SharePoint

Microsoft® SQL Server® 2012

Maximizing SQL 2012 Performance for SharePoint 2013 Whitepaper

This Technical white paper describes why and how you should make SQL Server changes in order to maximize SharePoint 2013 Performance.

Vlad Catrinescu

MCITP, MCSA, MCTS

Abstract

As a SharePoint Consultant I get to see dozens of SharePoint farms every year, and one of the most common complaints I get from clients - is that SharePoint is slow. A lot of people think that SharePoint is slow because the SharePoint servers are lacking resources or, simply, because SharePoint is a slow product. Although resources allocated to the SharePoint servers (e.g. Web Front Ends and App servers) are important, not a lot of people realize that SharePoint performance is directly related to the database, SQL Server. In fact, 94% of SharePoint data is stored in SQL[1].

In this whitepaper, we will explore many options on how to optimize SQL Server 2012 for SharePoint 2013. Here is a high level overview of what we intend to explore.

- **Plan before you install**
  In this section we will review important best practices on how to format your disks, as well as how to plan where databases, logs and your Temp DB will be located. Furthermore, depending on the purpose of your SharePoint 2013 Farm, we will determine which databases are most important.

- **How to install SQL Server 2012**
  In this section we will discuss what changes during installation can impact SQL 2012 Server performance and explain the reasons behind them. For example, did you know SharePoint 2013 uses a different collation than the default SQL one?

- **Post installation changes**
  In this section we will examine the changes required immediately after you have installed SQL 2012. This will ensure that we have the correct settings from the outset. We will cover everything from Initial DB size to Fill Factor to Instant File Initialization.

- **How to keep it performing well for the years to come**
  Now that you installed SharePoint 2013, Web Applications are starting to be created, content is added. Unfortunately, if you want to keep your SharePoint fast and your users happy, your job is not done. In this section we will make a list of things you need to check and modify, together with a schedule. Luckily, since we made things right in the **Post Installation changes**, you won't have to do them very often.

[1] http://www.microsoftvirtualacademy.com/training-courses/tuning-sql-server-2012-for-sharepoint-2013-jump-start

Author

Vlad Catrinescu

As a Microsoft Certified Professional since 2008 my whole IT career has been about mastering Microsoft technologies and providing the most value to organizations that choose to use them. I have a deep knowledge of SharePoint Technologies as well as many of the networking technologies used for IT Infrastructure allowing me to create and implement a comprehensive plan for creating information governance, business processes, and SharePoint infrastructure to meet the assessed needs.

My knowledge is backed up by my numerous Microsoft Certifications such as Microsoft Certified IT Professional (**MCITP**) in both SharePoint 2010 and Enterprise Desktop Administrator: Windows 7 as well as Microsoft Certified Solutions Associate (**MCSA**) on Windows Server 2012.

I am also very social and love to share my knowledge and help others on Technet, MSDN and other forums. I very often write posts on my own blog Absolute SharePoint and I am also a guest blogger on other popular sites like NothingButSharepoint. Furthermore, I am a founding partner and site administrator for the newest and highly thriving community at SharePoint-Community.net. Don't hesitate to connect with me on Twitter and LinkedIn using the links below (click on the images) and also to participate and win prizes at SharePoint-Community.net.

CONNECT WITH ME HERE …

twitter

Linked in ®

10 reasons to join the SharePoint-Community.Net

- Learn about SharePoint
- Make new friends
- Share your knowledge
- Stay informed
- Ask questions
- Help Others
- Make a name for yourself
- Generate traffic to your blog
- No recruiters or SPAM
- Win cool prizes

Reviewers

I would like to thank very much the following persons for taking the time to review this whitepaper to ensure its quality prior to publication.

Mark Jones

founder, **Collaboris**

Gokan Ozcifci

Microsoft Community Contributor, **Blog**

Jesper M. Christensen

SharePoint Architect and founder of **extri:co a/s**

Jasjit Chopra

SharePoint Architect & Founder of **www.SharePointPower.com**

Disclaimer

This document is provided "as is". The information and views expressed in this whitepaper, including URL and other web site references, may change without notice. You assume the entire risk of use.

This document does not provide you with legal rights to the ownership of a Microsoft product, but just the use, unless this is explicitly stated in the document. "Trial" keys are provided for a single purpose of test.

Product or company names mentioned herein may be the trademarks of their respective owners.

You can copy and use this whitepaper for your projects, labs - and other needs.

For more information, please contact Vlad Catrinescu at the following address vladcatrinescu@hotmail.com

Table of Contents

1. Planning your SQL Server install

"If you don't know where you are going, you'll end up someplace else."

Before installing SQL Server 2012, we first have to plan how we will configure it. This step is very important because if you don't spend enough time planning and preparing, this will lead to problems further down the line.

Physical vs Virtual

Do we virtualize SQL or we give it a dedicated physical server? That is a huge question that has been debated over and over. On one side, you have Microsoft that is "encouraging" you to virtualize your SQL because technology has advanced and cloud solutions are now as performing (who would put SQL in Azure if Microsoft said it's bad?) And on the other side, you have the SQL guys who talk about the hypervisor performance cost. The one thing for sure is that, a physical SQL Server will always outperform a virtual SQL server with the same specs, however virtualization has some nice pros! Unfortunately, I cannot give you the golden answer to this question, but I can give you the pros and cons!

Pros

- High availability because of the hypervisor, not because of clustering. For instance, the new HyperV Cluster in Windows Server 2012 will allow you to easily move machines between hosts when there are updates or problems.
- Easier to scale up & down. Since you're talking VMs, you can move the VM to bigger hardware if there's a need for more resources
- Can be easier to recover at DR. I say can be. The reason is you can basically snapshot the VM and restart it at the DR site intact. You're not in the position to reload the OS, reload SQL Server, etc.

Cons:

1. Very heavily loaded VMs have seen disk I/O issues, even with dedicated I/O paths, etc. Sometimes you need the physical hardware without interference.
2. Over allocation of a virtual host. If you have too much running on a host, it's not just I/O that could be impacted. It could be memory and CPU bottleneck issues on the physical host which impairs the SQL Server.
3. Harder to protect. Something that doesn't get a lot of press... But if it's easier for me at DR because I can snapshot the VM and copy it to somewhere. That means it's easier to take the system intact because I'm no longer having to breach physical security to get the system.[2]

For the production environment, go Physical if you can and have the budget, however for your QA/UAT I suggest you go Virtual.

[2] http://ask.sqlservercentral.com/questions/1139/what-are-the-pros-and-cons-of-running-sql-server-o.html

Gather the right Hardware

One of the most important considerations is choosing the correct hardware. If SQL Server is installed on poor performing hardware, or hardware that isn't up to the job, then this will have a big impact on overall performance.

RAM & CPU

Your SQL Server needs to have enough RAM to function properly. If your SQL Server runs **only** SharePoint, here are the minimum requirements[3] [4] [5].

	Small Farm Deployment (0-500GB of Data)	Medium Farm Deployment (501-1TB of data)	Large Farm Deployment (1-2TB of Data)	Very Large Farm Deployment (2-5 TB of Data)	Special Cases
Ram Required	8GB	16GB	32 GB	64GB	64GB+
CPU	4	4	8	8	8

Disks

For a fast and optimal SQL install, I suggest at least 6 Disks:

1. TempDB
2. TempDB Logs
3. SharePoint Databases
4. SharePoint Databases Logs
5. OS
6. SQL binaries and other application Installs

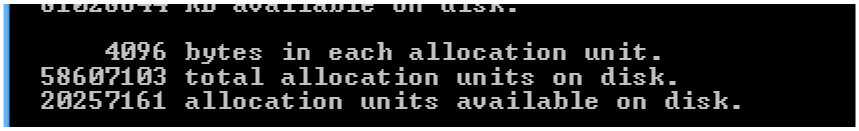
However, the more disks you get the better. For example, a 7th one could be used for Search Databases to make it faster. An 8th and a 9th one could be used to separate Temp DB and Content DB files on separate disks to make them even faster! Testing and user data shows that insufficient disk I/O for tempdb can significantly impede overall farm performance. To avoid this issue, allocate dedicated disks for the drive that stores tempdb data files[6].

[3] http://technet.microsoft.com/en-ca/library/cc262485.aspx
[4] http://technet.microsoft.com/en-US/library/hh292622(v=office.15).aspx
[5] http://technet.microsoft.com/en-US/library/ff758647(v=office.15).aspx
[6] http://technet.microsoft.com/en-US/library/hh292622(v=office.15).aspx

NTFS Allocation size

SQL server reads and writes 64k at a time, therefore is a best practice to make sure your disks are formatted with an allocation size of 64K and not the 4K default. This small change alone, will improve your SQL performance by up to 30%[7]

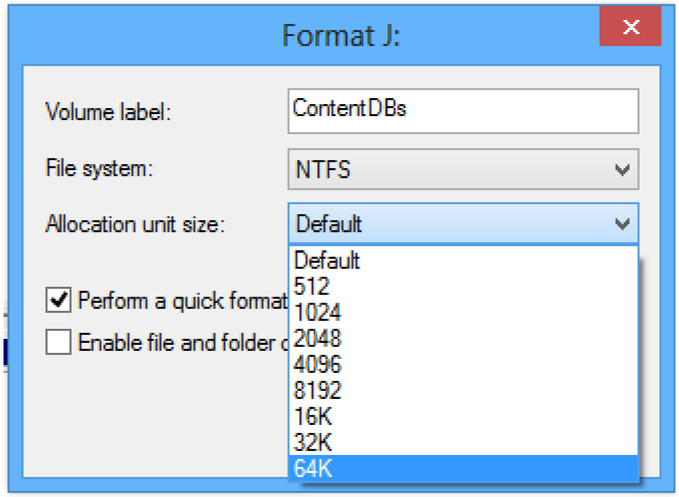
To check what your Allocation size is, in an Administrator Command Prompt type: "chkdsk c:" (Where C: is your disk letter) at the end you will see this:



```
81028644 KB available on disk.

     4096 bytes in each allocation unit.
 58607103 total allocation units on disk.
 20257161 allocation units available on disk.
```

In this example, you see my allocation size is 4K on this disk! To change it, you will have to reformat your hard drive, which isn't always possible. However, if you are installing your SQL now, you can change the default allocation size when formatting the disk.



