

Concurrency enhancements for the tempdb database : <https://support.microsoft.com/en-us/help/328551/concurrency-enhancements-for-the-tempdb-database>

### 1.4.3 Local Security Policies

Some local security accounts can have some performance impacts, and two of them are recommended to be applied to the SQL service engine account. Note that this can be performed automatically with a GPO (Group Policy Object).

#### **Recommendation**

It is highly recommended to apply the "Lock Page in Memory" and "Perform maintenance volume task" on the SQL service engine account.

#### **Why is that important**

The Lock Page In Memory (LPIM) privilege has to be used in 3 scenarios:

- To enable AWE feature on 32 bits platform... ok let's forget it, past is the past.
- Previous versions of Windows had issues with memory management that could cause large amounts of paging unnecessarily, requiring the use of LPIM to prevent a SQL Server instance from having a large percentage of its Working Set paged out to disk.

If you are running the latest version of SQL Server on the latest version of Windows Server, LPIM should not be necessary.

- LPIM is also required in order to support Large Page Allocation in SQL Server. Essentially, this setting allows for more efficient memory mapping as it leads to a smaller total number of allocations.

The Perform maintenance volume task privilege gives the Instant File Initialization (IFI) functionality to SQL Server when enabled. The benefits of IFI is not to fill the pages of the datafiles from all databases with zero in the following cases:

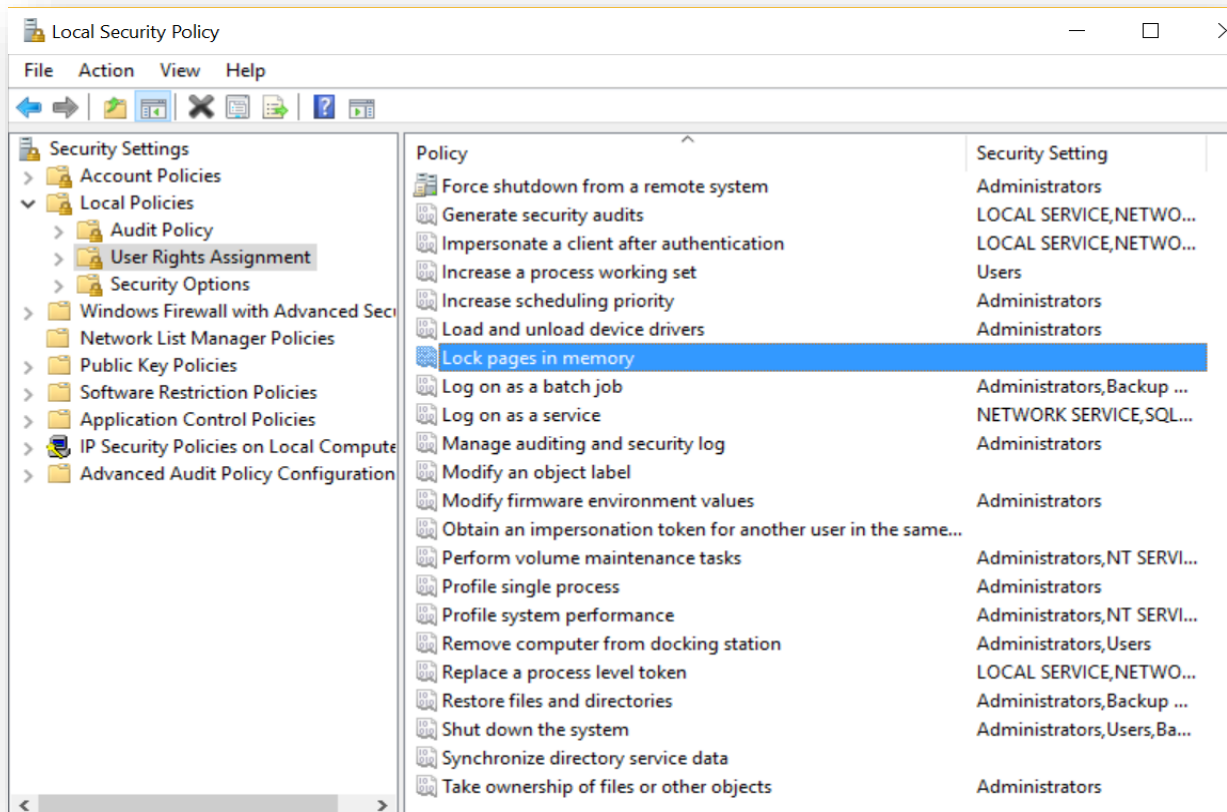
- Creating a database.
- Adding files to an existing database.
- Increasing the size of an existing file (including auto growth operations).
- Restoring a database or filegroup.

Notice that it does not apply on log files because virtual log files have to be formatted.

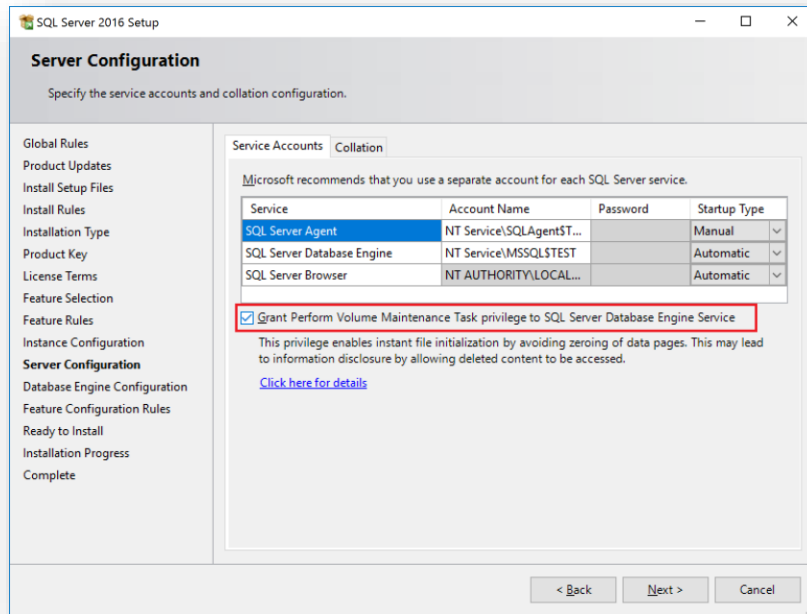
#### **How to check / configure**

These functionalities are available by applying local policies to SQL service accounts.

In the Local Security Policy management console, check and ensure that your SQL Service account is well associated to the two privileges "Lock Page in Memory" and "Perform maintenance volume task". If not, add it in the right click property windows.



Beginning with SQL Server 2016, Instant File Initialization can be applied directly from the SQL Server setup, by selecting **"Grant Perform Volume Maintenance Task to SQL Server Database Engine Service"**



## Technical Reference

Lock page in Memory and large pages:

<https://blogs.msdn.microsoft.com/psssql/2009/06/05/sql-server-and-large-pages-explained/>

Database Instant File Initialization:

<https://docs.microsoft.com/en-us/sql/relational-databases/databases/database-instant-file-initialization>

## 2. Database Settings

The Database settings are both relating to system databases and MECM databases. For each case it will be mentioned the specific scope.

### 2.1 TempDB and autogrowth settings

Tempdb is a system database used to hold temporary user objects, internal objects and version stores. For optimal performance, tempdb data files must be hosted on a dedicated volume.

The size and physical placement of the tempdb database can affect the performance of a system. For example, if the size that is defined for tempdb is too small, part of the system-processing load may be taken up with auto growing tempdb to the size required to support the workload every time you restart the instance of SQL Server. Multiple tempdb files can reduce contention on various global allocation structures by spreading activity over multiple physical files. This is particularly useful for those transactions using tempdb database, which spend a large percentage of time allocating and de-allocating tables.

## **Recommendation**

Multiple tempdb files can reduce contention on various global allocation structures by spreading activity over multiple physical files. This is particularly useful for those transactions using tempdb database, which spend a large percentage of time allocating and de-allocating tables.

As a general rule, if the number of logical processors is less than or equal to eight, use the same number of data files as logical processors. If the number of logical processors is greater than eight, use eight data files. If contention continues, increase the number of data files by multiples of four up to the number of logical processors until the contention is reduced to acceptable levels.

Preallocating space for these files can help to increase performance, preventing too frequent files expansion. But estimating the initial size of the tempdb is really difficult. However, the initial size of the tempdb file is too small (8Mb)

Once defined the number of files for your tempdb, there are two possibilities to set an initial size:

- If you have a dedicated volume for the tempdb database (recommended configuration), you can distribute equally 80-90% of your disk space between the tempdb files. As the space is dedicated for tempdb and there is no performance impact, this is the easiest way.
- Else you can choose a size around 20-30% of the total databases size and distribute it equally between the files.

Regarding to the autogrowth increment, you can start setting it to 512MB by file (which means, for a tempdb with 4 files, a total growth of 2GB). This setting may not be optimal for all workloads.

As SQL Server uses a proportional-fill algorithm, data files should be of equal size (and autogrowth as well)

As the better solution is to avoid automatic growths, in both cases, but especially in the second one, it is recommended to closely monitor the tempdb lifecycle when setting a new environment to adjust files size. Depending on the Microsoft Endpoint Configuration Manager activity, the growth of the CM scope or options used for the database maintenance, the file size can widely change. After some weeks, you may be able to have a better view on the tempdb size requirements and adjust the initial size accordingly.

## **How to check it**

You can use the system view `sys.database_files` to check if your **TempDB file** parameter is properly set:

```
USE tempdb
GO
SELECT
    name,
    type_desc AS File_type,
    size*8/1024 AS Initial_Size_MB,
    size*8/1024 AS Current_Size_MB,
    is_percent_growth,
    case is_percent_growth when 1 then size*8/1024*growth/100
    else growth*8/1024 end as Next_growth_MB
```

```
FROM
    sys.database_files
WHERE
    type = 0
```

### **Why is that important**

When the tempdb database is heavily used, SQL Server may experience contention when it tries to allocate pages. Depending on the degree of contention, this may cause queries and requests that involve tempdb to be unresponsive for short periods of time.

### **Advanced considerations - Optional**

When you examine the Dynamic Management Views `sys.dm_exec_request` or `sys.dm_os_waiting_tasks`, you observe that these requests or tasks are waiting for tempdb resources. You will notice that the wait type and wait resource point to LATCH waits on pages in tempdb. These pages might be of the format 2:1:1, 2:1:3, etc.

### **How to configure**

To set the correct value by script:

```
USE master;

GO

DECLARE
    @TempdbNumberOfFile int,
    @TempdbFileSize int,
    @TempdbFileGrowthMB int,
    @TempdbFilePath varchar(4000),
    @TempdbNewFileName sysname,
    @TempdbFileCount int,
    @SQL nvarchar(max) = N''

/* SET YOUR SETTINGS HERE */
SET @TempdbNumberOfFile = <VALUE>
SET @TempdbFileSize = <VALUE>
SET @TempdbFileGrowthMB = <VALUE>
SET @TempdbNewFileName = '<VALUE>'

/* Get current number of tempdb file */
SELECT @TempdbFileCount = COUNT(*) FROM tempdb.sys.database_files where type = 0

/* Get Tempdb File Path form first file*/
SELECT TOP 1 @TempdbFilePath = LEFT(physical_name, LEN(physical_name) - CHARINDEX('\',
REVERSE(physical_name)) + 1) from tempdb.sys.database_files WHERE type = 0 ORDER BY
file_id

WHILE (@TempdbFileCount < @TempdbNumberOfFile)
BEGIN
    SELECT @TempdbFileCount = @TempdbFileCount + 1
```



```

SELECT @SQL = @SQL + N'ALTER DATABASE [tempdb] ADD FILE (NAME = '' +
@TempdbNewFileName + '_' + cast(@TempdbFileCount as varchar(2)) + '', FILENAME = '' +
@TempdbFilePath + @TempdbNewFileName + '_' + cast(@TempdbFileCount as varchar(2)) +
'.ndf'', SIZE = ' + CAST(@TempdbFileSize as nvarchar(6)) + 'MB, FILEGROWTH = ' +
CAST(@TempdbFileGrowthMB as nvarchar(6)) + 'MB);'
END

/* Making sure to have equal size and autogrowth settings */
SELECT @SQL = @SQL + N'ALTER DATABASE [tempdb] MODIFY FILE (NAME = '' + name + '',
SIZE = ' + CAST(@TempdbFileSize as nvarchar(6)) + 'MB, FILEGROWTH = ' +
CAST(@TempdbFileGrowthMB as nvarchar(6)) + 'MB);'
FROM
    tempdb.sys.database_files
WHERE
    type = 0

EXEC sp_executesql @SQL
GO

```

Then replace all the values in the placeholders with your specific values

➤ To configure **TempDB File and Autogrowth** by the management studio:

- a. Expand Databases (by clicking the + (plus) sign)
- b. Expand System Databases (by clicking the + (plus) sign)
- c. Right-click the database tempdb, and in the pop-up menu, click Properties.
- d. On the Database Properties screen, in the Select a page section, click Files.
- e. Adjust initial file size for existing data files.
  - Select the column Initial Size (MB) for the tempdb data file and adjust it to the desired initial size value.
- f. Adjust AUTOGROWTH settings for the existing data files:

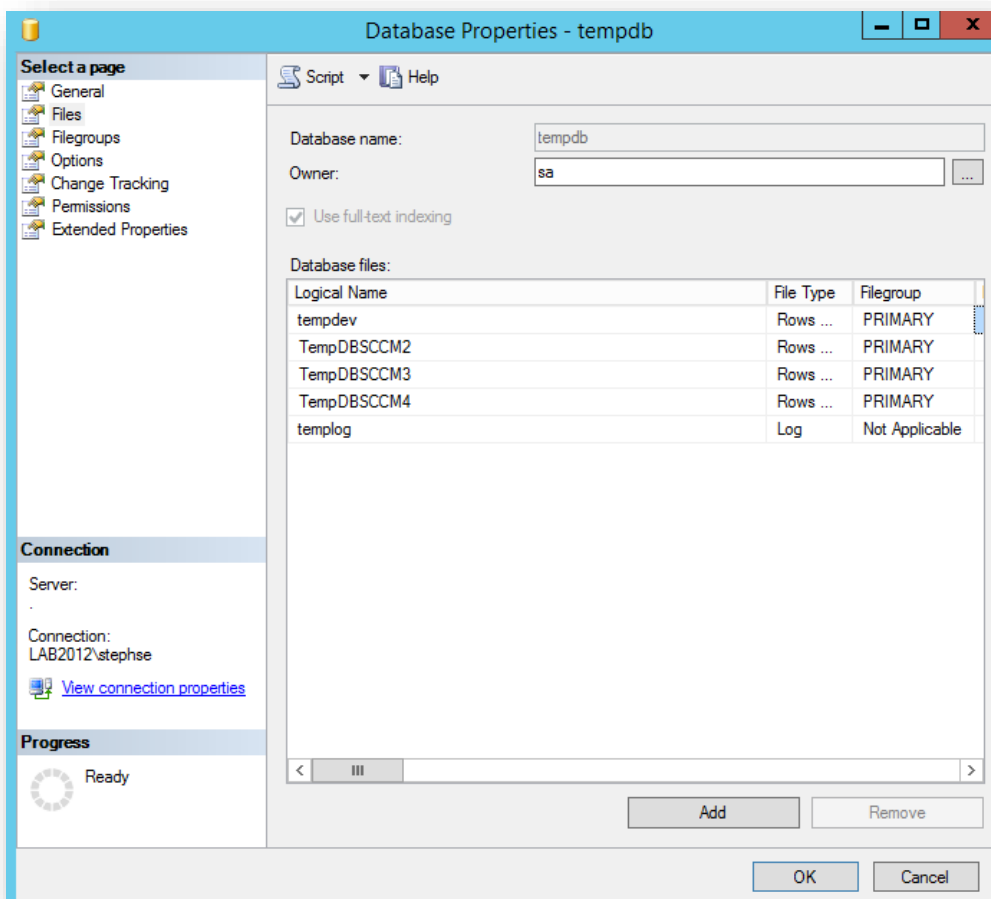
Click the button at AUTOGROWTH / Maxsize.