Format J:

Volume label: ContentDBs

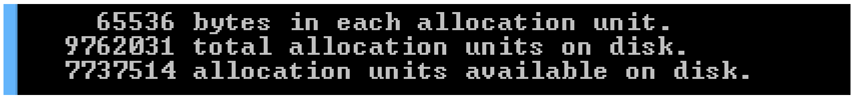
File system: NTFS

Allocation unit size: Default

☑ Perform a quick format

☐ Enable file and folder c

And if I check my disk again afterwards, I See it's successfully in 64K! (64*1024=65536)

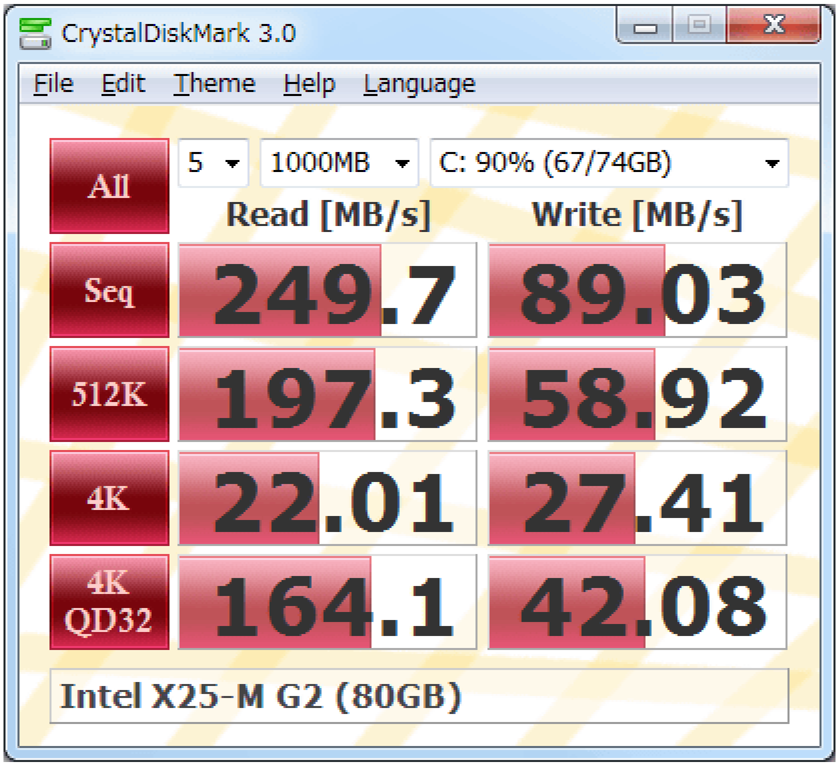


```
    65536 bytes in each allocation unit.
  9762031 total allocation units on disk.
  7737514 allocation units available on disk.
```

[7] http://www.microsoftvirtualacademy.com/training-courses/tuning-sql-server-2012-for-sharepoint-2013-jump-start

Benchmark your disks!

If your disks are not all the same model, benchmark them to know which one is fastest. This will be useful in the next step. For my tests I used a software called CrystalDiskMark. This is an example of the results it gives you!



5 | 1000MB | C: 90% (67/74GB)

	Read [MB/s]	Write [MB/s]
Seq	249.7	89.03
512K	197.3	58.92
4K	22.01	27.41
4K QD32	164.1	42.08

Intel X25-M G2 (80GB)

Network Speed

As you already know, everything between SQL and SharePoint travels through the network. If your SQL is all on SSD's with 20GB of Ram and same for your SharePoint, but you only got a 1Mbs cable between the two, that also answers client request, you will still hit a bottleneck. You need to make sure you have a fast Speed between your SQL and SharePoint. Furthermore, if you can, use a dedicated network adapter only for SQL-SharePoint communication, and a different one for client requests. You can also use Windows Server 2012 NIC Teaming feature to make them faster!

What is more important to you?

What are you going to use your SharePoint for? Will your farm be hosting a public Website where people will browse pages and download documents a lot? Or will your farm be hosting a Collaboration portal where your users will upload dozens of documents daily?

This information is very important, because it will change the way you prioritize your databases. [8]

[8] http://technet.microsoft.com/en-US/library/hh292622(v=office.15).aspx

In the case you have a farm where most of the content is only read and downloaded, and users don't upload a lot (ex: a public website), you will have to prioritize your content databases data files, instead of the logs. The table below shows the priority of Databases for a Read-Intensive Farm.

Speed/Scenario	Read Intensive Scenario (ex Public Website)
Fastest	Tempdb data files and transaction logs
.	Database data files
.	Search databases, except for the Search administration database
Slowest	Database transaction log files

However, in a Farm where your users will be collaborating, uploading documents, content, etc. ., you will need to prioritize the Database transaction log files to ensure optimum speed for your users.

Speed/Scenario	Write Intensive Scenario (collaborative workspace)
Fastest	Tempdb data files and transaction logs
.	Database transaction log files
.	Search databases, except for the Search administration database
Slowest	Database data files

Notice how in the read Intensive scenario, the Database data files (mdf/ndf) over the logs. That will make sure SQL will be able to answer faster to SharePoint's request for data. However, in the Write intensive scenario we prioritize the logs since that is where SharePoint will write the data before SQL moves it to the data files.

RAID

To improve performance and redundancy, you can also use hardware level RAID. Although RAID 5 seems the best bet for Performance VS Cost, it is highly recommended to use RAID 10 for your databases because RAID 5 has a bad write performance.

Document how you will use your disks before installing SQL

Before we start to install SQL Server 2012, we will first document where we put everything. Documentation can be a tedious task, however it is extremely useful as a reference point for others to that they can understand how SharePoint has been configured and built. . Here is a sample Example:

SharePoint 2013 Production Farm SQL Disks				
Drives from fastest to slowest	**Assigned Letter**	**Size***	**Usage**	**Notes**
0	I:	20GB	Temp DB Logs	Fixed Size VHD
1	H:	20GB	Temp DB	Fixed Size VHD
2	G:	50GB	SharePoint DB Logs	
3	F:	200GB	SharePoint Search	
4	C:	80GB	OS	
5	E:	2TB	SharePoint Databases	
6	D:	100GB	Software	

Size in the table is only an example. In your real database, disk size will depend on how big your farm needs to be!

Vlad Catrinescu
www.absolute-sharepoint.com
www.sharepoint-community.net

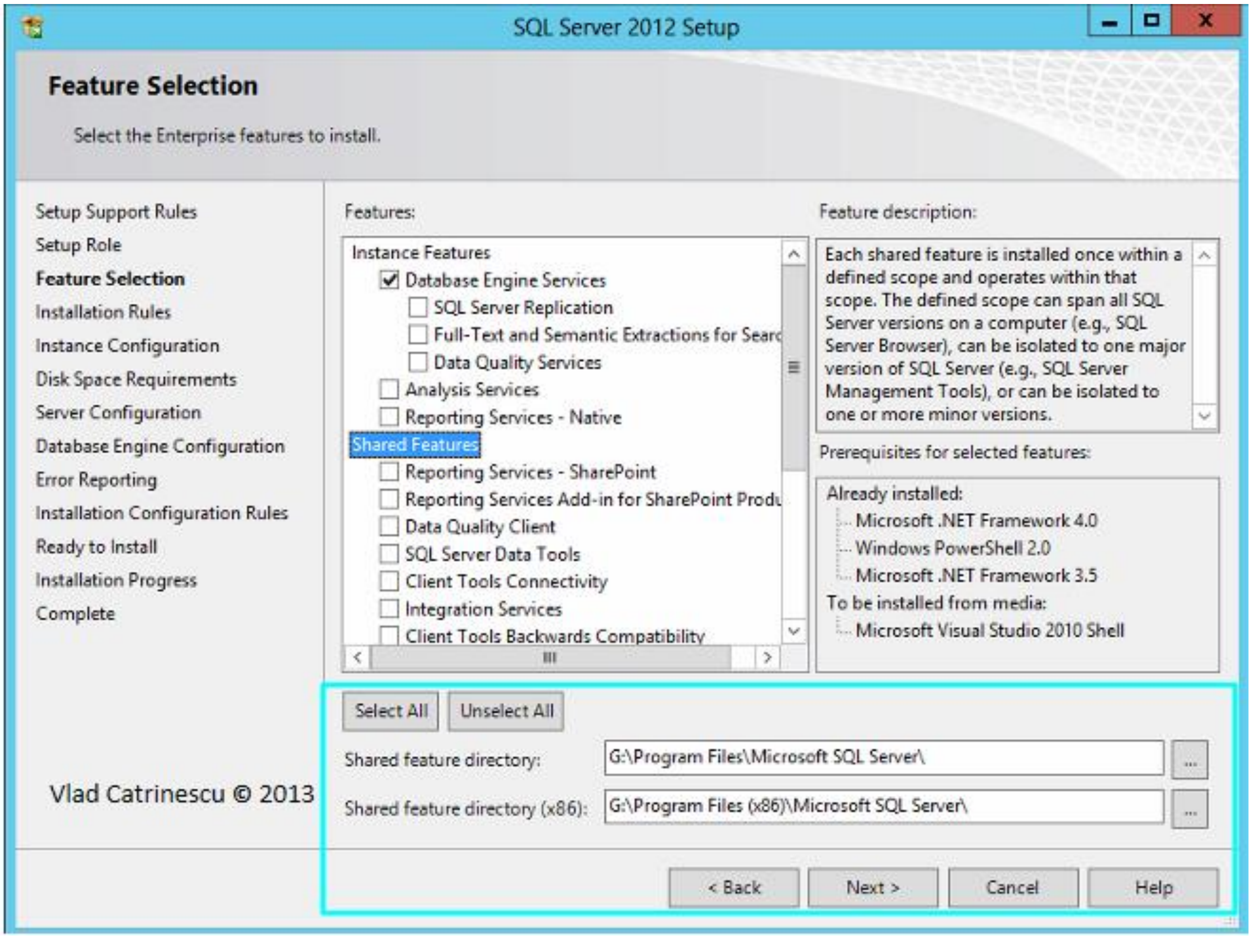
2. Installing SQL Server 2012

"The beginning is the most important part of the work". -- Plato

It's true that begging on the right foot is very important, especially in SQL, since some settings can only be done once and not changed afterwards! However, if your SQL is already installed don't worry! There are many optimizations you can still do. They are outlined in Module 3: Post Installation Configurations.