- On the Change AUTOGROWTH screen, click In Megabytes.
  - Change the value for megabytes to grow.
  - Click OK on the Change AUTOGROWTH screen.
- g. Repeat the Step e. and f. for each existing data file of the tempdb

➤ To **add** additional data files

- a. Click the ADD button
- b. Enter the Logical name for the new data file

- c. Adjust the initial file size to the value used in step e.
- d. Adjust AUTOGROWTH settings for the existing data files
  - Click the button at AUTOGROWTH / Maxsize.
  - On the Change AUTOGROWTH screen, click In Megabytes.
  - Change the value for megabytes to grow.
  - Click OK on the Change AUTOGROWTH screen
- e. Check path and adjust it if required
- f. Add a file name for the new data file
  - Repeat step h. for each data file which needs to be added
- j. Click OK.



## Technical Reference

[https://msdn.microsoft.com/en-us/library/ms175527\(v=SQL.105\).aspx](https://msdn.microsoft.com/en-us/library/ms175527(v=SQL.105).aspx)  
<https://support.microsoft.com/en-us/kb/2154845>

## 2.2 Database compatibility level

This section applies only if your SQL instance is on version 2014, 2016 or 2017.

SQL Server version	Supported compatibility level values	Recommended compatibility level for ConfigMgr	Recommended level for specific performance issues
SQL Server 2017	140, 130, 120, 110	<b>140</b>	110
SQL Server 2016	130, 120, 110	<b>130</b>	110
SQL Server 2014	120, 110	<b>110</b>	110

### Recommendation

Apply the right compatibility level of the databases regarding the SQL Server version.

### Why is that important

Since SQL 2014 introduce an enhancement of the engine base on the Cardinality Estimation, you may encounter performance issues with certain Microsoft Endpoint Configuration Manager queries in some environments, depending on variables such as site configuration, environment complexity, and the general state of load and performance of the SQL server.

If you upgrade or migrate a MECM database from a lower version than 2014 / 2016 to a newest one, you have to ensure that after the upgrade or migration your database is well adapted to the new environment. To do that, check the compatibility level of the upgraded / migrated database versus the expected recommendations mentioned above.

### How to check it

You can use the following SQL statement to check if the database compatibility level is well aligned.

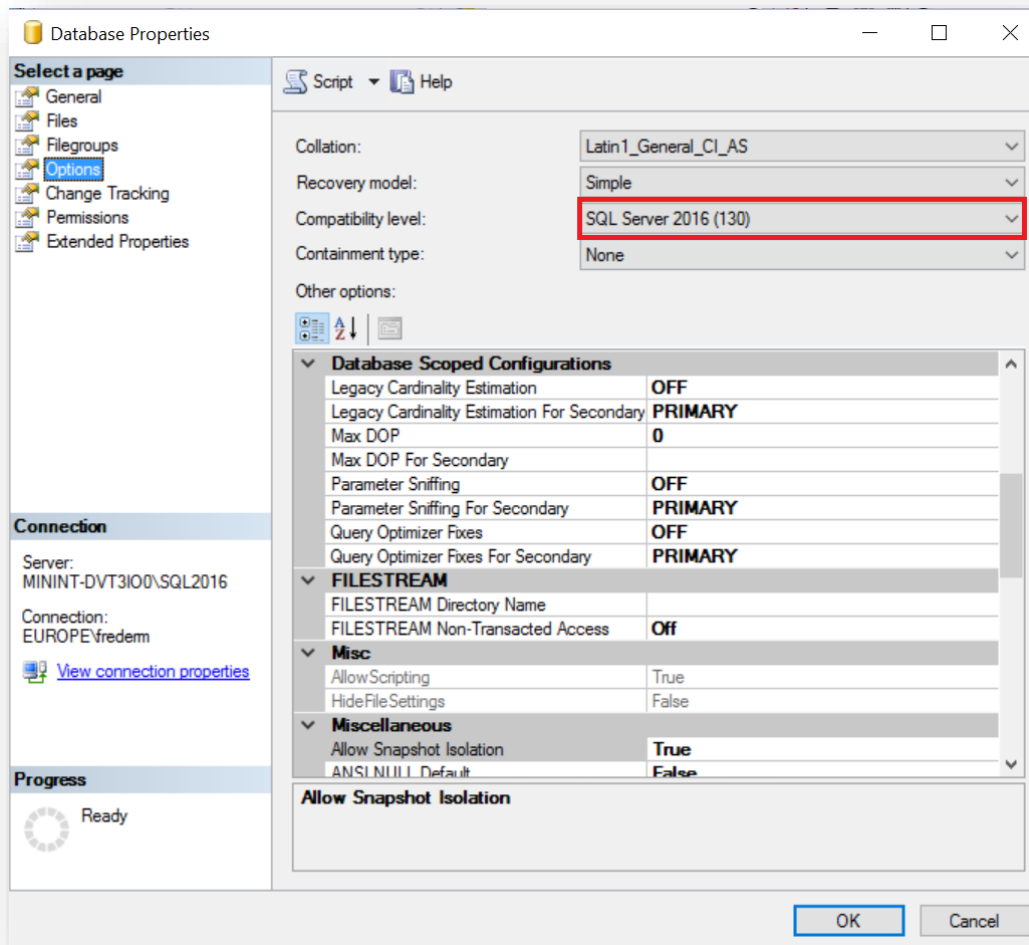
```
SELECT
    name,
    compatibility_level
FROM
    sys.databases
```

### How to configure it

To set the database compatibility level by script:

```
ALTER DATABASE <CM_DB> SET COMPATIBILITY_LEVEL = 110
```

To set the database compatibility level on the SSMS console:



### **Technical Reference**

<https://support.microsoft.com/en-us/help/3196320/sql-query-times-out-or-console-slow-on-certain-configuration-manager-database-queries>

## **2.3 Read committed snapshot isolation level**

### **Recommendation**

If the SQL application is suffering of multiple locks problems, the recommended solution is to enable the RCSI option on the database.

Setting this option requires an exclusive access on the CM\_DB database, meaning all the connections to the database will be terminated, but it does not require a restart of the service.

### **Why is that important**

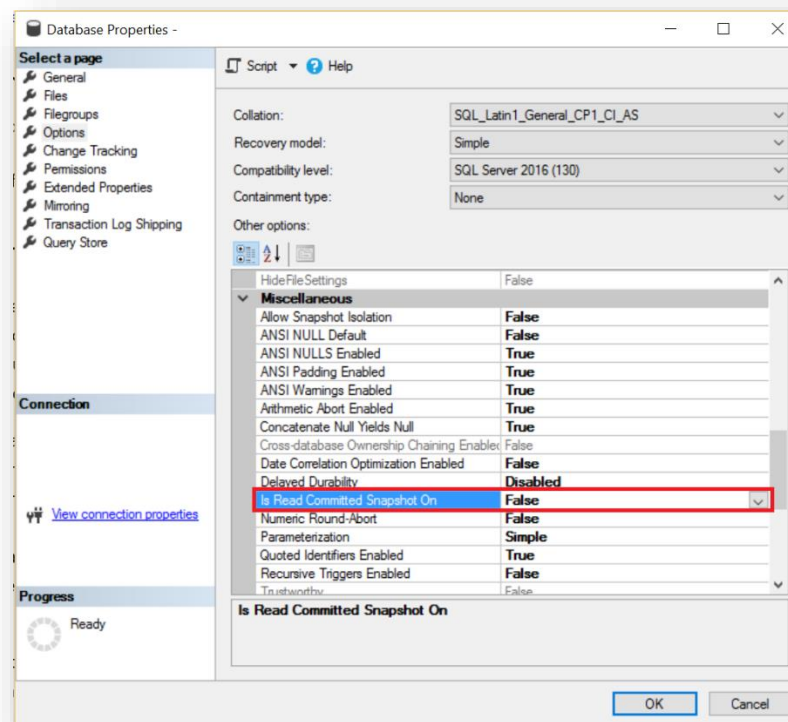
By default, all queries are executed in a read committed isolation level. It ensures atomic transactions and consistent reads. To make it simple, when writing or modifying data within a transaction, the data page where data is written may be locked until the end of the transaction, preventing other transactions to read this page. But when the other transactions need to read the same data page, they start waiting. The solution is to bypass the exclusive lock by enabling the read committed snapshot isolation level on the database. Then, any data page used in exclusive mode will be copied into a dedicated area in the TempDB named "Version Store". And then, any query will seek the version stored instead of being locked on the exclusive original resource, and it will return a consistent result immediately.

### How to check it

To know if the read committed snapshot isolation level is already enabled, you can query like following:

```
SELECT
    name,
    is_read_committed_snapshot_on
FROM
    sys.databases
```

Note that since SQL 2012 it's possible to check it as well with SSMS, in the database properties windows – Tab Options:



### How to configure it

To enable the read committed snapshot isolation level, first ensure that you have an exclusive access to the database. Then you can alter it:

```
ALTER DATABASE <CM_DB> SET READ_COMMITTED_SNAPSHOT ON WITH ROLLBACK IMMEDIATE
```

## Technical Reference

After the Microsoft Endpoint Configuration Manager SQL Site database is moved, you cannot create a Software Update package or application

<https://support.microsoft.com/en-us/help/3057073/after-the-system-center-2012-configmgr-sql-site-database-is-moved--you>

Snapshot isolation in SQL Server

<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/snapshot-isolation-in-sql-server>

## 3. Global Settings

### 3.1 Optimize for ad hoc workloads

The optimize for ad hoc workloads option is used to improve the efficiency of the plan cache for workloads that contain many single use ad hoc batches. When this option is set to 1, the Database Engine stores a small compiled plan stub in the plan cache when a batch is compiled for the first time, instead of the full compiled plan. This helps to relieve memory pressure by not allowing the plan cache to become filled with compiled plans that are not reused.

#### Recommendations

Avoid having a large number of single-use plans in the plan cache. A common cause of this problem is when the data types of query parameters is not consistently defined. This particularly applies to the length of strings but can apply to any data type that has a maxlength, a precision, or a scale. For example, if a parameter named @Greeting is passed as an nvarchar(10) on one call and an nvarchar(20) on the next call, separate plans are created for each parameter size. If a query has several parameters and they are not consistently defined when called, a large number of query plans could exist for each query. Plans could exist for each combination of query parameter data types and lengths that have been used.

If the number of single-use plans take a significant portion of SQL Server Database Engine memory in an OLTP server, and these plans are Ad-hoc plans, use this server option to decrease memory usage with these objects. To find the number of single-use cached plans, run the following query:

```
SELECT
    objtype,
    cacheobjtype,
    AVG(usecounts) AS Avg_UseCount,
```

```

SUM(refcounts) AS AllRefObjects,
SUM(CAST(size_in_bytes AS bigint))/1024/1024 AS Size_MB
FROM
    sys.dm_exec_cached_plans
WHERE
    objtype = 'Adhoc' AND usecounts = 1
GROUP BY
    objtype, cacheobjtype;

```

### Important

Setting the **optimize for ad hoc workloads** to 1 affects only new plans; plans that are already in the plan cache are unaffected. To affect already cached query plans immediately, the plan cache needs to be cleared using [ALTER DATABASE SCOPED CONFIGURATION CLEAR PROCEDURE\\_CACHE](#), or SQL Server has to restart.

### Verify current value:

```

SELECT
    name,
    value
FROM
    sys.configurations
WHERE
    name = 'optimize for ad hoc workloads'

```

1 = enabled

0 = disabled

### How to enable this setting:

```

EXEC sp_configure 'optimize for ad hoc workloads', 1
GO
RECONFIGURE
GO

```

### References

Optimize for ad hoc workloads Server Configuration Option

<https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/optimize-for-ad-hoc-workloads-server-configuration-option?view=sql-server-ver15>

Plan cache and optimizing for adhoc workloads

<https://www.sqlskills.com/blogs/kimberly/plan-cache-and-optimizing-for-adhoc-workloads/>

## 3.2 Forced Parameterization

You can override the default [simple parameterization](#) behaviour of SQL Server by specifying that all SELECT, INSERT, UPDATE, and DELETE statements in a database be parameterized, subject to certain limitations. Forced parameterization is enabled by setting the PARAMETERIZATION option to FORCED in the [ALTER DATABASE](#) statement. For large ad-hoc workloads, this setting may help reducing the CPU usage, by reducing query compilations and improving execution plan reuse. However, this is not a silver bullet. The MECM workload is heterogeneous and using this option may improve the performance in one use case context at the expense of another. If needed, you should work closely with your DBA team to monitor your SQL Server platform to be sure there are no side-effects.

How to verify setting:

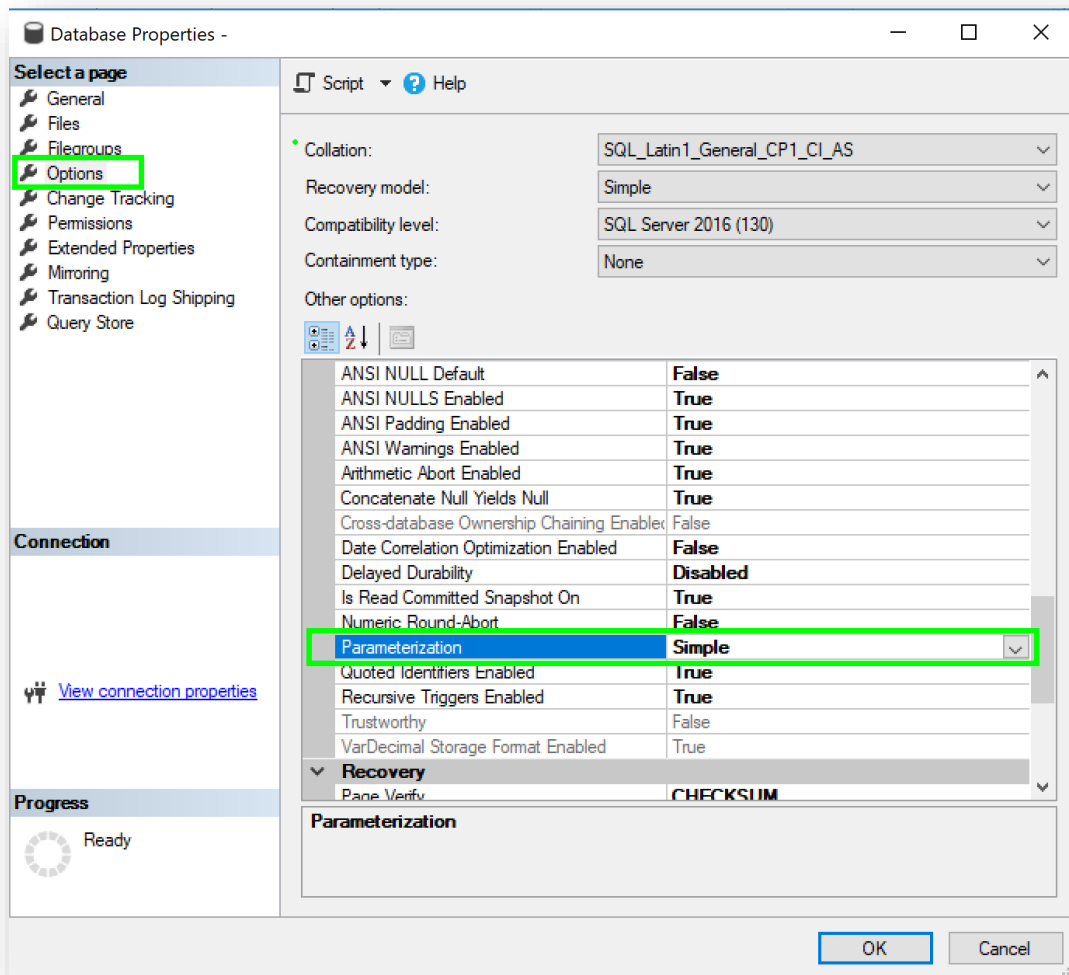
### T-SQL:

```
SELECT
    name,
    is_parameterization_forced
FROM
    sys.databases
WHERE
    name = '<database name>'
```