Feature Selection Screen

In the feature selection screen, you can change the location of the SQL binaries. Since I want to keep the C: drive only for the OS, I will install the binaries a separate drive (in my case the G: drive).



Feature Selection

Select the Enterprise features to install.

Features:

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

Shared Features
- □ Reporting Services - SharePoint
- □ Reporting Services Add-in for SharePoint Produ
- □ Data Quality Client
- □ SQL Server Data Tools
- □ Client Tools Connectivity
- □ Integration Services
- □ Client Tools Backwards Compatibility

Feature description:

Each shared feature is installed once within a defined scope and operates within that scope. The defined scope can span all SQL Server versions on a computer (e.g., SQL Server Browser), can be isolated to one major version of SQL Server (e.g., SQL Server Management Tools), or can be isolated to one or more minor versions.

Prerequisites for selected features:

Already installed:
- Microsoft .NET Framework 4.0
- Windows PowerShell 2.0
- Microsoft .NET Framework 3.5

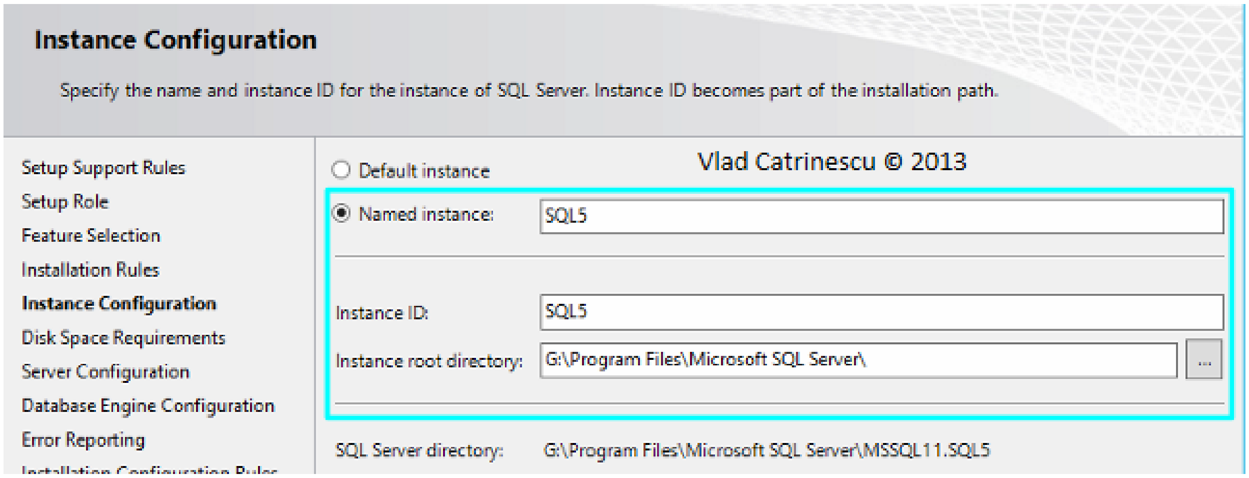
To be installed from media:
- Microsoft Visual Studio 2010 Shell

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

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

Instance Configuration

In the instance configuration page you can either specify to use the default instance (if you don't already have one) or use a named one. Personally, I always use named instances, but you don't have to! Furthermore, in this screen you also need to change the Instance root directory to match the settings from the "Feature Selection" screen. Using a named instance vs Default Instance does not improve performance



Instance Configuration

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

Vlad Catrinescu © 2013

○ Default instance

● Named instance: SQL5

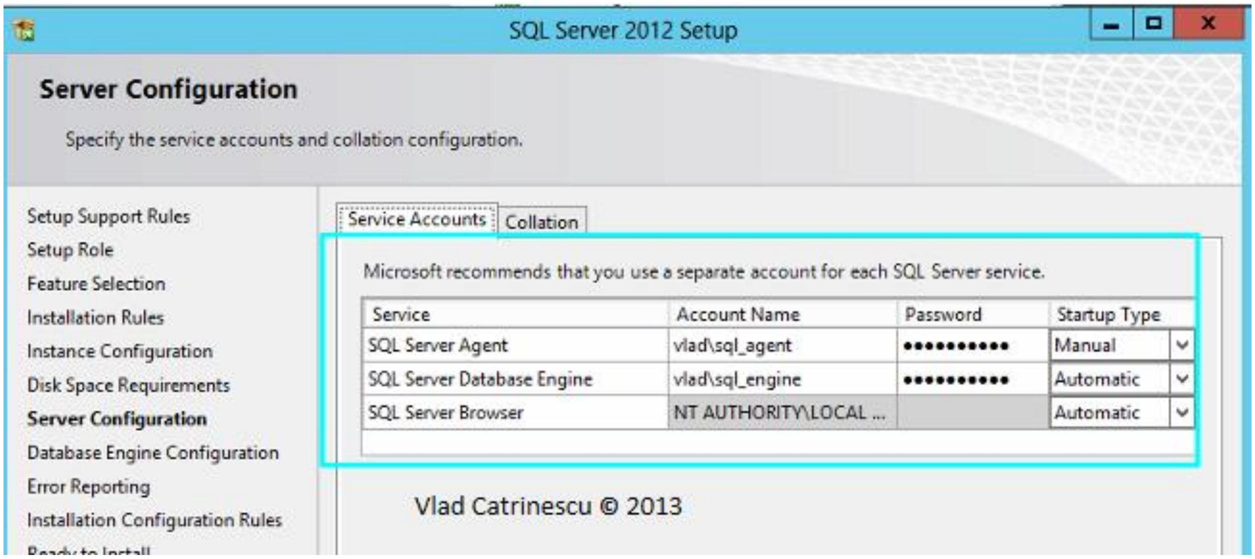
Instance ID: SQL5

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

SQL Server directory: G:\Program Files\Microsoft SQL Server\MSSQL11.SQL5

Server Configuration – Service Accounts

In this screen we will set the Service Accounts for the Agent and Database Engine Services. What I recommend is to have at least one account for both (ex: SQL_Services) or one account per service. This is also not for performance, but for security.



SQL Server 2012 Setup

Server Configuration

Specify the service accounts and collation configuration.

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

Service	Account Name	Password	Startup Type
SQL Server Agent	vlad\sql_agent	●●●●●●●●●●●	Manual
SQL Server Database Engine	vlad\sql_engine	●●●●●●●●●●●	Automatic
SQL Server Browser	NT AUTHORITY\LOCAL ...		Automatic

Vlad Catrinescu © 2013

Server Configuration – Collation

I am sure 95% of you never changed this setting, and it's not fatal. As a matter of fact, SharePoint is built to accept any SQL Server 2012 Collation, however that is not the collation SharePoint server 2013 Uses. SharePoint 2013 uses Latin1_General_CI_AS_KS_WS. This is a quote from Microsoft to prove it:

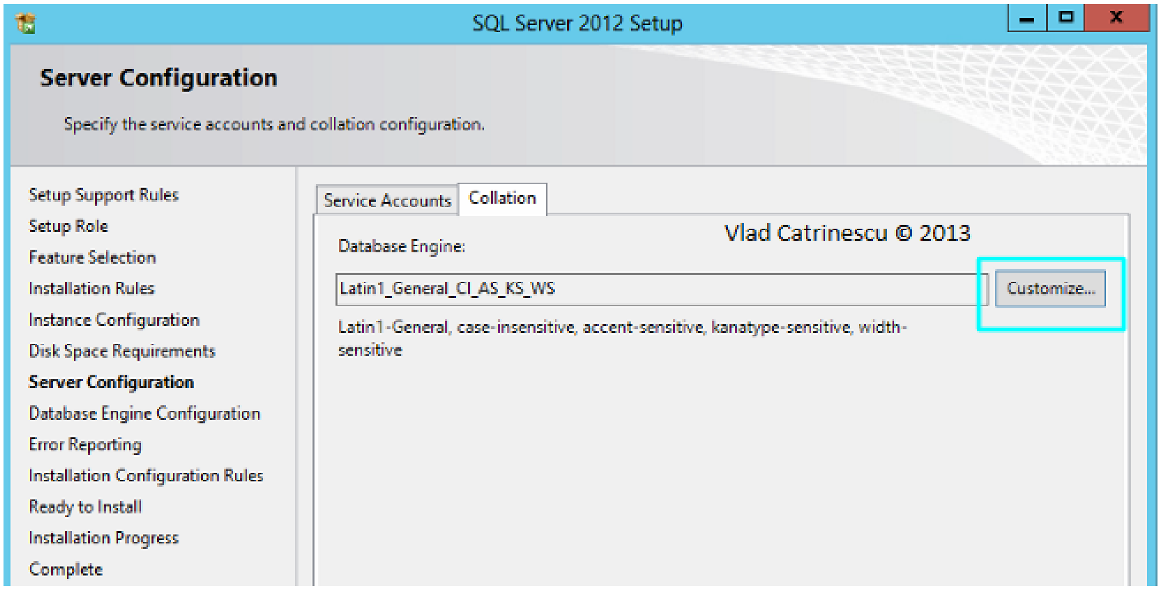
> *"We support any CI collation for the SQL instance (for master, tempdb databases). However we recommend using Latin1_General_CI_AS_KS_WS as the instance default collation (master, tempdb databases)."* [9]

But, what does CI_AS_KS_WS stand for anyway?[10]

- **CI – (Case Insensitive)** A and a ARE treated as the same character.
- **AS – (Accent Sensitive)** a and á are NOT treated as the same character.
- **KS – (Kana Sensitive)** Japanese Hirakana and Katakana characters which look the same are NOT treated as the same character.
- **WS – (Width Sensitive)** Single-Byte and Double-Byte versions of the same character are NOT treated as the same character.

This is another one of those settings that must be done correct from the start as you cannot change the Instance Collation after it's installed. But, how do we do it?

On the Collation page, click the Customize button.



Server Configuration

Specify the service accounts and collation configuration.

Database Engine:

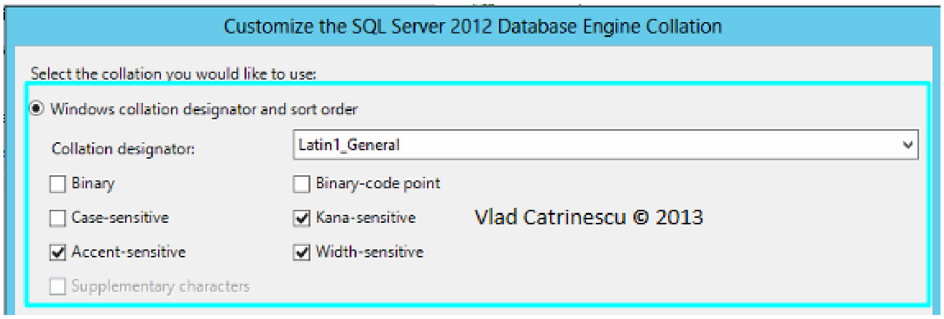
Latin1_General_CI_AS_KS_WS

Latin1-General, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive

In the following Screen select Latin1_General as Collation designator. And check the following boxes:

[9] http://support.microsoft.com/kb/2008668

[10] http://www.darrenmarsden.com/post/2012/03/24/SQL-Collation-settings-for-SharePoint.aspx



Customize the SQL Server 2012 Database Engine Collation

Select the collation you would like to use:

● Windows collation designator and sort order

Collation designator: Latin1_General

☐ Binary ☐ Binary-code point

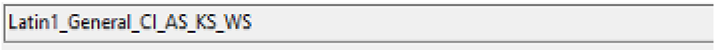
☐ Case-sensitive ☑ Kana-sensitive

☑ Accent-sensitive ☑ Width-sensitive

☐ Supplementary characters

Vlad Catrinescu © 2013

Now, your collation should look like this:

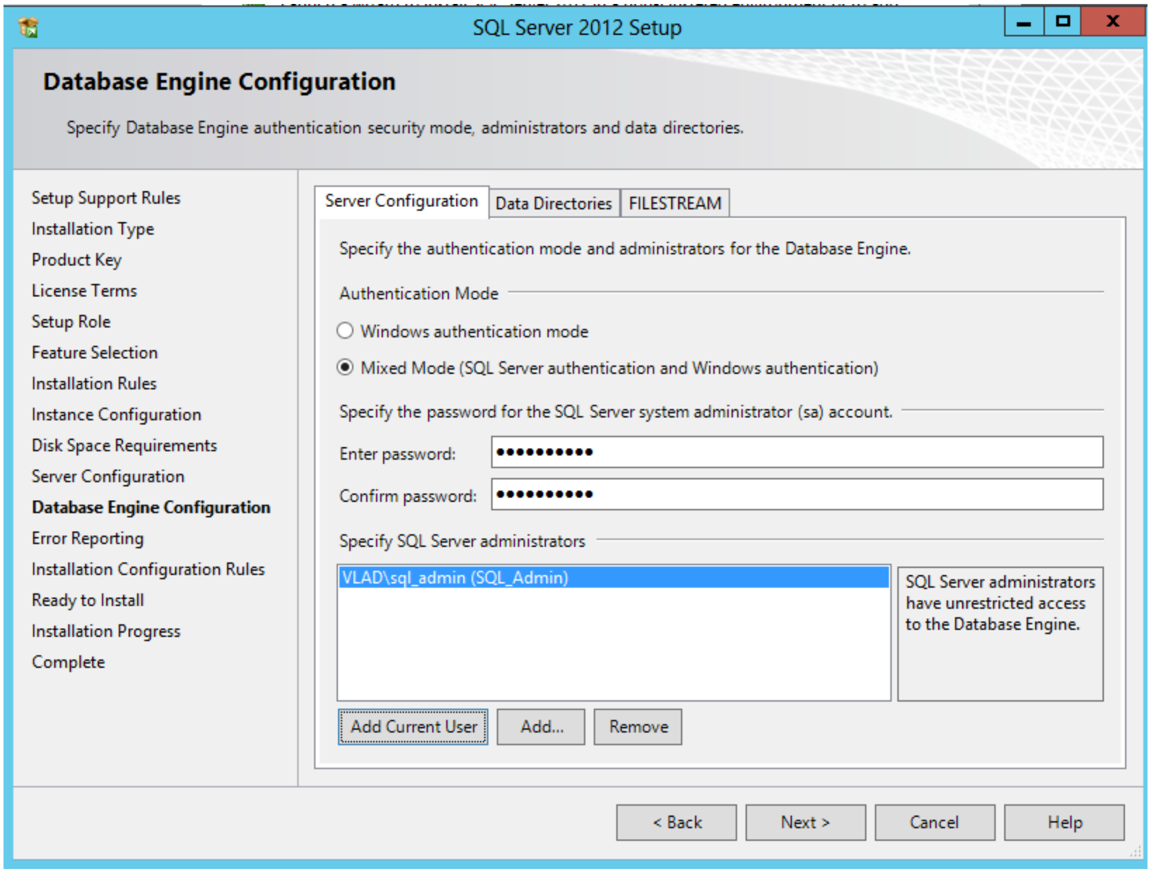


Latin1_General_CI_AS_KS_WS

If you did not do it initially, don't worry because SharePoint 2010/2013 will automatically set the correct collation during database creation. However, if you want your DBA to pre provision your databases, they will need to use the SharePoint collation when creating the databases.

Database Engine Configuration – Server Configuration

In this screen we decide if we only use Windows authentication or Mixed Mode as well as your SQL admins. I strongly suggest to use Mixed Mode and to keep the **sa** account password safe. Use sa only when there is no other way to log in to an instance of SQL (for example, when other system administrators are unavailable or have forgotten their passwords). This is again only for security and not for performance!



SQL Server 2012 Setup

Database Engine Configuration

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

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

○ Windows authentication mode

● Mixed Mode (SQL Server authentication and Windows authentication)