### GUI:

Right click on database name and select properties





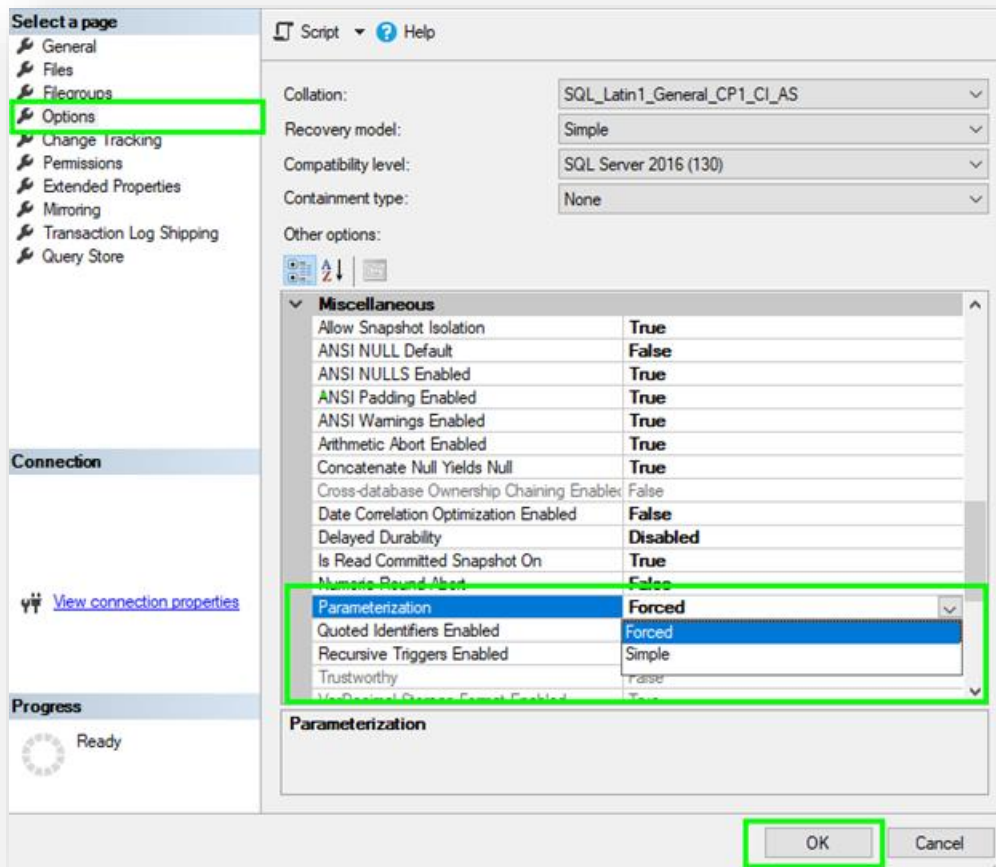
How to enable setting:

**T-SQL:**

```
USE [master]
GO
ALTER DATABASE [<<database name>>] SET PARAMETERIZATION FORCED WITH NO_WAIT
GO
```

**GUI:**

Right click on database name and select properties



## References

Forced parametrization

[https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms175037\(v=sql.105\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms175037(v=sql.105))

## 3.3 SQL Maintenance plan

A SQL Server maintenance plan is crucial both for the availability and consistency for databases, and for global SQL performance.

### Recommendation

There is no strict recommendation around the sequence of maintenance tasks, as it will depend on the available maintenance timeframe, especially if the MECM databases is hosted on a mutualized instance. However, if possible, you can consider these ones:

- Check Integrity
- Rebuild indexes
- Update Statistics
- Full backup
- Cleanup backups retention

These apply both on MECM databases and system databases (model, msdb, master).

The "when", "how" and "how often" will depend on the volume of the overall data and the available daily window.

### Integrity Check

An integrity check task will verify logical and physical integrity of all the objects of a specified database. It is typically operated by a DBCC CheckDB statement.

If any corruption occurs on the database, it will be identified. If it is, enough full backups have to be available to make a rollback possible if the corruption can't be fixed.

The integrity check task can both be done both via the maintenance plan task or via a script scheduled by job. You can refer for example to this kind of following script to apply it for all the databases of the SQL instance:

```
EXEC sp_MSforeachdb 'USE ?; DBCC CHECKDB (?); PRINT ''DB '' + DB_NAME() + '' Checked''  
+ CHAR(10) + CHAR(13)'
```

The last execution of the DBCC CheckDB Statement can be defined as show below:

```
DBCC DBINFO('NameOfTheDB') WITH TABLERESULTS
```

You can search into the resultset to find the value from the dbccLastKnownGood indicator. If the value refers to the 1st January 1900, that means that the database have never been checked.

ParentObject	Object	Field	VALUE
DBINFO STRUCTURE:	@0x000000ABA53FDAB0	<b>dbi_dbccLastKnownGood</b>	1900-01-01 00:00:00.

This information can be found as well with:

```
SELECT DATABASEPROPERTYEX (<DatabaseName>, 'LastGoodCheckDbTime')
```

The recommendation is to apply it ideally daily.

### **Note**

Performing a consistency check on a large database can be difficult as it is a time and resource consuming operation. Starting with SQL Server 2014 SP2, DBCC CHECKDB operation is parallelized by default using the max degree of parallelism setting of the instance. But it can be worth to override this value with the option WITH MAXDOP to increase the degree of parallelism for this operation and speed it up.

```
DBCC CHECKDB(<DatabaseName>) WITH MAXDOP = <number_of_processors>
```

If the consistency is still too long to run or impacting too much your production environment, you should consider changing your approach. Using PHYSICAL\_ONLY option will reduce the impact by

performing only a subset of the consistency check and can be used more frequently between full consistency checks. You can also consider performing a more granular consistency check, by using the consistency check underlying commands (please refer to the DBCC CHECKDB documentation). At last, you could restore the database on another environment to perform the consistency check.

## **Reference**

DBCC CHECKDB

<https://docs.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-checkdb-transact-sql?view=sql-server-ver15>

<https://blogs.msdn.microsoft.com/psssql/2016/02/25/sql-2016-it-just-runs-faster-dbcc-scales-7x-better/>

## **Rebuild Index**

The role of the rebuild index task is to avoid as much as possible the page fragmentation from indexes.

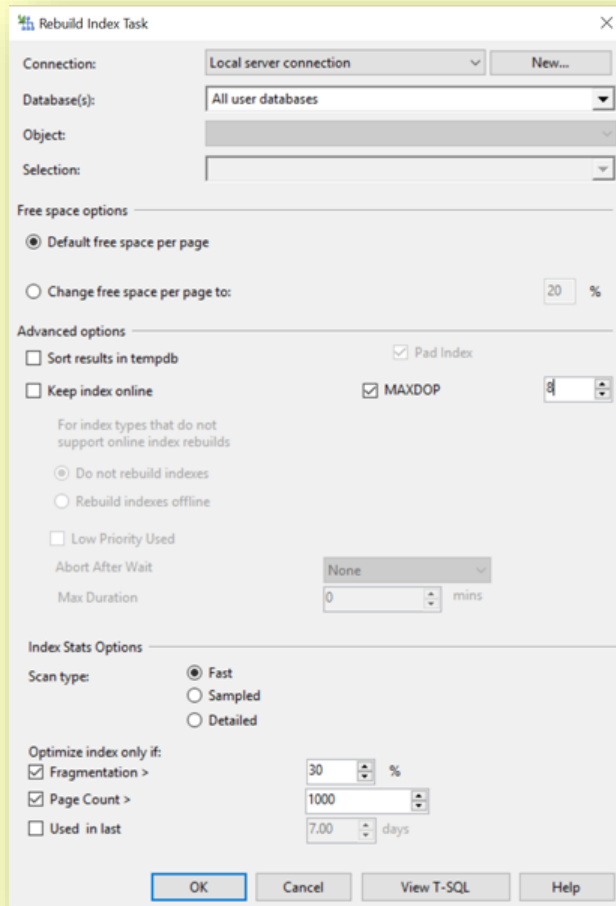
Note that a rebuild index task from the integrated SQL maintenance plan will rebuild ALL the indexes from a specified database, whatever is its size, and whatever is its fragmentation rate. So, this has no impact on small and medium databases, but on big ones it may spend some time.

It is possible to proceed a rebuild index task with a custom T-SQL script, for example to rebuild only the most fragmented indexes. Be aware that scripts available on internet blogs are supported by the authors of these scripts. It's not Microsoft role to troubleshoot if there is any "bug" in it.

### **SQL Server 2016 enhancement**

SQL 2016 involve the possibility to define a fragmentation rate threshold. Then it's possible to rebuild only fragmented enough indexes, and then only spend time on necessary objects. Additionally, it's also possible to specify a page count value to consider only big enough indexes.

Note that proposed values are default values and may be changed.



Even if the Max degree of parallelism is recommended to be set to 2 for ie for MECM databases, you can define here a value of 8 for the MaxDOP dedicated to the Rebuild Index task.

Finally, keep in mind that Rebuild Index task applies only on Clustered and Non-Clustered indexes, not on heaps. It's one of the many reasons to have a clustered index per table.

The Rebuild Index task can both be done both via the maintenance plan task or via a script scheduled by job. You can refer for example to the scripts proposed by the Tiger Team on his blogs for customized rebuilding index script:

<https://github.com/microsoft/tigertoolbox/tree/master/AdaptiveIndexDefrag>

As explained in the section above, the rebuild index task role is to avoid index fragmentation. So, you can check it with the following statement (per database) :

```
SELECT
    TOP 20
    OBJECT_NAME(si.object_id) AS TableName,
    si.name AS IndexName,
```

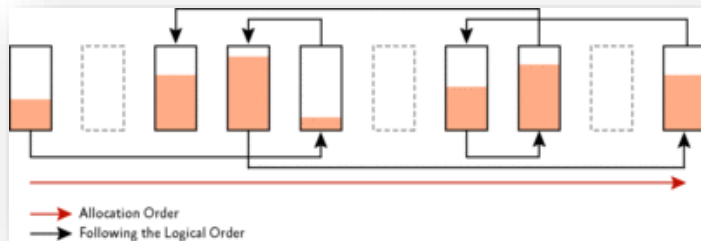
```

        si.index_id,
        ips.index_type_desc,
        ips.avg_fragmentation_in_percent,
        ips.page_count
FROM
    sys.indexes si
CROSS APPLY sys.dm_db_index_physical_stats(DB_ID(), si.object_id, si.index_id,
NULL, 'LIMITED') ips
WHERE
    ips.page_count > 1000
ORDER BY
    ips.avg_fragmentation_in_percent DESC

```

### **Advanced considerations - Optional**

You may also involve a reorganize index task, but it's not mandatory. Indeed the rebuild index task impacts data page fulfillment and the allocation page order while reorganize index task only impacts the allocation page order only.



Generally, if a rebuild index task is scheduled, it's enough to recover a global good enough fragmentation on indexes. Consider only reorganize operation on huge MECM data platform.

### **Update Statistics**

Index statistics and column statistics are both registered in the sys.stats system view. It's possible to retrieve update information thru the dm\_db\_stats\_properties function :

```

SELECT
    OBJECT_NAME(stat.object_id) AS TableName,
    stat.object_id,
    stat.name,
    stat.stats_id,
    sp.last_updated,
    sp.rows,
    sp.rows_sampled,
    sp.modification_counter
FROM
    sys.stats stat
CROSS APPLY sys.dm_db_stats_properties(stat.object_id, stat.stats_id) sp
ORDER BY
    TableName

```

The result indicates for each statistic of each table the weight of the object, the number of modifications on these statistics and the last update date.

It is crucial to keep up-to-date statistics for good performances. When an index is rebuilt, the associated statistic is automatically updated. However, custom, auto-created (`_WA_SYS_*`) and column statistics are not.

The Update Statistics task can be performed both via the maintenance plan task or via a script scheduled by job. You can refer for example to the following script to update all the indexes from all the databases of the SQL instance:

```
EXEC sp_MSforeachdb 'USE ?; EXEC sp_updatestats; PRINT ''Statistics updated on DB '' + DB_NAME() + CHAR(10) + CHAR(13)'
```

Note that `sp_updatestats` uses a default sampling rate when updating statistics and may sometimes lead to inaccurate statistics. As much as possible, update the statistics with `FULLSCAN`.

### Reference

#### UPDATE STATISTICS

<https://docs.microsoft.com/en-us/sql/t-sql/statements/update-statistics-transact-sql?view=sql-server-ver15>

#### sp\_updatestat

<https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-updatestats-transact-sql?view=sql-server-ver15>

### Full Backup Database

Once you ensured that your database is not corrupted, that the indexes are properly rebuilt and that the statistics are up-to-date, you can safely backup the database. MECM databases are expected to be on a simple recovery mode. So only full backup should be required.

Ensure that your backup retention is aligned with your integrity check frequency to be able to restore a valid database.

#### Backup your MECM Site with SQL maintenance plan

Configuration Manager can recover the site database from the Configuration Manager backup maintenance task or from a site database backup that you created with another process. For example, you can restore the site database from a backup that is created as part of a Microsoft SQL Server maintenance plan

In the same spirit, ensure that the "verify backup integrity" option is checked.

Finally, it's recommended to enable backup compression.

## SQL Backup to Azure Storage

It's possible since SQL Server 2012 SP1 CU2, and more commonly on SQL 2014 and SQL 2016 to backup a database directly to the Windows Azure Blob service. It can be done in the maintenance plan context by specifying URL as the backup destination.

As mentioned previously, the classical way to check if the backups have been run is to right click on the Jobs node and choose "jobs history". By querying the Backupset table in the MSDB database, you may review more accurate information, like the last backup date, the backup size, and so on..

```
SELECT TOP 50
    database_name,
    type,
    backup_start_date,
    backup_size,
    compressed_backup_size,
    user_name
FROM
    msdb.dbo.backupset
ORDER BY
    backup_start_date DESC
```

The backupset table registers only successful backups. So, if the resultset indicates that there is no backups since several days, it's definitively related to a backup issue.

### Cleanup backups retention

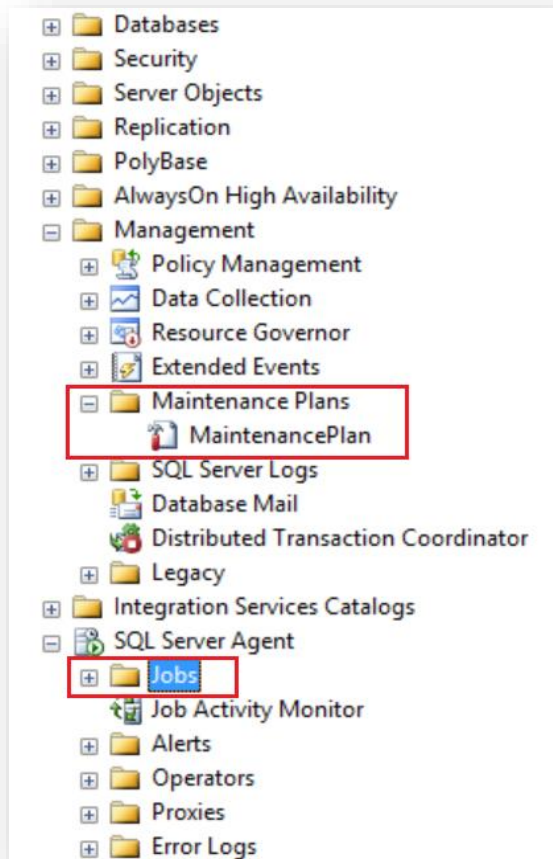
A cleanup retention task is simply a step to help you delete old backup files beyond a defined retention period. It is necessary to avoid volume saturation.

The most reliable way to apply it is still the cleanup task from the maintenance plan.

### How to check it

You will find existing maintenance plan in SQL Server Management Studio in maintenance plans node or amongst the list of existing jobs. Additionally, in the jobs node you can get the details of each execution by requesting the Jobs History.





Alternatively, you can check the impact and the accuracy of maintenance plan components detailed above.

### **Why is that important**

Maintenance Plan are crucial in SQL Server, for a couple of obvious reasons mentioned above. Database integrity check and full backups ensure the integrity and the recovery state of the databases, and the Rebuild index and Update Statistics tasks guaranty stable performances.