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

Enter password: ••••••••••

Confirm password: ••••••••••

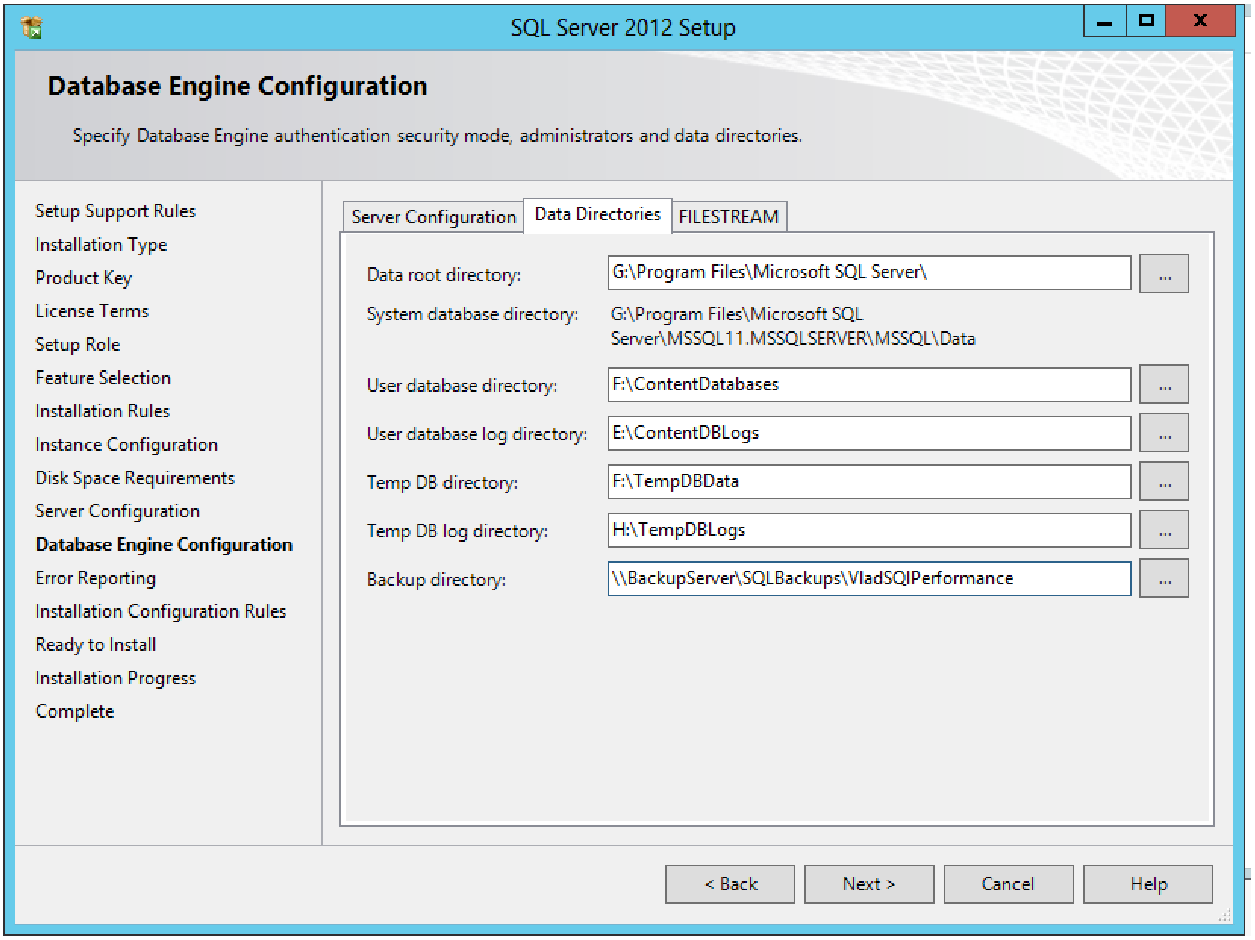
Specify SQL Server administrators

VLAD\sql_admin (SQL_Admin)

SQL Server administrators have unrestricted access to the Database Engine.

Database Engine Configuration-Data Directories

This screen will put in practice everything we planned in first part of the White Paper. Here is where we tell the SQL server the default location for the TempDB, ContentDB's , etc, data and logs. Use what we planned in the first step and configure it!



Database Engine Configuration

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

Data root directory: G:\Program Files\Microsoft SQL Server\

System database directory: G:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data

User database directory: F:\ContentDatabases

User database log directory: E:\ContentDBLogs

Temp DB directory: F:\TempDBData

Temp DB log directory: H:\TempDBLogs

Backup directory: \\BackupServer\SQLBackups\VladSQlPerformance

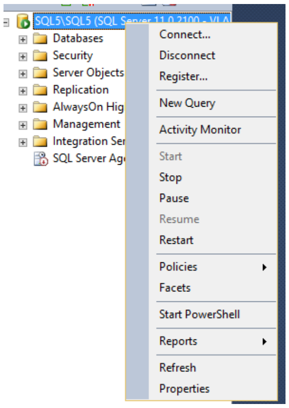
3. Post Installation Configurations

"If there is no struggle, there is no progress." - Frederick Douglass

After the SQL Server installation is done, there are some changes we need to do before installing SharePoint Server 2013 to make sure everything is setup. If you already installed your SharePoint, you can also make those changes now to make it faster. I will try to group them by screen where you make the modification!

Server Properties

To access the Server Property, right click on the server name and click properties



Maximum Server Memory

By Default, SQL Server is set to use max 2TB of Ram, however I am sure that in 2013 no one has that much! This effectively means that SQL can consume all of the RAM in your server leaving nothing for the OS or other applications. This can cause performance issues. Here is how Thomas Larock, from SQL Rockstar explains it:

> *SQL Server (and other database systems such as Oracle and Sybase) need to read data pages into their internal memory before they can be used. Of course your server needs memory to operate as well. When your database engine and your server are competing for the same memory resources, you get bad performance. You want your server and your database engine to be like dancing partners, and less like my kids fighting over the last cupcake.*[11]

There is a nice formula to define how much RAM you should dedicate to all the SQL instances on the server, to make sure there is enough left for the OS but... unfortunately it's not easy!

SQL Max Memory = TotalPhyMem - (NumOfSQLThreads * ThreadStackSize) - (1GB * CEILING(NumOfCores/4))

NumOfSQLThreads = 256 + (NumOfProcessors*- 4) * 8 (* If NumOfProcessors > 4, else 0)

ThreadStackSize = 2MB on x64 or 4 MB on 64-bit (IA64)

[11] http://thomaslarock.com/2012/08/youre-doing-it-wrong-5-factors-that-affect-database-performance/

I know, it's not easy, and what I think it's missing is a part really dedicated for the OS. To help you set the correct values, I developed a utility that automatically calculates this figure. There are some examples shown below.

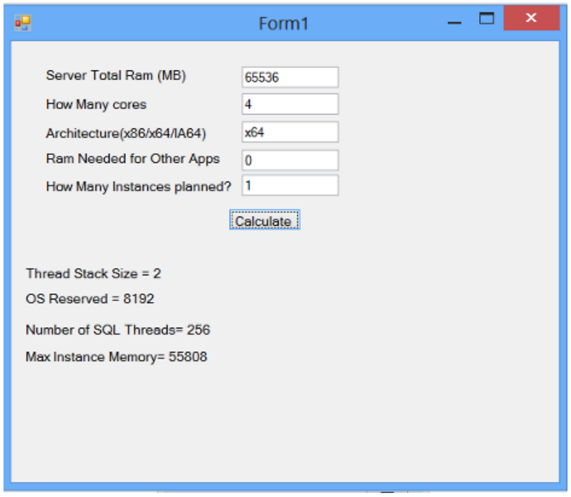
(If there is enough Interest, I will put this tool onto Codeplex. Tweet @vladcatrinescu if you would find this useful).

Here are some examples based on a quad core server on x64 architecture!

Server Ram(IN GB)	Ram for all SQL Instances together (IN MB)
8	5017
12	8294
16	11571
24	19968
32	27136
64	55808

So for example, if you have an 8GB, 4 core, x64 server with two instances, you will need to split the 5017 MB of ram between the 2. Ex: instance 1: 2000, instance 2: 3017

Here is a screenshot from my utility:



Server Total Ram (MB)	65536
How Many cores	4
Architecture(x86/x64/IA64)	x64
Ram Needed for Other Apps	0
How Many Instances planned?	1

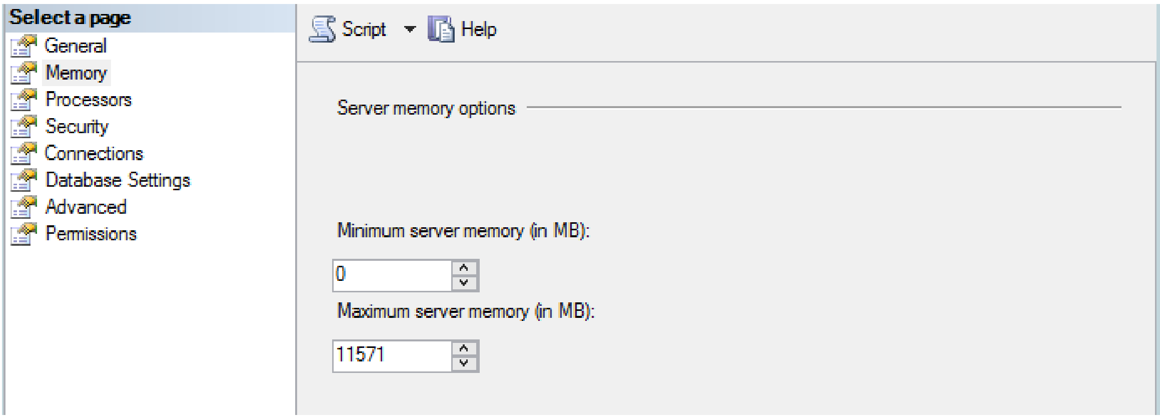
Thread Stack Size = 2

OS Reserved = 8192

Number of SQL Threads= 256

Max Instance Memory= 55808

Here is where you set it:



Server memory options

Minimum server memory (in MB): 0

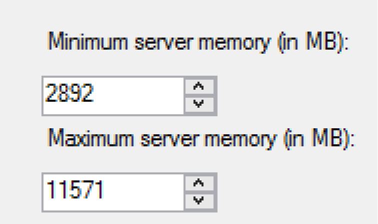
Maximum server memory (in MB): 11571

However, since SQL 2008 R2, you don’t have to set the Max Server Memory if you only have **one** instance. The memory manager component of Microsoft SQL Server eliminates the need for manual management of the memory available to SQL Server. When SQL Server starts, it dynamically determines how much memory to allocate based on how much memory the operating system and other applications are currently using. As the load on the computer and SQL Server changes, so does the memory allocated[12]

Minimum Server Memory

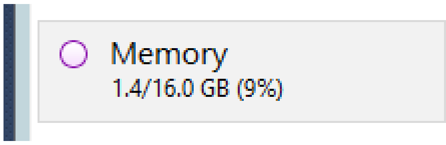
The **min server memory** server configuration option can be used to ensure that SQL Server does not release memory below the configured minimum server memory once that threshold is reached. This configuration option can be set to a specific value based on the size and activity of your SQL Server. If you choose to set this value, set it to some reasonable value to ensure that the operating system does not request too much memory from SQL Server, which can affect SQL Server performance. What I would advise you to do is set it at 25% of the Max server memory for a farm where data requested by SharePoint changes a lot, and to 60% of the Max server memory when the data requested by SharePoint is almost always the same.

Also, note that even if you set the minimum server memory, it does not mean SQL will take automatically all that memory. It will only take it when it needs it! For my SQL, I set it at 25% of the Maximum Server Memory because I expect content to change a lot!



Minimum server memory (in MB):
2892
Maximum server memory (in MB):
11571

In Task manager you can also see that because SharePoint is not actively used at the moment, my server only consumes 1.4 GB.