### **Technical Reference**

Top Tips for Effective Database Maintenance

<https://technet.microsoft.com/en-us/library/cc671165.aspx>

## **3.4 Ola.Hallengren SQL Server Maintenance**

A highly configurable and dynamic database maintenance script, which is highly recommended for all SQL Server maintenance implementations, namely:

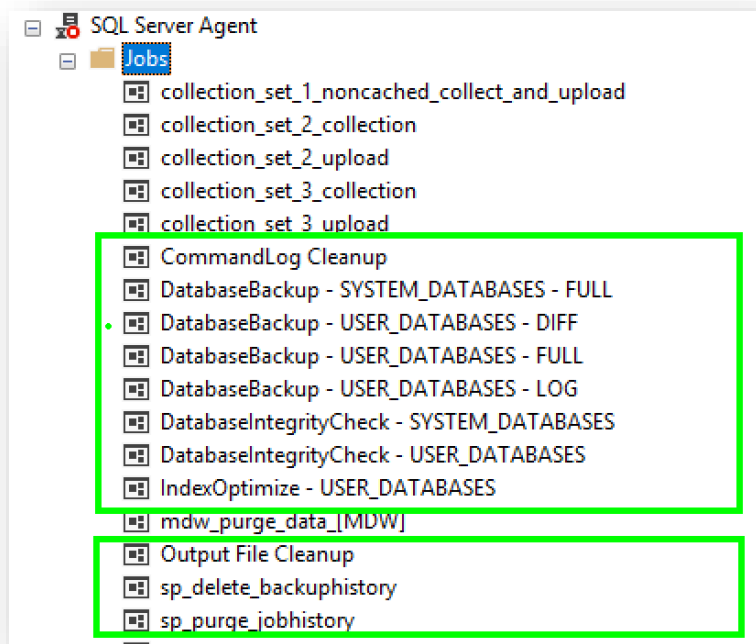
- User Database backups
- System Database Backups
- Database file retention and clean-ups
- Index and Statistical Maintenance
- Database Consistency Checks

## How to Configure

- Create a user database
  - This database will contain tables and stored procedure needed by the maintenance solution
- Download the maintenance script [here](#)
- Configure the script to reflect the following information:

```
SET @CreateJobs          = 'Y'          -- Specify whether jobs should be
created.
SET @BackupDirectory     = N'C:\Backup' -- Specify the backup root directory.
SET @CleanupTime         = NULL         -- Time in hours, after which backup
files are deleted. If no time is specified, then no backup files are deleted.
SET @OutputFileDirectory = NULL         -- Specify the output file directory. If
no directory is specified, then the SQL Server error log directory is used.
SET @LogToTable          = 'Y'          -- Log commands to a table.
```

- Configure SQL Server Jobs schedules
  - These maintenance schedules will be based on customer maintenance windows.



For further information on customization of maintenance jobs, please refer to the following [link](#).

## 3.5 Windows operating system configuration: power saving settings

### **Recommendation**

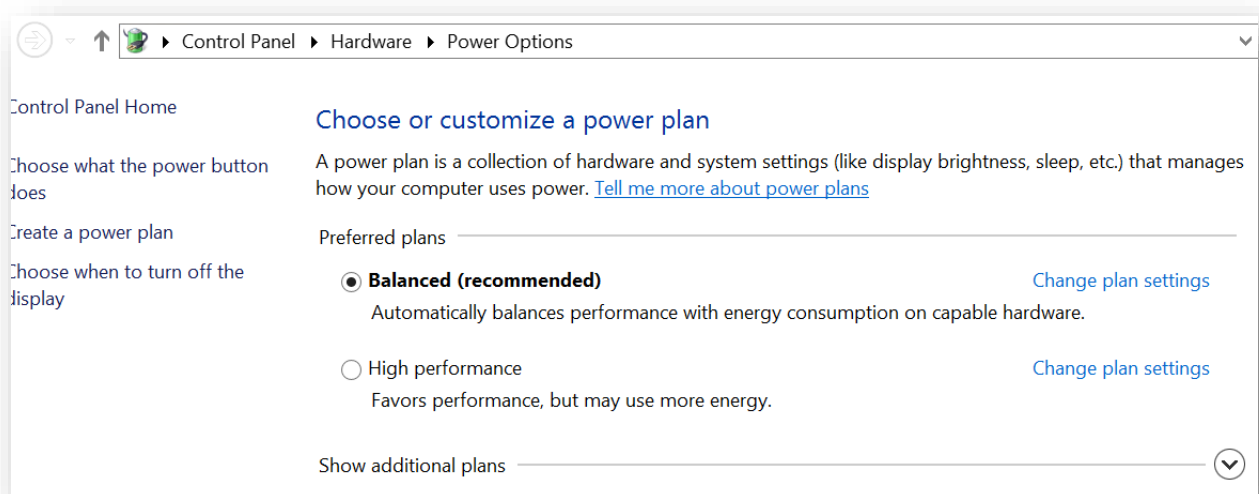
It is highly recommended to set the operating system power saving plan to **High Performance**.

### **Why it is important**

In Windows Server 2008 and later OS, the default power saving setting is set to Balanced, which means that components such as CPU and storage will be scaled back if the system is not busy. In some cases, this may result in performance degradation for SQL Server.

### **How to configure it**

The settings are located in the path Control Panel > Hardware > Power options



If Balanced setting is shown as recommended, it's meant from an energy consumption point of view. From global SQL performances the recommendation is to set it in High performance.

Note that you can manage power saving with GPO.

### **Technical references**

Slow Performance on Windows Server when using the "Balanced" Power Plan

<https://support.microsoft.com/en-us/help/2207548/slow-performance-on-windows-server-when-using-the-balanced-power-plan>

Managing power with Group Policy

<https://blogs.technet.microsoft.com/askds/2008/03/17/managing-power-with-group-policy-part-1-of-3/>

## 3.6 Antivirus exclusion

### **Recommendation**

It is highly recommended to set properly the antivirus exclusion if any is installed (which is recommended as well) on your SQL Server.

### **Why it is important**

If no exclusion is set into the antivirus software, any SQL activity may be caught by the antivirus and then the disk activity will be highly impacted with significant performance deterioration.

### **How to configure it**

Globally there's two ways to set the antivirus exclusions:

- By folder
- By file extension

The main targets have to be related to the following topics:

- SQL Server service binary file  
    %Root%\MSSQL\Binn\SQLServr.exe
- SQL Server data files
  - o \*.mdf, \*.ldf, \*.ndf
- SQL Server backup files
  - o \*.bak, \*.trn
- Full-Text catalog files
  - o Default instance: Program Files\Microsoft SQL Server\MSSQL\FTDATA
  - o Named instance: Program Files\Microsoft SQL Server\MSSQL\$instancename\FTDATA
- Trace files
  - o \*.trc
- SQL Query Files
  - o \*.sql
- SQL audit files
  - o \*.sqlaudit
- Extended Events files
  - o \*.xel, \*.xem
- Dump files
  - o \*.mdmp
- In-Memory OLTP files
  - o Files: xtp\_<t/p>\_<dbid>\_<objid>
  - o Extensions: \*.c, \*.dll, \*.obj, \*.out, \*.pdb, \*.xml
- Checkpoint and delta files
- Filestream data files

- Files in replication snapshot data
- In a failover cluster environment
  - o Q:\ (Quorum drive)
  - o C:\Windows\Cluster

### **Technical references**

How to choose antivirus software to run on computers that are running SQL Server

<https://support.microsoft.com/en-us/help/309422/how-to-choose-antivirus-software-to-run-on-computers-that-are-running>

Recommended antivirus exclusions for Configuration Manager 2012 and Current Branch Site Servers, Site Systems, and Clients

<https://support.microsoft.com/en-us/help/327453/recommended-antivirus-exclusions-for-configuration-manager-2012-and-cu>

## 3.7 Change Tracking

MECM uses SQL Change Tracking for replication when your hierarchy includes a Central Administration Site (CAS). SQL takes care of cleaning up change tracking automatically. However, sometimes SQL just doesn't do the cleanup or do it well enough.

### **Why is this important?**

If SQL doesn't cleanup the old change tracking records within the change tracking tables performance can suffer severely. Replication will suffer and thus the problem will compound.

### **Recommendation**

It is recommended to regularly check if the cleanup is happening on the change tracking tables. SQL recommends doing that by using a stored procedure named "sp\_flush\_commit\_table\_on\_demand". However, there is a known issue with this procedure which means it may not work correctly at times. To check these tables without this procedure a "Dedicated Admin Connection" (DAC) must be used. For MECM there is a stored procedure within the database that can check and clean these tables for you (as long as you are connected to the DAC). This stored procedure is named "spDiagChangeTracking" and takes a few parameters:

- @CleanupChangeTracking
  - o 0 = check the status of change tracking but do not cleanup if cleanup is required (this is the default)
  - o 1 = check the status of change tracking and perform the cleanup of old records
- @CheckSideTableMinTime

- 0 = do not find the oldest date in each side table (this is the default and recommended)
- 1 = find the oldest date in each side table – this can cause the script to take much longer and is not recommended
- @RowsToDeletePerIteration
  - Default is 5,000; recommendation is to increase this to something greater – for larger environments this can be set to 500,000 or higher

Information will be returned after the completion of the script. However, if you wish to see the status of the script while it's running you can open a different query window and run:

```
SELECT * FROM ##SPDiagCleanupCT
```

A good recommendation for this procedure is to run the following command (in a DAC) every week:

```
EXECUTE dbo.spDiagChangeTracking 1, 0, 500000
```

## Reference

Change Tracking (aka Syscommittab) Issues and Cleanup

<https://sqlbenjamin.wordpress.com/2017/07/04/change-tracking-aka-syscommittab-issues-and-cleanup-part-1/>

## 3.8 Virtualization

In general, Microsoft does not make any recommendation on non-Microsoft virtualization environments.

It is recommended to follow the configuration best practices published by your solution provider.

## Reference

Support policy for Microsoft software that runs on non-Microsoft hardware virtualization software

<https://support.microsoft.com/en-us/help/897615/support-policy-for-microsoft-software-that-runs-on-non-microsoft-hardw>

Support policy for Microsoft SQL Server products that are running in a hardware virtualization environment

<https://support.microsoft.com/en-us/help/956893/support-policy-for-microsoft-sql-server-products-that-are-running-in-a>

Best Practices for Virtualizing and Managing SQL Server

[http://download.microsoft.com/download/6/1/d/61dde9b6-ab46-48ca-8380-d7714c9cb1ab/best\\_practices\\_for\\_virtualizing\\_and\\_managing\\_sql\\_server\\_2012.pdf](http://download.microsoft.com/download/6/1/d/61dde9b6-ab46-48ca-8380-d7714c9cb1ab/best_practices_for_virtualizing_and_managing_sql_server_2012.pdf)

Performance Tuning Hyper-V Servers

<https://docs.microsoft.com/en-us/windows-server/administration/performance-tuning/role/hyper-v-server/>

Architecting Microsoft SQL Server on VMware vSphere®

<http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-best-practices-guide.pdf>

VMware Setup for Failover Clustering and Microsoft Cluster Service

<https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-672-setup-mscs.pdf>

### 3.9 Virtual Machines sizing

Microsoft published insights regarding VM configuration for high performance.

Configuring optimal disk performance is often viewed as an obscure part from data administrators. A best practice that is essential yet often overlooked is disk partition alignment. An understanding of disk performance best practices can result in significant improvements. Some of the many factors that affect disk I/O performance include the number, size, and speed of disks.

Volume alignment, commonly referred to as sector alignment, should be performed on the file system (NTFS) whenever a volume is created on a RAID device. Failure to do so can lead to significant performance degradation; these are most commonly the result of partition misalignment with stripe unit boundaries. This can also lead to hardware cache misalignment, resulting in inefficient utilization of the array cache. When formatting the partition that will be used for SQL Server data files, it is recommended that you use a 64-KB allocation unit size (that is, 65,536 bytes) for data, logs, and tempdb. Be aware however, that using allocation unit sizes greater than 4 KB results in the inability to use NTFS compression on the volume. While SQL Server does support read-only data on compressed volumes, it is not recommended.

#### **Technical references**

<https://docs.microsoft.com/en-us/sccm/core/plan-design/configs/recommended-hardware>

[https://technet.microsoft.com/en-us/library/dd758814\(v=sql.100\).aspx](https://technet.microsoft.com/en-us/library/dd758814(v=sql.100).aspx)

#### **Performance**

[Azure VM size and type](#), Azure VM disks (premium storage is recommended, especially for SQL Server), networking latency, and speed are the most important areas.

#### **So, tell me more about Azure virtual machines; what size VMs should I use?**

In general, your compute power (CPU and Memory) need to meet the [recommended hardware for System Center Configuration Manager](#). But there are some differences between regular computer hardware and Azure VMs, especially when it comes to the disks these VMs use. What size VMs you use depends on the size of your environment but here are some recommendations:

- For production deployments of any significant size we recommend “S” class Azure VMs. This is because they can leverage Premium Storage disks. Non “S” class VMs use blob storage and in

general will not meet the performance requirements necessary for an acceptable production experience.

- Multiple Premium Storage disks should be used for higher scale, and striped in the Windows Disk Management console for maximum IOPS.
- We recommend using better or multiple premium disks during your initial site deployment (like P30 instead of P20, and 2xP30 in a striped volume instead of 1xP30). Then, if your site later needs to ramp up in VM size due to additional load, you can take advantage of the additional CPU and memory that a larger VM size provides. You will also have disks already in place that can take advantage of the additional IOPS throughput that the larger VM size allows.

The following tables list the initial suggested disk counts to utilize at primary and central administration sites for various size installations:

#### Co-located site database

Primary or central administration site with the site database on the site server:

##### Desktop Clients Recommended VM size Recommended Disks

Up to 25k	DS4_V2	2xP30 (striped)
25k to 50k	DS13_V2	2xP30 (striped)
50k to 100k	DS14_V2	3xP30 (striped)

#### Remote site database

Primary or central administration site with the site database on a remote server:

##### Desktop Clients Recommended VM size Recommended Disks

Up to 25k	Site server: F4S	Site server: 1xP30
	Database server: DS12_V2	Database server: 2xP30 (striped)
25k to 50k	Site server: F4S	Site server: 1xP30
	Database server: DS13_V2	Database server: 2xP30 (striped)



## Desktop Clients Recommended VM size

## Recommended Disks

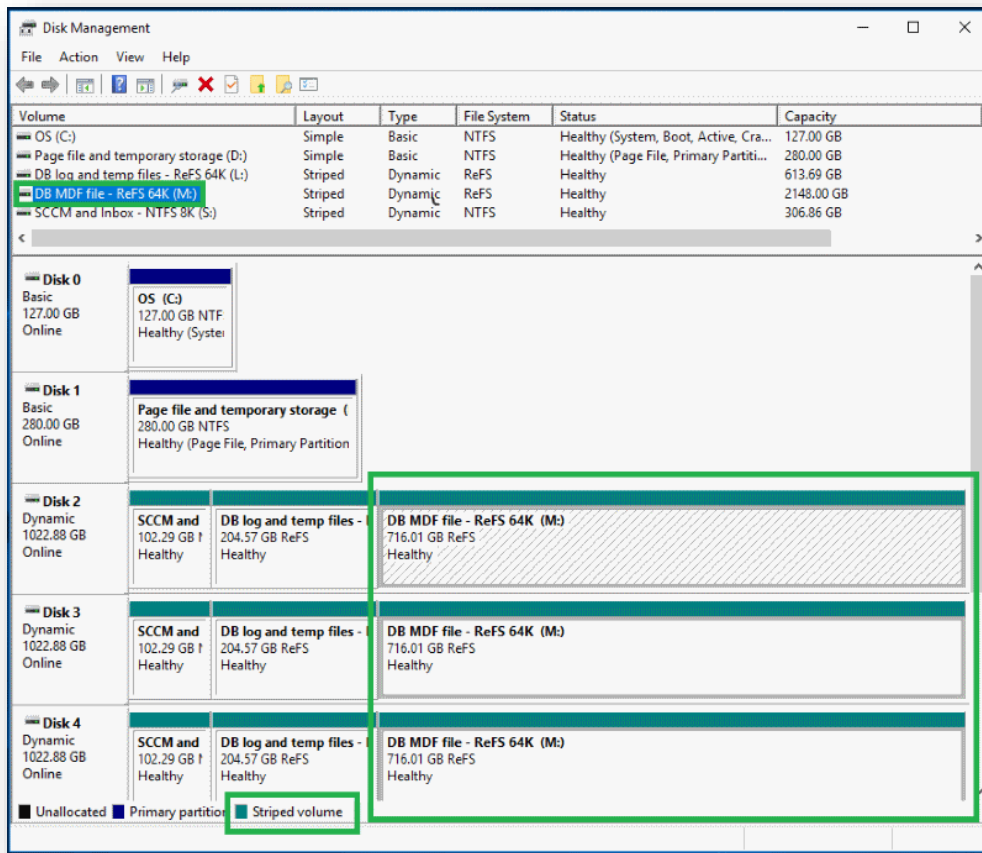
50k to 100k

Site server: F8S

Site server: 2xP30 (striped)

Database server: DS14\_V2 Database server: 3xP30 (striped)

The following shows an example configuration for 50k to 100k clients on DS14\_V2 with 3xP30 disks in a striped volume with separate logical volumes for the Microsoft Endpoint Configuration Manager install and database files:



### 3.10 Microsoft Endpoint Configuration Manager (current branch) site sizing guidance

Microsoft has documented online are the maximum supported scalability limits (<https://docs.microsoft.com/en-us/sccm/core/plan-design/configs/size-and-scale-numbers>) as well as the hardware guidelines (<https://docs.microsoft.com/sccm/core/plan-design/configs/recommendedhardware>) needed to run sites at the very largest environment sizes. This document gives supplemental performance guidance for Microsoft Endpoint Configuration Manager environments of all sizes.