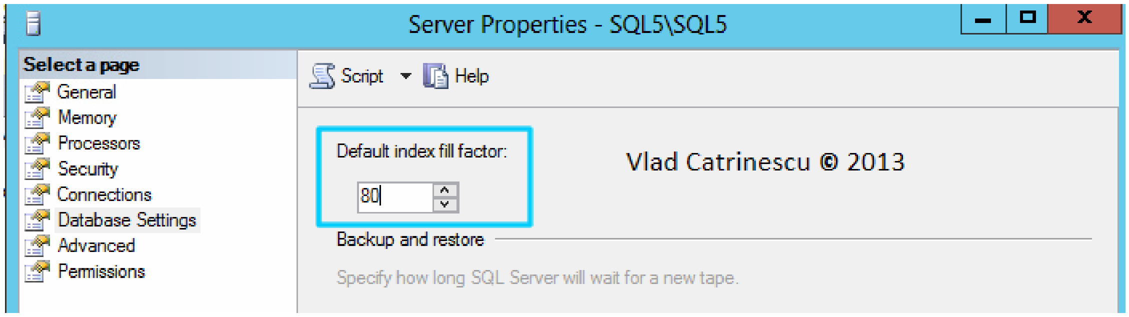


Memory
1.4/16.0 GB (9%)

[12] http://msdn.microsoft.com/en-us/library/ms177455(v=sql.105).aspx

Fill Factor

To further improve index data storage and performance, use fill factor. When indexes are created or rebuilt, the fill factor value (1-100) determines the percentage of space that can be filled with data on each leaf level page. The remaining space is reserved for future growth. For many situations, the default server-wide fill factor level of 0 (fill each page to 100% full) is optimal. However, for SharePoint, a server-wide setting of 80 is optimal to support growth and minimize fragmentation.[13]



Server Properties - SQL5\SQL5

Default index fill factor: 80

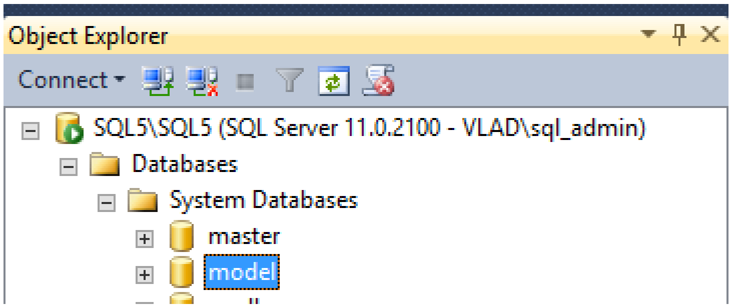
Vlad Catrinescu © 2013

Backup and restore

Specify how long SQL Server will wait for a new tape.

Model Database

SQL Server uses the Model database as a template for creating new user databases. When a new user database is created, SQL Server copies contents of model to the new database and fills the rest of the space with empty data pages. Before even installing SharePoint 2013, we will make the Model DB as good as possible so all the future databases will inherit these settings. However, be careful,   SharePoint does not use ALL the settings, so there are some things you will need to modify later (See more in Section 4).



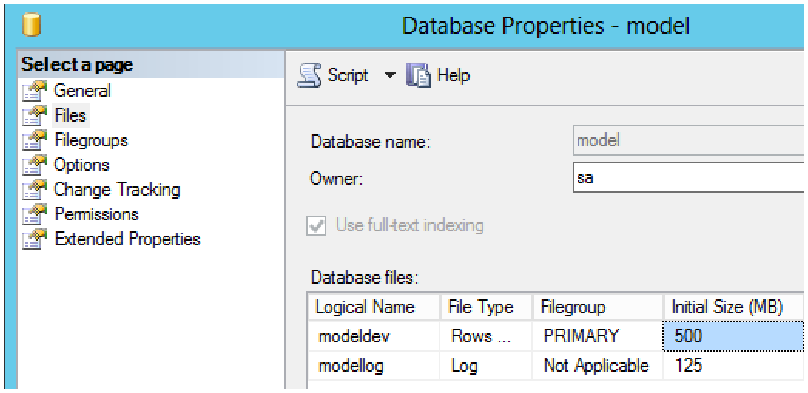
SQL5\SQL5 (SQL Server 11.0.2100 - VLAD\sql_admin)

Initial Size

The initial size is the size of the Databases when they will be created. The default is only set at 3MB. That means that when your content Databases are created, they only take up 10MB. This means every time you do something in SharePoint, such as a adding a document, the Database will have to grow (more info later) before being able to write that data. This is an operation that can be avoided by correctly setting a Database size up front. It is a best practice to set your initial size to how much data you expect to have per content database in a year. Yes, it will take more space on your disks initially, however your performance will be far better.In my example, I estimated that my Content Databases will

[13] http://technet.microsoft.com/en-us/library/cc262731

reach 500MB after the first year, so my Initial Size is 500MB. Furthermore, the log should be at about **25%** of the database initial size.



Database name: model

Owner: sa

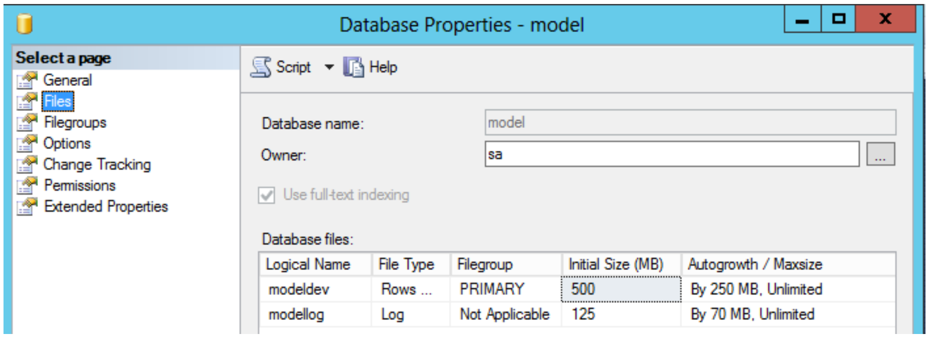
☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)
modeldev	Rows ...	PRIMARY	500
modellog	Log	Not Applicable	125

Auto growth

The auto growth is your insurance policy if your current database size has been reached. Instead of making the Database read only and stopping users from writing to it, SQL allows you to set by how much you want your database to grow when DB size has been reached. By default, for the data files is set at only 1MB, but that is not optimal at all. Why? Imagine you are 1MB away from your size, and user uploads a 10MB document. Your SQL will have to grow 9 different times in order to be able to put the document in the database. That adds tasks to the SQL which will make it slower for the user. Now you might ask, but if we put it at a bigger number, ex: 200MB, won't the user have to wait for it to grow 200MB before being able to write his document? Yes that's exactly true, however in a few pages I will give a trick which will make this auto grow almost instant, and you will only have to grow the database once instead of 9 times. I suggest setting the Auto Growth in MB and not in % because it allows you to monitor and know exactly by how much it will grow. Furthermore, the Auto Growth should be at about 50% of the Initial Size!



Database name: model

Owner: sa

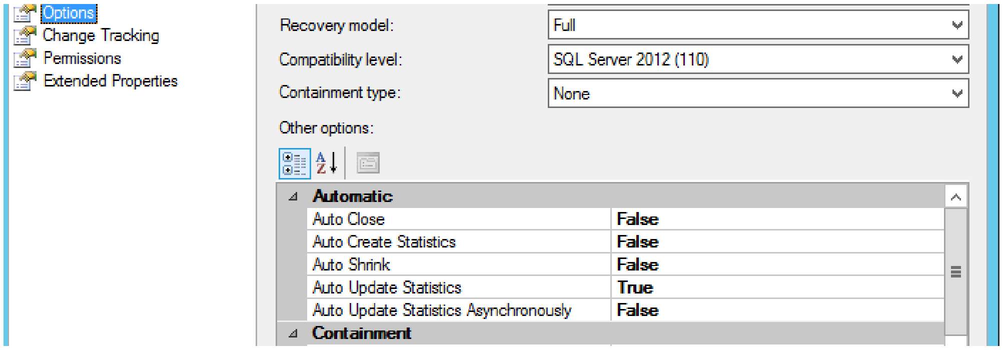
☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
modeldev	Rows ...	PRIMARY	500	By 250 MB, Unlimited
modellog	Log	Not Applicable	125	By 70 MB, Unlimited

Auto Create Statistics

Do not enable auto-create statistics on a server that hosts SQL Server and SharePoint Server. Enabling auto-create statistics is not supported for SharePoint Server. SharePoint Server configures the required settings during provisioning and upgrade. Manually enabling auto-create statistics on a SharePoint database can significantly change the execution plan of a query. The SharePoint databases either use a stored procedure that maintains the statistics (proc_UpdateStatistics) or rely on SQL Server to do this.[14]



Recovery model: Full

Compatibility level: SQL Server 2012 (110)

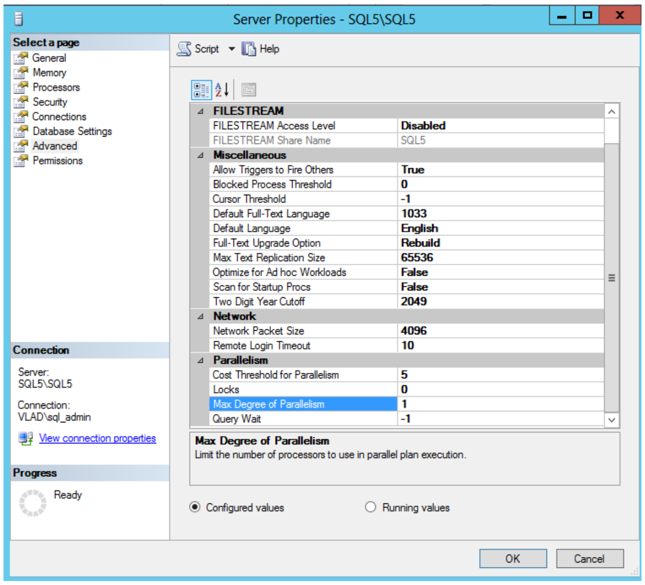
Containment type: None

Other options:

Automatic	
Auto Close	False
Auto Create Statistics	False
Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False
Containment	

Max Degree of Parallelism

In SharePoint 2010 this setting was optional, however in SharePoint 2013 you have to do it or you won't be able to run the Configuration Wizard. Set max degree of parallelism (MAXDOP) to 1 for instances of SQL Server that host SharePoint databases to make sure that a single SQL Server process serves each request. Setting the max degree of parallelism to any other number can cause a less optimal query plan to be used that will decrease SharePoint Server 2013 performance[15].