Specific details and test results generated by the product team are supplied in this document, especially around the requirements for the largest contributor of performance bottlenecks in Microsoft Endpoint Configuration Manager environments, that of the disk subsystem. Suggested disk IOPS requirements are provided for various size environments as well as a documented repeatable method that you can use to reproduce these numbers in your own environments and on your own hardware. This allows for more accurate estimates of the hardware required to deploy Microsoft Endpoint Configuration Manager.

They also address frequently asked questions for common performance issues.

### **Technical references**

<https://cloudblogs.microsoft.com/enterprisemobility/2018/02/22/public-preview-configuration-manager-performance-guidance-for-environments-of-all-sizes/>

<https://gallery.technet.microsoft.com/Configuration-Manager-ba55428e>

## **3.11 SQL Server Reporting Services**

SQL Server Reporting Services (SSRS) is a component of the SQL Server BI Suite used by Microsoft Endpoint Configuration Manager.

When used in the Microsoft Endpoint Configuration Manager context, it will not require any specific maintenance.

However, some actions should be performed to avoid any unpleasant situation.

SSRS relies on two databases, ReportServer and ReportServerTempDB (default name). These two databases can be hosted on any available SQL Server instance.

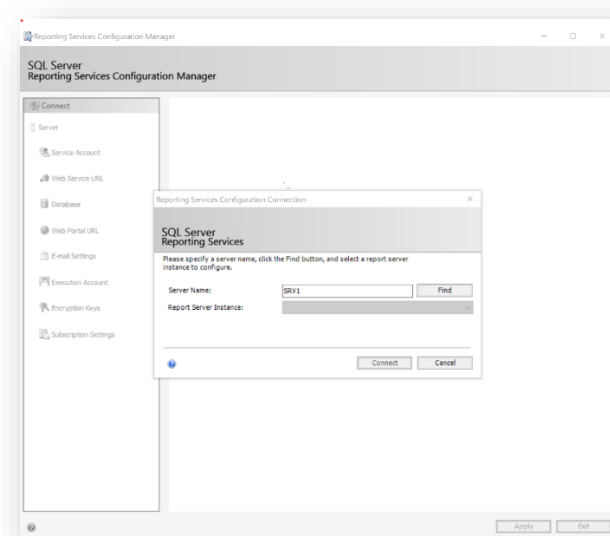
### **Encryption Key**

Once installed, the first thing to do is to perform a backup of the SSRS encryption key. This key is used to encrypt sensitive data recorded in the database. This key will be necessary if SSRS is moved or reinstalled to access the sensitive data in the existing databases. Without this key, the sensitive data could not be read.

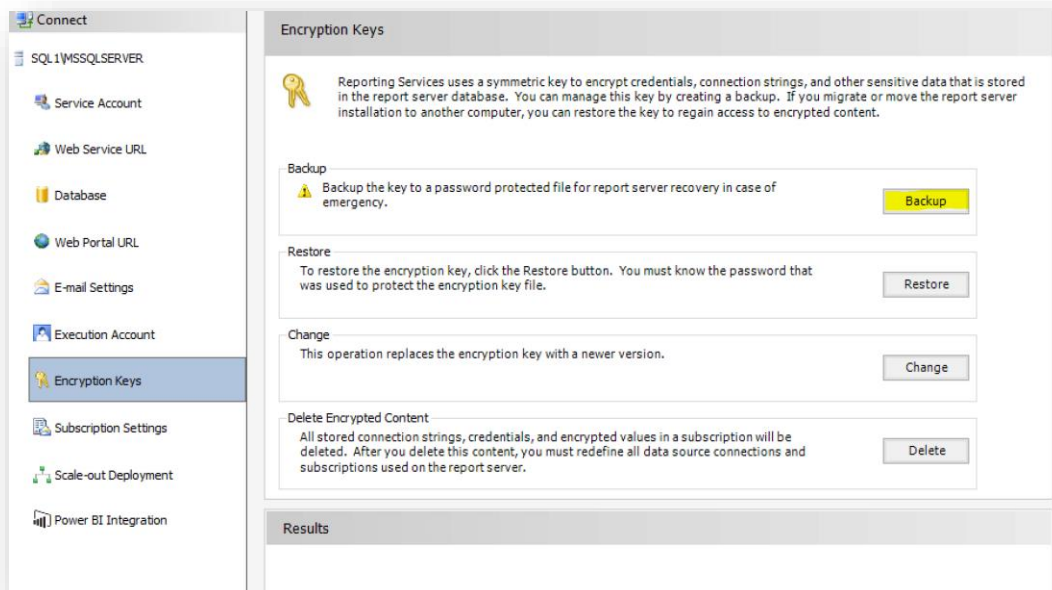
To perform the encryption key backup:

Via the GUI

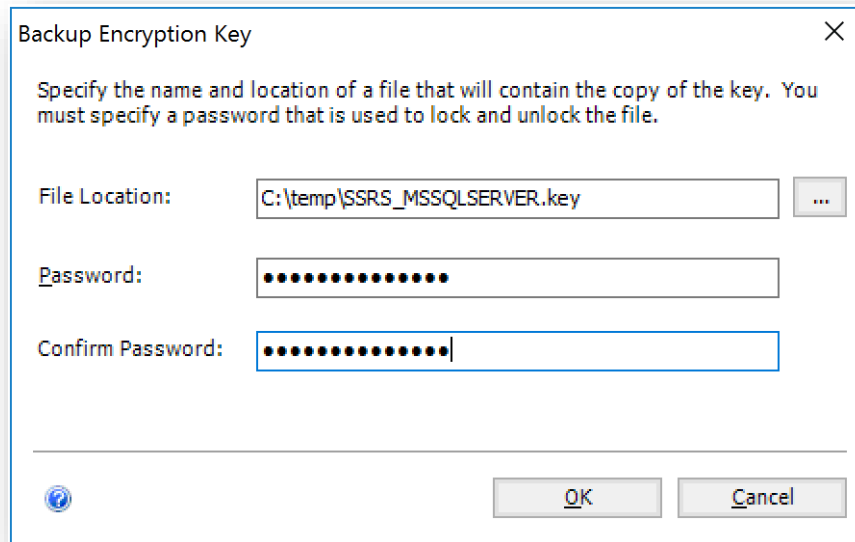
Open the Reporting Services Configuration Manager and connect to your SSRS instance.



Select the Encryption Keys menu, and click on Backup



Select a location and enter a password. This password will be used to encrypt the key backup file.



### Via Command Line

You can also perform the backup using the following command:

***rskeymgmt -e -f <FileName> -p <StrongPassword> -i <InstanceName>***

FileName: The path and filename of the backup file

StrongPassword: The password used to encrypt the key backup file.

InstanceName: The SSRS instance name

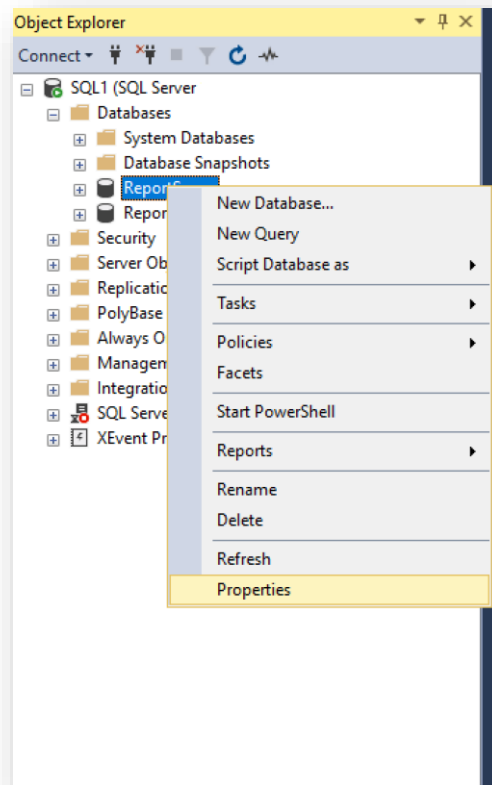
### Recovery Model

By default, ReportServer is configured to **Full recovery model**. This setting can be changed to avoid a large growth of the transaction log file.

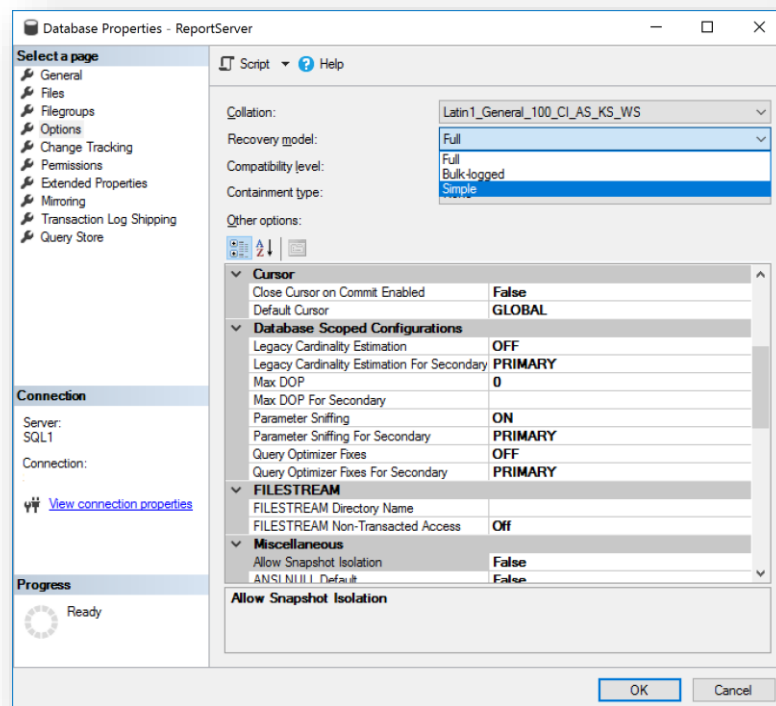
In order to change the recovery model to **Simple model**, you can proceed as follow:

### Via the GUI

Right-Click on the ReportServer database and select “Properties”.



In the Options menu, select **Simple** in the Recovery model dropdown list. Then click **OK**.



### Via Command line

Using SQL Server Management Studio, open a new query on the SQL Server instance hosting the database and execute the following command:

```
USE [master]
GO
ALTER DATABASE [ReportServer] SET RECOVERY SIMPLE WITH NO_WAIT
```

**NB:** the previous examples is based on the default database name (ReportServer)

### Maintenance

As any database, the Reporting Services databases require maintenance, such as backup, statistics update, index maintenance or consistency check.

Please refer to the maintenance paragraphs [3.3](#) and [3.4](#)

## 3.12 SCOM

The original monitoring pack will allow you to monitor the health of Microsoft Microsoft Endpoint Configuration Manager by monitoring general health; data replication between Microsoft Endpoint Configuration Manager sites; server and service availability; SQL Server configurations; Backup and recovery; backlog monitoring; software update synchronization; and other server role configuration. This update will extend the capabilities to monitor availability, performance, and health of the Microsoft Intune connector site system role for companies who integrate Configuration Manager and Microsoft Intune in a hybrid environment.

### **Updated Feature Summary**

This release of the monitoring pack delivers improved capabilities including the following:

- Monitors availability status of the Intune connector
  - SMS\_DMP\_DOWNLOADER – SMS\_Executive component
  - SMS\_DMP\_UPLOADER – SMS\_Executive component
  - SMS\_CLOUD\_USERSYNC – SMS\_Executive component
  - SMS\_OUTGOING\_CONTENT\_MANAGER – SMS\_Executive component
- Monitors hman.box for backlogs that exceed the threshold
  - Monitor inbox - hman.box performance counter
- Monitors availability status of the Microsoft Intune service
  - Microsoft Intune service offline
  - Local network issue prevents DMP connector from reaching Intune service
  - DMP connector does not sync with the Intune service

### **Link to download Management Pack for Configuration Manager**

[https://www.microsoft.com/en-us/download/details.aspx?id=34709&WT.mc\\_id=rss\\_alldownloads\\_all](https://www.microsoft.com/en-us/download/details.aspx?id=34709&WT.mc_id=rss_alldownloads_all)

## 3.13 MECM Custom Reports

### 3.13.1 Builtin Reports in Microsoft Endpoint Configuration Manager

Reporting in Microsoft Endpoint Configuration Manager provides a set of tools and resources that help you use the advanced reporting capabilities of SQL Server Reporting Services (SSRS) and the rich authoring experience that Reporting Services Report Builder provides. Reporting helps you gather, organize, and present information about users, hardware and software inventory, software updates, applications, site status, and other Configuration Manager operations in your organization. Reporting provides you with a number of predefined reports that you can use without changes, or that you can modify to meet your requirements, and you can create custom reports. Use the following sections to help you manage reporting in Configuration Manager.

Configuration Manager provides report definitions for over 400 reports in over 50 report folders, which are copied to the root report folder in SQL Server Reporting Services during the reporting services point installation process. The reports are displayed in the Configuration Manager console and organized in subfolders based on the report category. Reports are not propagated up or down the Configuration Manager hierarchy; they run only against the database of the site in which they are created.

#### **Recommendation**

Builtin reports in Configuration help you to follow deployments, monitoring your Configuration Manager infrastructure and more.

### 3.13.2 Custom Reports in Microsoft Endpoint Configuration Manager

Microsoft Endpoint Configuration Manager uses Microsoft SQL Server Reporting Services Report Builder as the exclusive authoring and editing tool for both model-based and SQL-based reports. When you initiate the action to create or edit a report in the Configuration Manager console, Report Builder opens. When you create or modify a report for the first time, Report Builder is installed automatically. The version of Report Builder associated with the installed version of SQL Server opens when you run or edit reports. The Report Builder installation adds support for over 20 languages. When you run Report Builder, it displays data in the language of the operating system that is running on the local computer. If Report Builder does not support the language, the data is displayed in English. Report Builder supports the full capabilities of SQL Server 2008 Reporting Services, which includes the following capabilities:

- Delivers an intuitive report authoring environment with an appearance similar to Microsoft Office.
- Offers the flexible report layout of SQL Server 2008 Report Definition Language (RDL).
- Provides various forms of data visualization including charts and gauges.
- Provides richly formatted text boxes.
- Exports to Microsoft Word format.

You can also open Report Builder from SQL Server Reporting Services.

#### **Recommendation**

In case missing information that are not provided in built-in reports in Configuration Manager, you can use Report Builder to create your custom reports and design it the way you need.

### 3.13.3 Microsoft Endpoint Configuration Manager: Advanced Dashboards

The CMAD solution (Configuration Manager Advanced Dashboards) delivers a data-driven reporting overview of the Microsoft Endpoint Configuration Manager environment.

This solution consists of a rich set of dashboards designed to deliver real-time reporting of ongoing activity in your Configuration Manager environment.

Native Configuration Manager Reports are not replaced with this solution, the CMAD solution amplifies the data they show by providing additional data insights.

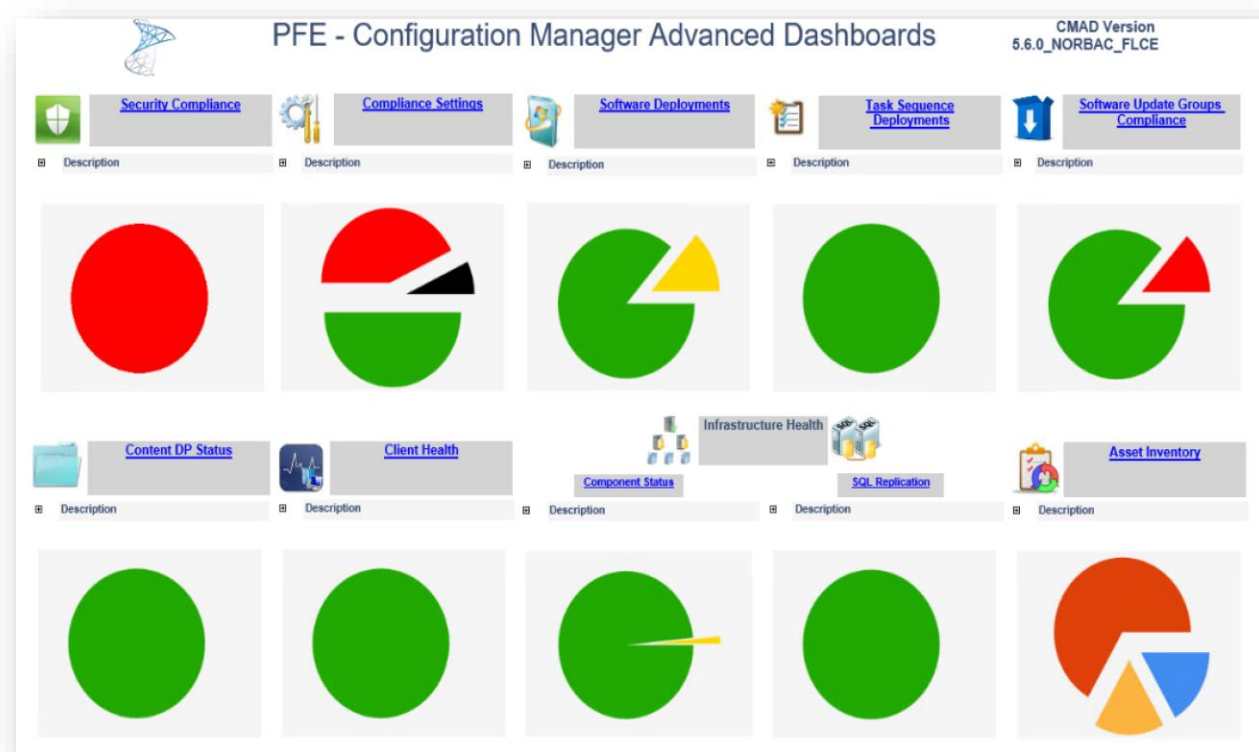
The dashboards in this solution were created based on field experience and on customers' needs to provide an overall view of various Configuration Manager functionality. The embedded charts and graphics provide details across the entire infrastructure.

#### **Key Features and Benefits**

The CMAD solution consist of 195+ dashboards covering the following Configuration Manager topics:

- Asset Inventory
- SoftwareUpdateManagement
- ApplicationDeployment
- ComplianceSettings
- InfrastructureMonitoring:
  - Site Replica
  - Contentreplication
- SoftwareDistribution
- Clients Health
- ServersHealth
- Security





## Blogpost

<https://secureinfra.blog/2019/01/10/configuration-manager-advanced-dashboards-rich-view-of-your-configuration-manager-environment/>

### 3.13.4 Microsoft Endpoint Manager: PowerBI Dashboard Integration

Engage with Microsoft Premier Field Engineering (PFE) to kickstart a modern reporting and visualization effort for Client Management scenarios with Power BI. Our structured engagement helps lay the groundwork for current and historical data visualization in MECM/Intune. Power BI is an industry leading SaaS suite that enables you to derive insights from unrelated sets of data in a coherent, visually immersive manner. Power BI is enterprise grade, ready for modeling and enables selfservice reporting and visualization.

Our engineers will walk your team through the onboarding of the Data Warehouse role in MECM, facilitating long term reporting. Knowledge transfer sessions around Power BI will allow your team to build familiarity with acquiring data, visualizing what's important and sharing with others in your organization.

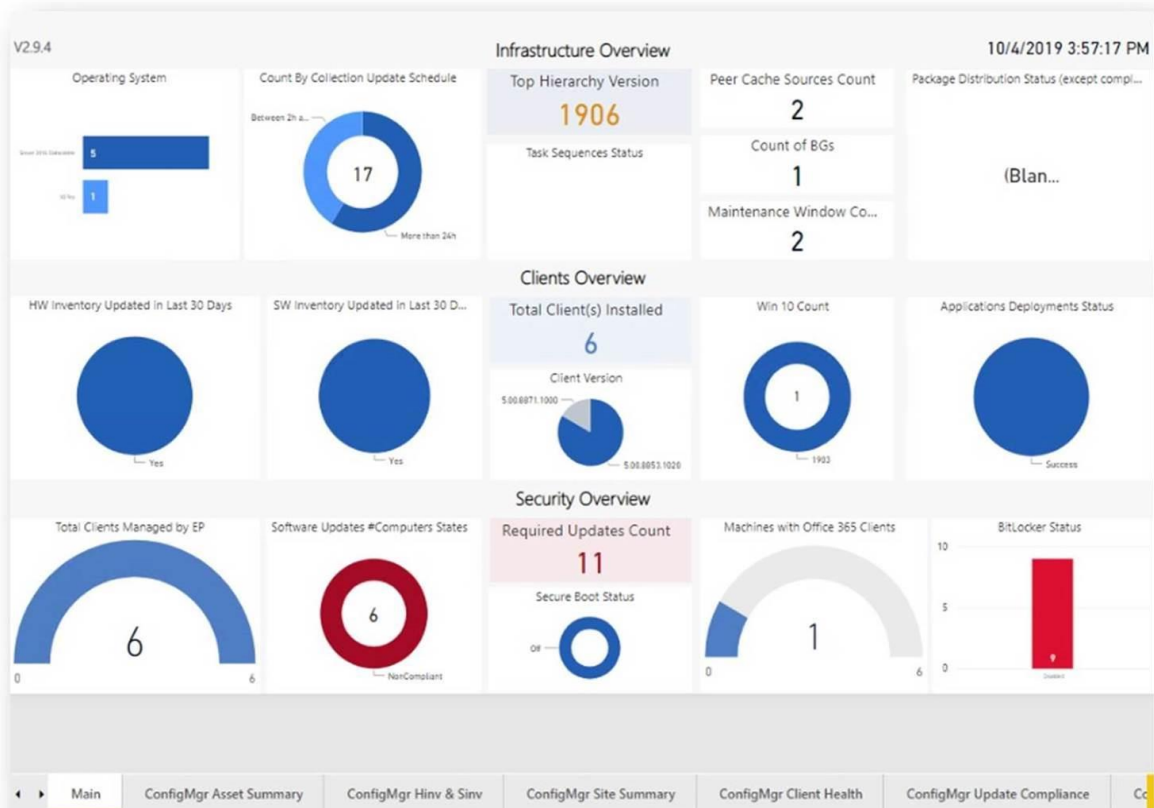
## Maximize Your Microsoft Investment

The following modules will be covered as a part of the delivery:

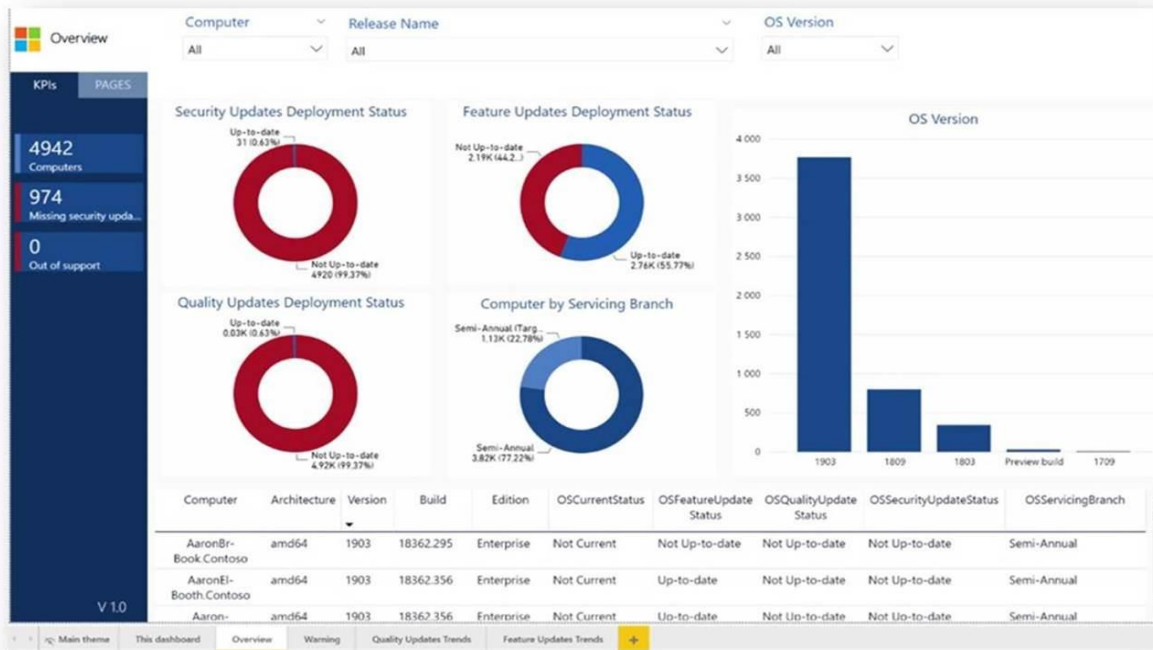
Pre-engagement Scoping Call to plan for publishing reports either to Power BI Service or Power BI Report Server

- Installation/Configuration of MECM Data Warehouse
- Knowledge Transfer Sessions
  - MECM Data Warehouse
  - What is Power BI?
  - Report Authoring with Power BI Desktop
  - Publishing Reports
  - Enabling Power BI Report Server
  - Reports for MECM Data Warehouse
  - Intune Data Warehouse Reports
  - Integration with Windows Analytics – Update Compliance
- Implementation of reference MECM/Windows Analytics and Intune Dashboard

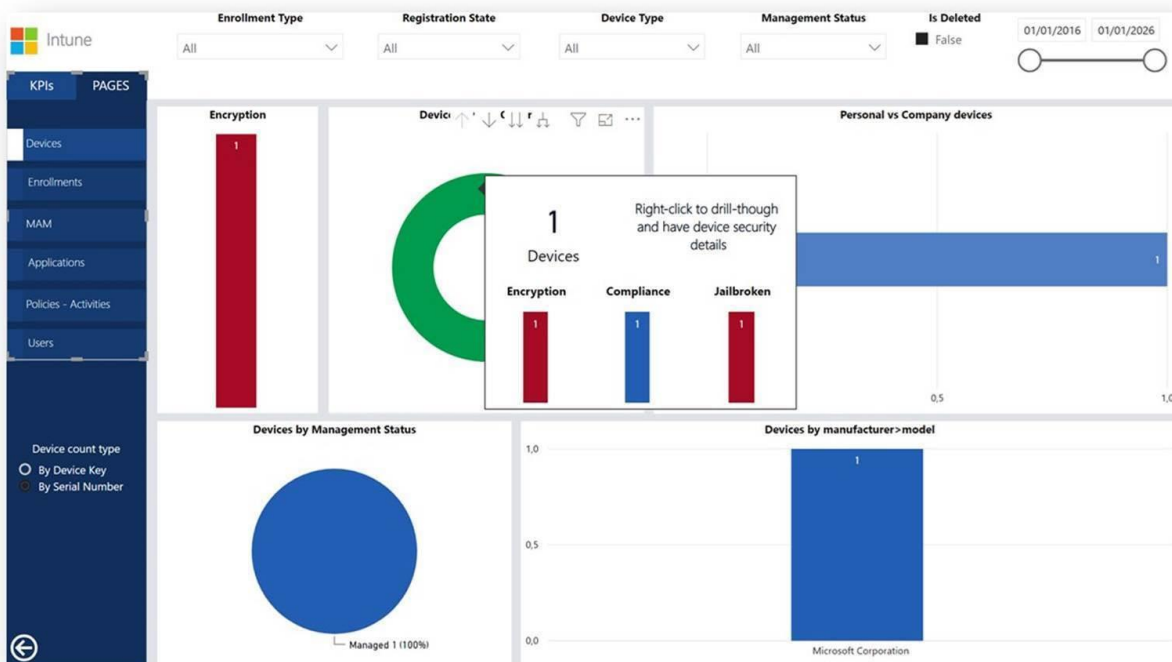
## MECM Power BI Dashboard



## Windows Analytics Power BI Dashboard



## Intune Power BI Dashboard

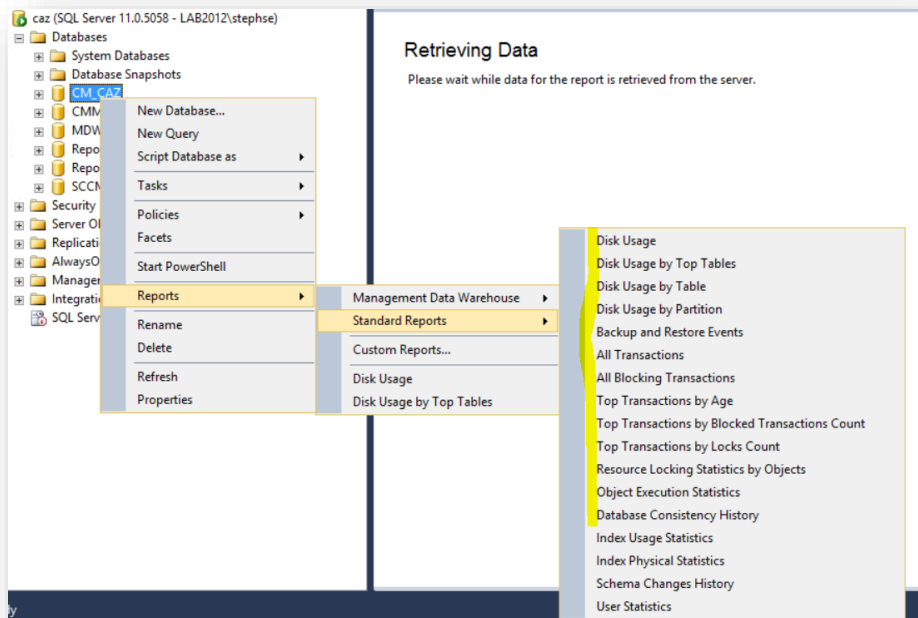


## Blogpost

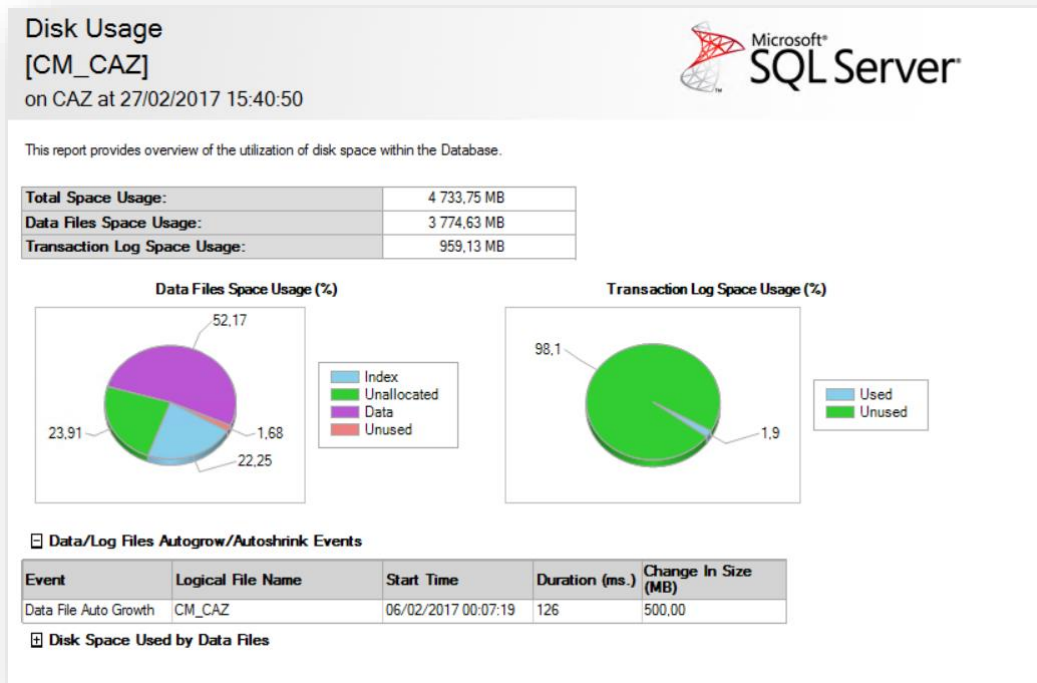
<https://techcommunity.microsoft.com/t5/premier-field-engineering/premier-offerings-system-center-configuration-manager-and-intune/ba-p/1044351>

### 3.13.5 SQL Custom Report

Built-in SQL reports are available through SQL management Studio.  
In order to access to those reports, follow the below instruction:



### Reports Examples



## Disk Usage by Top Tables

[CM\_CAS]

on 27/02/2017 15:38:32



This report provides detailed data on the utilization of disk space by top 1000 tables within the Database.