FILESTREAM	
FILESTREAM Access Level	Disabled
FILESTREAM Share Name	SQL5
Miscellaneous	
Allow Triggers to Fire Others	True
Blocked Process Threshold	0
Cursor Threshold	-1
Default Full-Text Language	1033
Default Language	English
Full-Text Upgrade Option	Rebuild
Max Text Replication Size	65536
Optimize for Ad hoc Workloads	False
Scan for Startup Procs	False
Two Digit Year Cutoff	2049
Network	
Network Packet Size	4096
Remote Login Timeout	10
Parallelism	
Cost Threshold for Parallelism	5
Locks	0
Max Degree of Parallelism	1
Query Wait	-1

Max Degree of Parallelism
Limit the number of processors to use in parallel plan execution.

● Configured values ○ Running values

Connection

Server:
SQL5\SQL5

Connection:
VLAD\sql_admin

[14] http://technet.microsoft.com/en-us/library/hh292622.aspx

[15] http://technet.microsoft.com/en-us/library/hh292622.aspx

Instant File Initialization

When SQL Server increases the size of a file, it must first initialize the new space before it can be used. This is a blocking operation that involves filling the new space with empty pages (zeroes). That means, before SQL can create or auto grow, SQL must first write the size required with zeroes, and then it can save the data. However, since SQL 2005 we can enable “Instant File Initialization”. It's a feature that's seemingly simple; it allows file allocation requests to skip zero initialization on creation. As a result, file allocation requests can occur instantly – no matter what the file size.[16] [17]

Here are some tests done by Kimberly L., author at www.sqlskills.com:

Performance Test with Zero Initialization
Hardware: Dell Precision 670 Dual Proc (x64) with Dual Core, 4 GB Memory, RAID 1+0 array w/4-142 GB, 15000rpm disks
CREATE DATABASE with 20 GB Data file = **14:02 minutes**
ALTER DATABASE BY 10 GB = **7:01 minutes**
RESTORE 30 GB DATABASE (EMPTY Backup) = **21:07 minutes**
RESTORE 30 GB DATABASE (11GB Backup) = **38:28 minutes**

Performance Test with Instant Initialization
Hardware: Dell Precision 670 Dual Proc (x64) with Dual Core, 4 GB Memory, RAID 1+0 array w/4-142 GB, 15000rpm disks

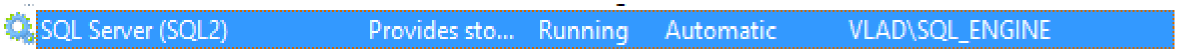
CREATE DATABASE with 20 GB Data file = **1.3 seconds**
ALTER DATABASE BY 10 GB = **.4 seconds**
RESTORE 30 GB DATABASE (EMPTY Backup) = **5 seconds**
RESTORE 30 GB DATABASE (11GB Backup) = **19:42 minutes**

However, Instant file Initialization does not apply to Log Files, and that’s why we still have to set a decent amount as initial size!

As you can see, the differences in time are HUGE, but how do we enable this? It’s super easy.

Instant file initialization is only available if the SQL Server (MSSQLSERVER) service account has been granted SE_MANAGE_VOLUME_NAME. Members of the Windows Administrator group have this right and can grant it to other users by adding them to the Perform Volume Maintenance Tasks security policy. For more information about assigning user rights, see the Windows documentation.

1. Open Services.msc and check what service account is running your SQL Server(Instance Name) service.

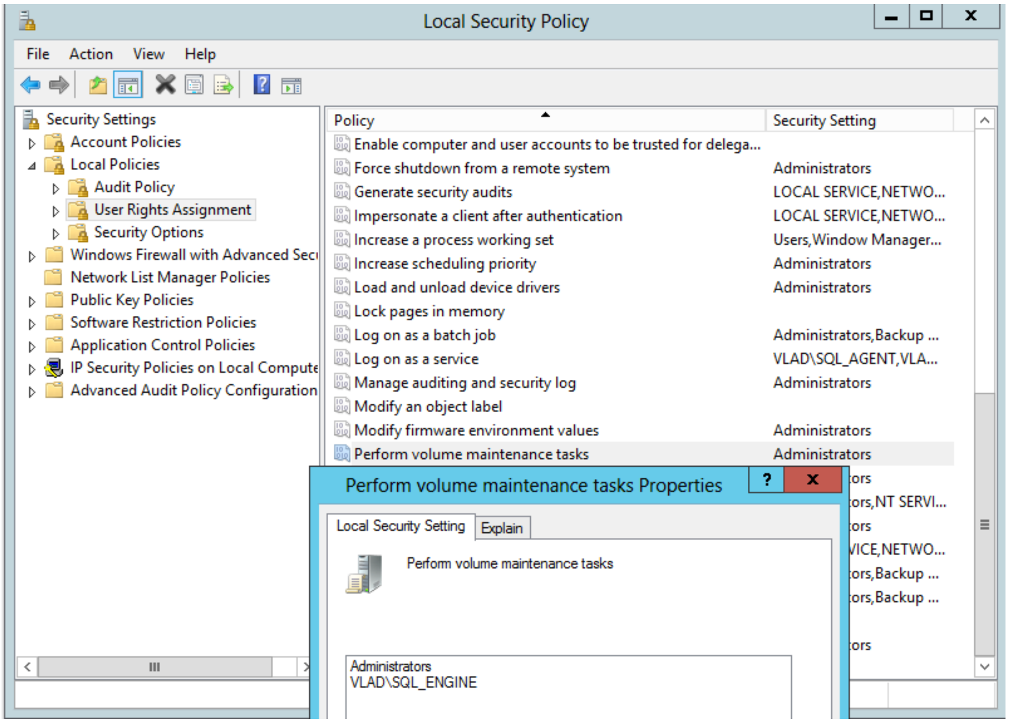


SQL Server (SQL2)   Provides sto...   Running   Automatic   VLAD\SQL_ENGINE

2. Open Local Security Policy> Local Policies > User Rights Management and go to the Perform Volume Maintenance Task. By default, only Local Admins have the access to it. If the service

[16] http://msdn.microsoft.com/en-us/library/gg634626.aspx
[17] http://www.sqlskills.com/blogs/kimberly/instant-initialization-what-why-and-how/

account above is not in your local admins (and it shouldn't), add it.



- Security Settings
  - Account Policies
  - Local Policies
    - Audit Policy
    - User Rights Assignment
    - Security Options
  - Windows Firewall with Advanced Secu
  - Network List Manager Policies
  - Public Key Policies
  - Software Restriction Policies
  - Application Control Policies
  - IP Security Policies on Local Compute
  - Advanced Audit Policy Configuration

Policy	Security Setting
Enable computer and user accounts to be trusted for delega...	
Force shutdown from a remote system	Administrators
Generate security audits	LOCAL SERVICE,NETWO...
Impersonate a client after authentication	LOCAL SERVICE,NETWO...
Increase a process working set	Users,Window Manager...
Increase scheduling priority	Administrators
Load and unload device drivers	Administrators
Lock pages in memory	
Log on as a batch job	Administrators,Backup ...
Log on as a service	VLAD\SQL_AGENT,VLA...
Manage auditing and security log	Administrators
Modify an object label	
Modify firmware environment values	Administrators
Perform volume maintenance tasks	Administrators
	ors
	ors,NT SERVI...
	ors
	VICE,NETWO...
	ors,Backup ...
	ors,Backup ...
	ors

Perform volume maintenance tasks Properties

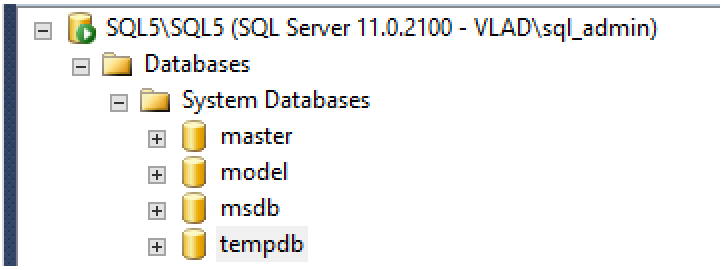
Local Security Setting | Explain

Perform volume maintenance tasks

Administrators
VLAD\SQL_ENGINE

3. You will then need to restart the SQL Server service in order for the changes to take effect.

TempDb



- SQL5\SQL5 (SQL Server 11.0.2100 - VLAD\sql_admin)
  - Databases
    - System Databases
      - master
      - model
      - msdb
      - tempdb