Table Name	# Records	Reserved (KB)	Data (KB)	Indexes (KB)	Unused (KB)
dbo.RecentlyUsedApps_DATA	24 544 834	35 990 840	21 382 424	14 366 600	241 816
dbo.DRSentMessages	4 998 872	17 337 464	17 218 208	68 328	50 928
dbo.CI_CurrentComplianceStatusDetails	12 740 184	13 728 080	13 416 816	175 648	135 616
dbo.INSTALLED_SOFTWARE_DATA	6 045 895	11 201 224	7 920 512	3 180 088	100 624
dbo.DBSchemaChangeHistory	52 033 620	9 909 296	9 845 224	58 464	5 608
dbo.CI_ComplianceHistory	31 671 632	9 083 128	4 011 856	5 021 536	49 736
dbo.Update_ComplianceStatus	25 668 955	7 601 784	2 860 528	4 714 992	26 264
dbo.Add_Remove_Programs_DATA	11 107 108	7 104 960	4 730 384	2 240 232	134 344
dbo.Services_DATA	12 125 328	6 579 416	5 607 256	863 920	108 240
dbo.CI_CurrentComplianceStatus	12 739 719	5 795 792	1 963 560	3 794 056	38 176
dbo.DRSReceivedMessages	1 479 799	5 364 600	5 335 104	24 136	5 360
dbo.TaskExecutionStatus	6 283 979	5 300 952	4 835 792	440 752	24 408
dbo.PNP_DEVICE_DRIVER_DATA	10 083 682	5 015 912	3 628 168	1 328 744	59 000
dbo.SoftwareInventory	35 162 848	4 828 320	2 815 528	1 977 160	35 632
dbo.CI_DocumentStore	358 280	3 944 592	3 798 920	105 944	39 728

### 3.14 Management Datawarehouse

The management data warehouse is a relational database that contains the data that is collected from a server that is a data collection target. This data is used to generate the reports for the System Data collection sets, and can also be used to create custom reports.

The data collector infrastructure defines the jobs and maintenance plans that are needed to implement the retention policies defined by the database administrator.

This feature allows to collect information specific to SQL Server. This information can subsequently be viewed through 3 types of report:

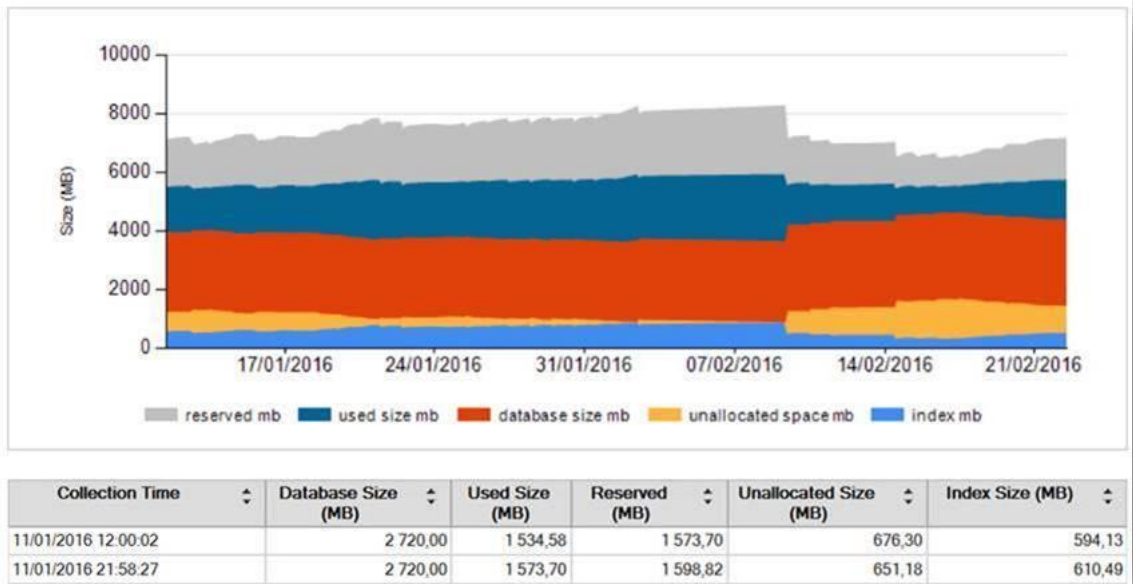
- Disk use
- Server Activity
- Query Statistics

The MDW, also called Data Collector, is based on a principle of collection of metadata stored, consolidated and aggregated in a dedicated database. For reasons of control of consumption of resources, it is highly recommended in production environments to set up this base on a separate SQL instance to the one monitored. The objective of these data storage, and power "timestamp" a metadata and thus to the retromonitoring, to be able to find information or a measure through time.

The 3 types of reports available are as follows.

3.14.1 Disk Usage

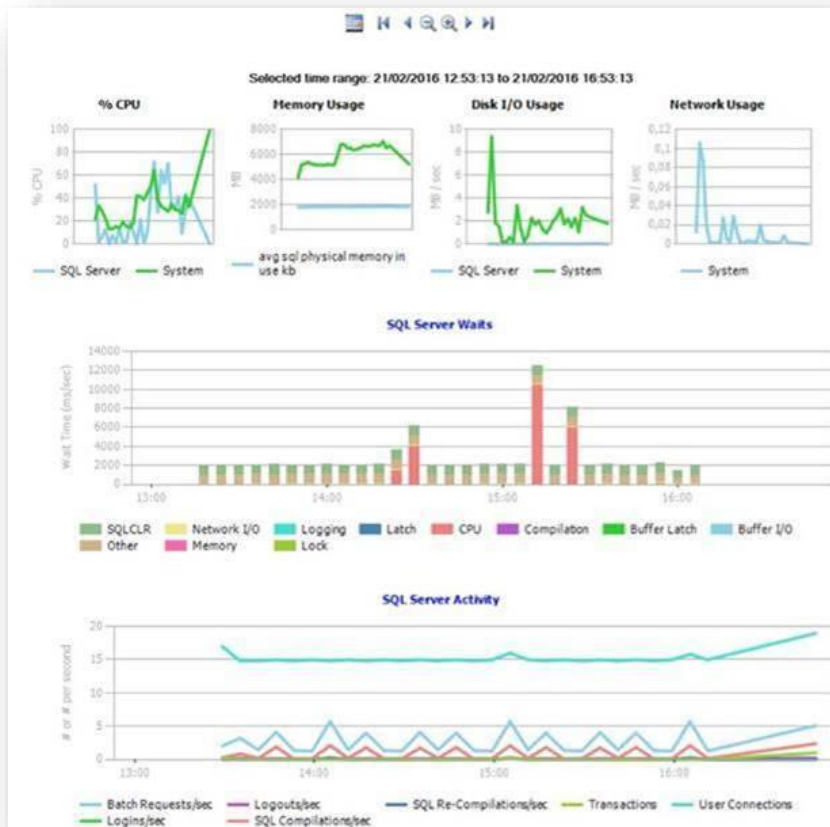
This report differs from the other two where it fits in the context of capacity planning. Via daily or weekly collection, it is possible to measure the evolution of the size of the data files and of their filling. Therefore, we can use it as a base in order to make projections. This report can also be useful to have a better understanding of the behaviour of the TempDB database for a given application.



3.14.2 Server Activity & Query Statistics

Server Activity and Query Statistics reports are in contrast with a view to supervision of performance. They provide information both on the state and health of the platform level (both system and SQL), and they also shed light on SQL Server workloads, displaying expensive queries, top SQL Server Wait Stats etc. This makes it possible for example to identify a potential bottleneck on a period, such a lock, a Page IO latch, etc... Finally, these reports provide the ability to operate a drill-down drill-up in order to move from a macroscopic view of the report to the request related to the event, or even to its execution plan.

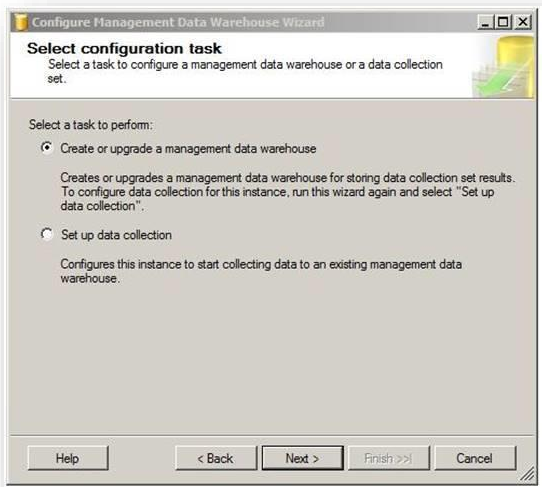




### 3.14.3 Installation

The configuration of a MDW solution implements two-step.

First of all, the first step is to launch the "Configure Management Data Warehouse" Wizard to create the MDW database.



The second step is setting itself the MDW, via the option "Set up data collection. Through this wizard, defined roles map users as authorized to administer the MDW. SQL Agent jobs are then created for collecting and loading metadata in the base dedicated to this purpose. Collection and upload frequency can be modified later on each of the collection sets via right-click - property.

#### 3.14.4 Configuration

After you run the Wizard "Configuration Management Data Warehouse", you will have three default Collection Set in the two modes of collection. The collection set "Disk use" is not in cache mode (because the goal is to set up a kind of report capacity planning, so collect data once per day is sufficient and it is not useful to be cached), and 'Server Activity' and 'Query Statistics' collection sets are in cached mode. The caching mode determines the relationship between data collection and downloading data to the MDW database.

##### Non-Cached

In non-cached mode, the collection and upload are set on the same schedule. Packages start, collect and upload data to their configured frequency and run until they finished. There is only one job created for non-cached mode Collection Set. This mode allows also the collection and upload of application data, the intervals specified in the job.

##### Cached Mode

In cached mode, the collection and the data upload are attached to two schedules of two different jobs. So in this case two jobs are created for each collection set. Each collection should be in general relatively close, while the upload is scheduled less frequently. For example, the "Server Activity" report, the upload is set by default every 15 minutes while the collection is made every 60 seconds. It's the frequency by default, but this is completely adjustable for each Collection Set.

Summary of the default settings:



Collection mode	Non-cached			Cached					
Upload schedule frequency	Every 6 hours			Every 15 minutes					
Data retention	730 days			14 days					
Collection Sets	Disk Usage			Server Activity			Query Statistics		
Collection Items	Item Name	Item Type	Output	Item Name	Item Type	Output	Item Type	Item Name	Output
	Disk Usage: Data Files	TSQL	snapshots.disk_usage	Server Activity: DMV Snapshots	7 TSQL queries	snapshots.os_wait_stats snapshots.os_latch_stats snapshots.  sql_process_and_system_memory snapshots.os_memory_nodes snapshots.os_memory_cleaks snapshots.os_schedulers snapshots.io_virtual_file_stats	Query Activity	Query Statistics – Query Activity	snapshots.query_stats, snapshots.notable_query_text, snapshots.notable_query_plan
	Disk Usage: Log Files	TSQL	snapshots.log_usage	Server Activity: Performance Counters	Performance Counters	snapshots.performance_counter_values snapshots.performance_counter_instances			

Technical Reference : [https://technet.microsoft.com/en-us/library/dd939169\(v=sql.100\).aspx](https://technet.microsoft.com/en-us/library/dd939169(v=sql.100).aspx)

### 3.15 Custom queries

#### System settings and resources

From a performance perspective, several counters and indicators can be monitored as well directly through SQL Server queries. Some of them have been already mentioned in custom reports parts. The purpose of this chapter is to allow point in time reference.

As explained previously, some configuration settings (like Max Degree of Parallelism for example) are in relation with the global settings and the resources of the server. Typically, it can be interesting to get the value of CPU processors count or global available memory to define these settings. This information can be caught up with the dm\_os\_sys\_info view:

```
SELECT
    sqlserver_start_time,
    cpu_count,
    hyperthread_ratio,
    physical_memory_kb,
    virtual_memory_kb,
    virtual_machine_type_desc
FROM
    sys.dm_os_sys_info
```

This query gets the information as well about when the SQL Server service has been started and if it is a physical or virtual server.

#### Performance counters

In the same scope, we have seen in the SCOM Management Pack chapter a couple of useful counters. Some of them can be queried as well thru SQL. The global rule is "each Perfmon counters specifically related to SQL".

It is also possible, with system views to easily get the value of the Page Life Expectancy counter, or the free space available in the TempDB database:

```
SELECT
    *
FROM
    sys.dm_os_performance_counters
WHERE
    counter_name IN ('Page Life Expectancy', 'Free Space in tempdb (KB)')
```

### Wait types and waiting sessions

```
SELECT TOP 20 *
FROM
    sys.dm_os_wait_stats
ORDER BY
    wait_time_ms DESC
```

```
SELECT
    session_id,
    database_id,
    DB_NAME(database_id) AS DBName,
    blocking_session_id,
    last_wait_type,
    cpu_time,
    total_elapsed_time
FROM
    sys.dm_exec_requests
WHERE
    session_id > 50
```

### Expensive queries

```
SELECT TOP 10
    qs.execution_count,
    qs.last_execution_time,
    qs.total_elapsed_time / qs.execution_count / 1000 as [Avg duration(ms)],
    qs.total_worker_time / qs.execution_count / 1000 as [Avg CPU (ms)],
    qs.total_logical_reads / qs.execution_count as [Avg Reads],
    st.text,
    qp.query_plan
FROM
    sys.dm_exec_query_stats qs
    CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) st
    CROSS APPLY sys.dm_exec_query_plan(qs.plan_handle) qp
ORDER BY
    [Avg duration(ms)] DESC
```

## 3.16 Supported Versions of SQL Server

For more information on which versions of SQL Server are supported for Microsoft Endpoint Configuration Manager, please refer to the following [link](#).

## 3.17 WSUS

Software Update Management in Microsoft Endpoint Configuration Manager provides a set of tools and resources that help manage the complex task of tracking and applying software updates to client computers in the enterprise. An effective software update management process is necessary to maintain operational efficiency, overcome security issues, and maintain the stability of the network infrastructure. However, because of the changing nature of technology and the continual appearance of new security threats, effective software update management requires consistent and continual attention

### 3.17.1 MECM WSUS cleanup

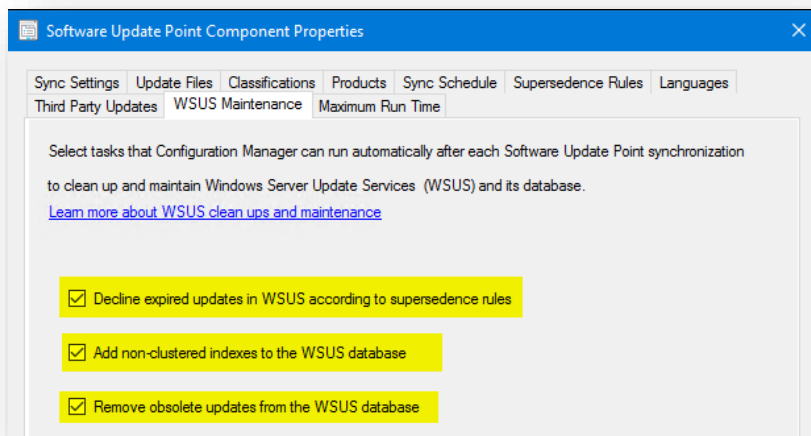
If you are using Microsoft Endpoint Configuration Manager version **1906 or newer**, enabling the WSUS Maintenance options in the Software Update Point configuration at the top-level site is recommended **to automate the cleanup procedures** after each synchronization. This would effectively handle all cleanup operations described in this article except backup and re-indexing of WSUS database. You should still automate backup of WSUS database along with re-indexing of the WSUS database on a schedule.

### **Decline Expired Updates/Add non-clustered indexes/Remove Obsolete Updates to the WSUS database**

1. In the Configuration Manager console, navigate to **Administration > Overview > Site Configuration > Sites**.
2. Select the site at the top of your Configuration Manager hierarchy.
3. Click **Configure Site Components** in the Settings group, and then click **Software Update Point** to open Software Update Point Component Properties.
4. In the **WSUS Maintenance** tab, select **Add non-clustered indexes to the WSUS database**.
  - On each SUSDB used by Configuration Manager, indexes are added to the following tables:
    - tbLocalizedPropertyForRevision
    - tbRevisionSupersedesUpdate
5. In the **WSUS Maintenance** tab, select **Remove obsolete updates from the WSUS database**.
  - The obsolete update removal will be allowed to run for a maximum of 30 minutes before being stopped. It will start up again after the next synchronization occurs.
6. In the **WSUS Maintenance** tab, select **Decline expired updates in WSUS according to supersedence rules**

### **Reference**

<https://docs.microsoft.com/en-us/configmgr/sum/deploy-use/software-updates-maintenance#wsus-cleanup-starting-in-version-1906>



### 3.17.2 Rebuilding WSUS

The performance of large Windows Server Update Services (WSUS) deployments will degrade over time if the WSUS database is not maintained properly. The WSUSDBMaintenance script is a T-SQL script that can be run by SQL Server administrators to re-index and defragment WSUS 3.0 databases. It should not be used on WSUS 2.0 databases.

Use the SQL script [WSUSDBMaintenanceScript](#) to Re-index WSUS database

#### **Reference**

An informative blog and the complete guide to Microsoft WSUS and Configuration Manager SUP maintenance is <https://support.microsoft.com/en-us/help/4490644/complete-guide-to-microsoft-wsus-and-configuration-manager-sup-maint>

### 3.17.3 Shared Database for Configuration Manager Software Update Points

When you install more than one software update point at a primary site, use the same WSUS database for each software update point in the same Active Directory forest. By sharing the same database you can significantly **mitigate the client and network performance impact** that can occur **when clients switch to a new software update point**. When a client switches to a new software update point that shares a database with the old software update point, a delta scan still occurs, but this scan is much smaller than it would be if the WSUS server had its own database.

#### **References**

<https://docs.microsoft.com/en-gb/sccm/sum/plan-design/software-updates-best-practices>

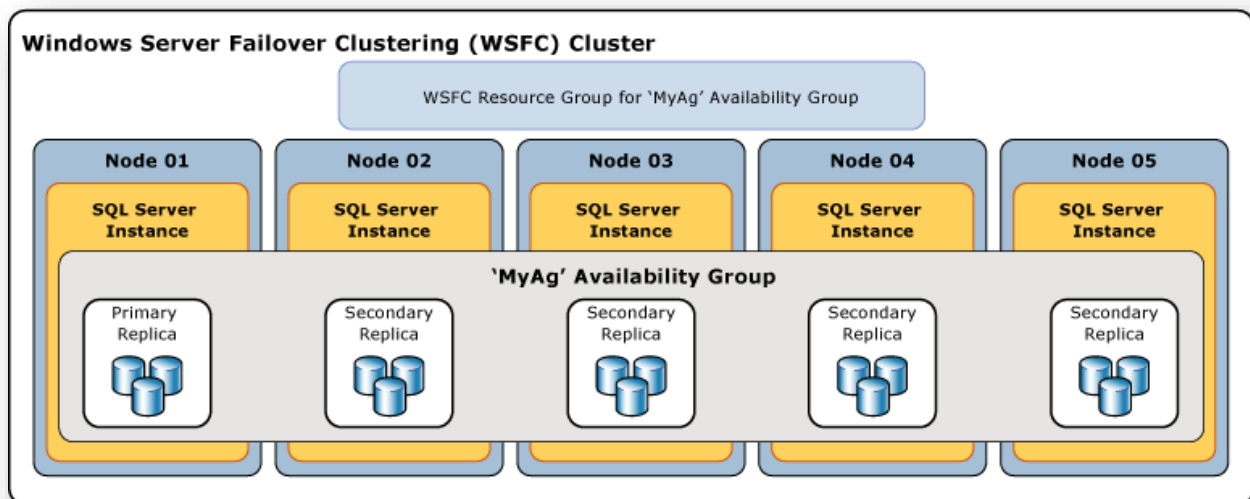
Microsoft release a blog post on how to setup a Shared SUSDB for Configuration Manager Software Update Points.

### 3.18 Always ON Availability Groups

The Always On availability groups feature is a high-availability and disaster-recovery solution introduced in SQL Server 2012 to maximize the availability of a set of user databases. This set is called an availability group and the included databases (availability databases) will fail over together.

These groups are host by *availability replicas*, composed of a primary replica which hosts the primary databases and *secondary replicas* which hosts a set of secondary databases and serves as a potential failover targets for the availability group.

This feature relies on Windows Server Failover Cluster. Each availability replica of a given availability group must reside on a different node of the same WSFC.



Configuration Manager supports the use of Availability Groups at primary sites and the central administration site and can be done On-Premises or in Azure

#### Prerequisites

Each supported edition of SQL Server can be used to set up an availability group.

But only Advanced Availability Groups (AAG) can be used. It requires **Enterprise Edition** of SQL Server.

When performing the installation, the Configuration manager setup account must be

- A member of the local Administrators group on each computer that's a member of the availability group.
- A sysadmin on each instance of SQL Server that hosts the site database.

- The computer account of the site server must be a member of the local **Administrators** group on each computer part of the availability group.

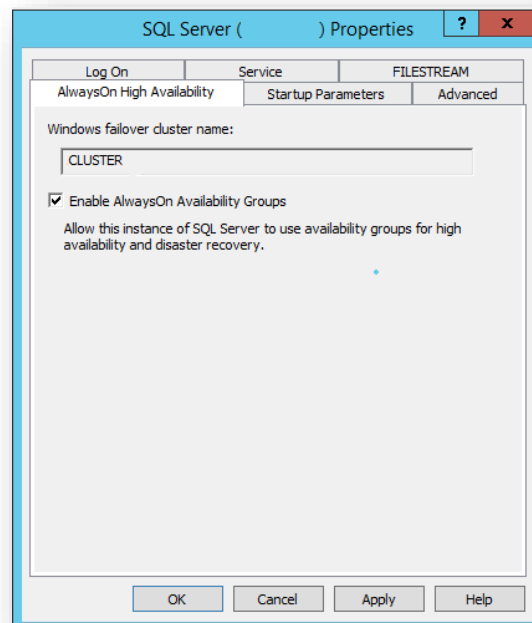
The databases must be in the **Full recovery model** to be included in an availability group.

As an example, for a simple two replicas topology with one availability groups and one listener, you'll need:

- 2 different hosts, part of the same WSFC Cluster
- 1 virtual IP to configure the availability group listener
- Prestaging the computer account for a listener virtual network name or granting Create Computer objects right to the CNO
- A shared location, if adding the database using the Full method

### Enabling Always ON Availability Groups feature

Always ON Availability Groups feature is disabled by default. You need to activate it in the SQL Server services properties, using the SQL Server Configuration Manager



After restarting the service, you can check the status with the following T-SQL:

```
SELECT SERVERPROPERTY('IsHadrEnabled')
```

### Availability modes

When using availability groups, you can choose between two availability modes:

Asynchronous Commit: In this mode, the primary replica does not wait for any secondary replica to write incoming transaction log records to disk. Under *asynchronous-commit mode*, the secondary replica never becomes synchronized with the primary replica. Asynchronous-commit mode can be useful in a disaster-recovery scenario in which the primary replica and the secondary replica are separated by a significant distance and where you do not want small errors to impact the primary replica or in or situations where performance is more important than synchronized data protection

Synchronous Commit: In this mode, the primary replica waits for the secondary replica to confirm that it has hardened the log, at the cost of increased transaction latency. The secondary database remains SYNCHRONIZED as long as data synchronization continues. This guarantees that every transaction that is committed on a given primary database has also been committed on the corresponding secondary database. When all the secondary databases of a given secondary replica are synchronized, synchronous-commit mode supports manual failover and automatic failover.

Use of an asynchronous commit replica as the site database can put the integrity of your site and data at risk. An asynchronous replica can be out of sync by design.

### **Availability Groups**

For each replica part of the availability group configuration:

The Connections in Primary Role setting is **Allow all connections**

The Readable Secondary setting is **Yes**

### **Failover**

Three forms of failover exist: automatic failover (without data loss), manual failover (without data loss), and forced manual failover (with possible data loss), called forced failover. Automatic and planned manual failover preserve all your data.

During the failover, the failover target takes over the primary role, recovers its databases, and brings them online as the new primary databases. The former primary replica, when available, switches to the secondary role, and its databases become secondary databases.

**Automatic failover** is only supported when using the **synchronous-commit** availability mode.

In version 1902 and earlier, you need to configure all availability groups on the SQL Server for manual failover. This configuration is needed even if it doesn't host the site database.

Starting in version 1906, Microsoft Endpoint Configuration Manager supports using the availability group synchronous replicas when set to Automatic Failover.

Set Manual Failover when:

- You run Configuration Manager setup to specify use of the site database in the availability group.
- You install any update to Configuration Manager. (Not just updates that apply to the site database).

### **Adding a database**

Adding a database in an availability group can be done by different ways:

Manually, by restoring a Full backup on the secondary replica, and all the following log backups, with the NORECOVERY option. Once restored, join the database to the availability group. (Join Only)

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/manually-prepare-a-secondary-database-for-an-availability-group-sql-server?view=sql-server-ver15>

Automatically (Full), the database backup will then be done and restored on the secondary replica. It requires a shared location accessible by all the replicas

Finally, by using the automatic seeding option.

Starting with SQL Server 2016, Automatic Seeding is a feature to initialize a secondary replica. Automatic seeding uses the log stream transport to stream the backup to the secondary replica for each database of the availability group using the configured endpoints. This feature can be used either during the initial creation of an availability group or when a database is added to one.

All members need the same seeding mode.

When setup creates the database, and you configure automatic seeding, the availability group must have permissions to create the database. This requirement applies to both a new database or recovery.

For Automatic Seeding, It is recommended to use the same data and log file path on all replicas participating in an availability group when using this feature in SQL Server 2017 (but is required in SQL Server 2016)

For the Full method, using the same data and log file path is required

Configuration Manager setup includes a prerequisite check to verify this configuration when creating a database through install or recovery.



It's not supported to install a new site with its database in an availability group.

## Databases

Configuration Manager requires its databases to be specifically configured.

To check if the prerequisites are met, you can use the database verification script on each replica

```
SET NOCOUNT ON
DECLARE @dbname NVARCHAR(128)

SELECT @dbname = sd.name FROM sys.sysdatabases sd WHERE sd.dbid = DB_ID()

IF (@dbname = N'master' OR @dbname = N'model' OR @dbname = N'msdb' OR @dbname =
N'tempdb' OR @dbname = N'distribution')
BEGIN
RAISERROR(N'ERROR: Script is targetting a system database. It should be targetting the
DB you created instead.', 0, 1)
GOTO Branch_Exit;
END ELSE
PRINT N'INFO: Targeted database is ' + @dbname + N'.'

PRINT N'INFO: Running verifications....'

IF NOT EXISTS (SELECT * FROM sys.configurations c WHERE c.name = 'clr enabled' AND
c.value_in_use = 1)
PRINT N'ERROR: CLR is not enabled!'
ELSE
PRINT N'PASS: CLR is enabled.'

DECLARE @repltable TABLE (name nvarchar(max),minimum int,maximum int,config_value
int,run_value int)

INSERT INTO @repltable
EXEC sp_configure 'max text repl size (B)'

IF NOT EXISTS(SELECT * from @repltable where config_value = 2147483647 and run_value =
2147483647)
PRINT N'ERROR: Max text repl size is not correct!'
ELSE
PRINT N'PASS: Max text repl size is correct.'

IF NOT EXISTS (SELECT db.owner_sid FROM sys.databases db WHERE db.database_id = DB_ID()
AND db.owner_sid = 0x01)
PRINT N'ERROR: Database owner is not sa account!'
ELSE
PRINT N'PASS: Database owner is sa account.'

IF NOT EXISTS (SELECT * FROM sys.databases db WHERE db.database_id = DB_ID() AND
db.is_trustworthy_on = 1)
PRINT N'ERROR: Trustworthy bit is not on!'
ELSE
PRINT N'PASS: Trustworthy bit is on.'

IF NOT EXISTS (SELECT * FROM sys.databases db WHERE db.database_id = DB_ID() AND
db.is_broker_enabled = 1)
```

```

PRINT N'ERROR: Service broker is not enabled!'
ELSE
PRINT N'PASS: Service broker is enabled.'

IF NOT EXISTS (SELECT * FROM sys.databases db WHERE db.database_id = DB_ID() AND
db.is_honor_broker_priority_on = 1)
PRINT N'ERROR: Service broker priority is not set!'
ELSE
PRINT N'PASS: Service broker priority is set.'

PRINT N'Done!'

Branch_Exit:

```

It is not supported to add to the availability group, the following databases:

- Reporting database
- WSUS database

### **Listener**

An availability group listener is a virtual network name (VNN) to which clients can connect in order to access a database in a primary or secondary replica of an Always On availability group. An availability group listener enables a client to connect to an availability replica without knowing the name of the physical instance of SQL Server to which the client is connecting. The client connection string does not need to be modified to connect to the current location of the current primary replica.

To add a listener, you can use the Wizard or the following T-SQL statement:

```

ALTER AVAILABILITY GROUP <AAGName> ADD LISTENER '<AAGListenerName>' (WITH IP
((<'<IPAddress>'>)), PORT = <PortNumber>);

```

### **Multiple Subnet Failover**

Starting in version 1906, you can enable the MultiSubnetFailover connection string keyword in SQL Server. This option allows more aggressive connection retries, especially useful in a multi subnet configuration, to reduce the reconnection time.

To use this option, you need to add the following value in both registry on the site server:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\SMS\Identification -> MSF Enabled = 1 (DWORD)

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\SMS\SQL Server -> MSF Enabled = 1 (DWORD)

In version 1906 and 1910, it may happen that these registry values revert back to 0. In this case, it will be necessary to set it back to 1. This behavior is under investigation

## **Management**

In an Always ON Availability Groups configuration, replicas are totally independent instances.

It means each configuration or maintenance action must be performed on all replicas.

- Instance configuration
- Logins creation
- Jobs creation
- Maintenance tasks for system databases
- Maintenance tasks for database not joined to an availability group
- Server-Level objects (endpoint, linked servers, ...)
- .....

**Do not use the Failover Cluster Manager to manipulate availability groups**, for example:

- Do not add or remove resources in the clustered service (resource group) for the availability group.
- Do not change any availability group properties, such as the possible owners and preferred owners. These properties are set automatically by the availability group.
- Do not use the Failover Cluster Manager to move availability groups to different nodes or to fail over availability groups. The Failover Cluster Manager is not aware of the synchronization status of the availability replicas, and doing so can lead to extended downtime. You must use Transact-SQL or SQL Server Management Studio.

## **References**

[Getting Started with Always ON Availability Groups](#)

[Add a database to an availability group with the Wizard](#)

<https://docs.microsoft.com/en-us/configmgr/core/servers/deploy/configure/sql-server-alwayson-for-a-highly-available-site-database>

<https://docs.microsoft.com/en-us/configmgr/core/servers/deploy/configure/configure-aoag>