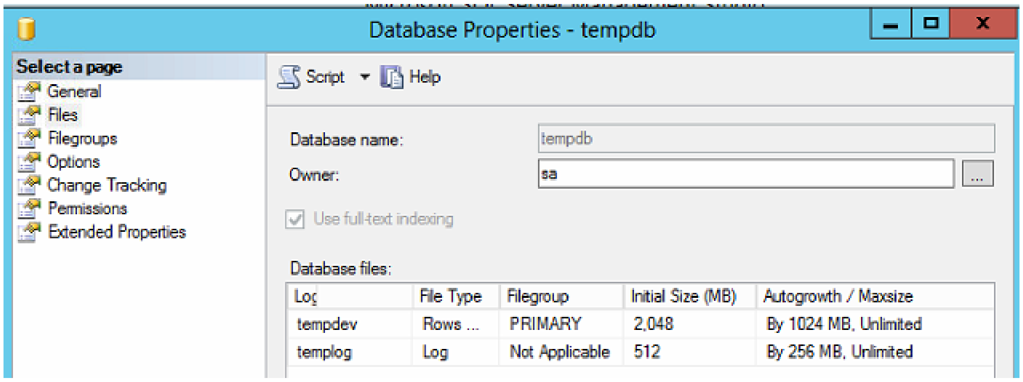
The **tempdb** system database is a global resource that is available to all users connected to the instance of SQL Server and is used to hold the following:

- Temporary user objects that are explicitly created, such as: global or local temporary tables, temporary stored procedures, table variables, or cursors.
- Internal objects that are created by the SQL Server Database Engine, for example, work tables to store intermediate results for spools or sorting.
- Row versions that are generated by data modification transactions in a database that uses read-committed using row versioning isolation or snapshot isolation transactions.
- Row versions that are generated by data modification transactions for features, such as: online index operations, Multiple Active Result Sets (MARS), and AFTER triggers.[18]

[18] http://msdn.microsoft.com/en-ca/library/ms190768.aspx

Initial Size & Autogrow

The size of **tempdb** can affect the performance of a system. For example, if the **tempdb** size is too small, the system processing could be too occupied with autogrowing the database to support your workload requirement every time that you start SQL Server. You can avoid this overhead by increasing the size of **tempdb.** This Database is the busiest database of your instance, so don't be shy to give it a good initial size. I gave it a 2048MB Initial size and 1024MB autogrow. Thanks to Instant File initialization, the autogrow is done almost instantly and because it pretty high, the database can actually be busy writing/reading files instead of autogrowing!



Database name: tempdb

Owner: sa

☑ Use full-text indexing

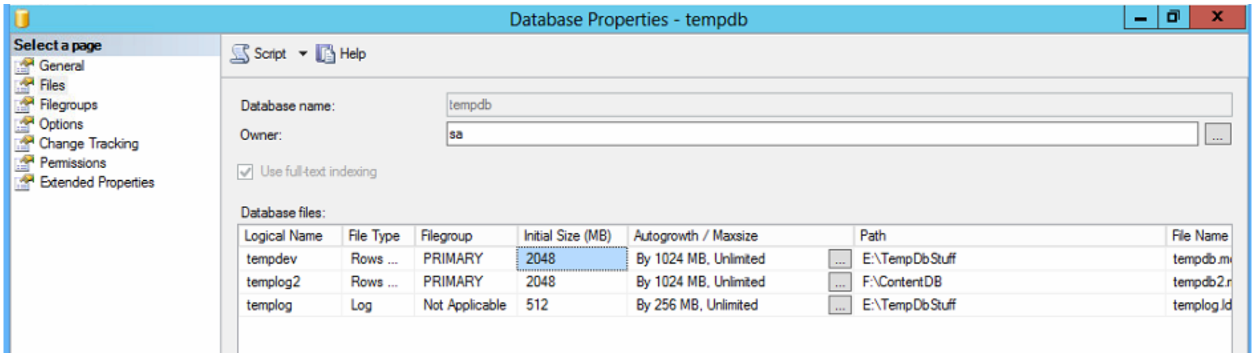
Database files:

Loç	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
tempdev	Rows ...	PRIMARY	2,048	By 1024 MB, Unlimited
templog	Log	Not Applicable	512	By 256 MB, Unlimited

Multiple Files

By now, we know that the tempdb is the busiest database and needs to be on the fastest drive. But what if we create multiple files on multiple drives? That would allow the SQL Server to write in both files at the same time, therefore increasing performance. (Think of it as a TempDB Raid 0). The general recommendation is 1 File/ CPU therefore, for a Dual Core CPU, you would have 2 Files. However, although Microsoft still suggests that, most SQL experts think you should only allocate 1 File per 2/4 CPU's.

Important Note: Make sure all your data files have the same initial size and Auto growth settings! This will allow optimal proportional-fill performance.



Database name: tempdb

Owner: sa

☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize		Path	File Name
tempdev	Rows ...	PRIMARY	2048	By 1024 MB, Unlimited	...	E:\TempDbStuff	tempdb.m
templog2	Rows ...	PRIMARY	2048	By 1024 MB, Unlimited	...	F:\ContentDB	tempdb2.r
templog	Log	Not Applicable	512	By 256 MB, Unlimited	...	E:\TempDbStuff	templog.ld

Other Considerations

- Avoid shrinking TempDB (or any database) files unless you are very certain you will never need the space again.
- Do not change collation from the SQL Server instance collation.
- Do not change the database owner from sa.
- Do not drop the TempDB database.
- Do not drop the guest user from the database.
- Do not change the recovery model from SIMPLE.
- Ensure the disk drives TempDB resides on have RAID protection i.e. 1, 1 + 0 or 5 in order to prevent a single disk failure from shutting down SQL Server. Keep in mind that if TempDB is not available then SQL Server cannot operate.[19]

Install SharePoint 2013

After all these optimizations, it's finally the time to install SharePoint Server 2013! Make sure you created the proper SharePoint 2013 Service Accounts and you gave appropriate permissions!

[19] http://www.mssqltips.com/sqlservertip/1432/tempdb-configuration-best-practices-in-sql-server/

4. Maintenance Tasks

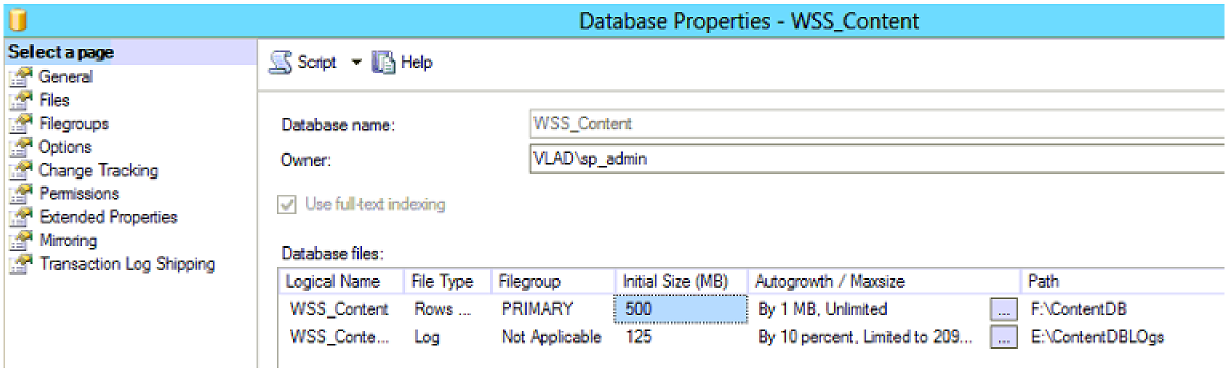
"I know that the hard work got me here. And the day I stop working hard, this can all go away." - Kevin Durant

You have now installed SharePoint 2013 and your users are amazed at its speed. They're uploading, downloading documents, use SharePoint's social features every day and they are happy! Unfortunately, your job as a SharePoint Admin/ DBA is not over. Think of your SQL as a nice sports car, if you don't change the oil and maintain it properly, its performance will slowly degrade until it becomes slower than a scooter.

However, because we planned everything correctly and all the Post Installation Configurations were done right, we won't need to do this too often.

The SharePoint Databases Autogrow "Feature"

Remember that in the Post Installation Configurations, we modified our model database and changed the autogrow settings? Well, when you create a SharePoint Content Database or Service Application Database, **SharePoint 2013(and 2010) only copies the Initial size setting, but not the autogrow.**



Database name: WSS_Content

Owner: VLAD\sp_admin

☑ Use full-text indexing

Database files:

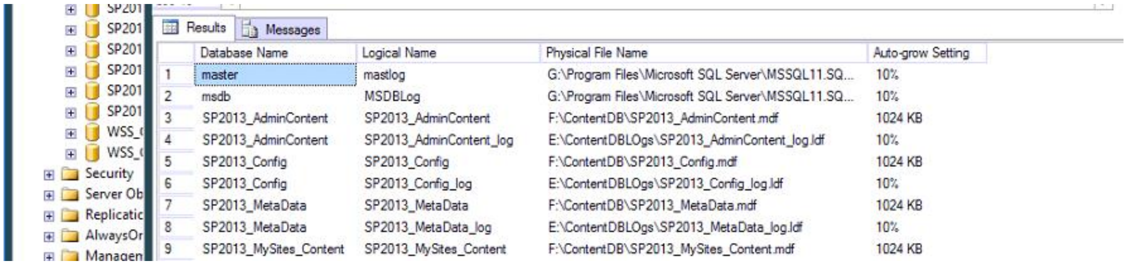
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize	Path
WSS_Content	Rows ...	PRIMARY	500	By 1 MB, Unlimited ...	F:\ContentDB
WSS_Conte...	Log	Not Applicable	125	By 10 percent, Limited to 209... ...	E:\ContentDBLOgs

However, since the initial size of our Databases was set to accommodate one year of content, you won't have to modify the setting after every time you create a database, but try to do it monthly. You might ask, why aren't the SharePoint databases also copying the autogrow settings? Most SharePoint professionals would agree it's a bug, however if you would ask a developer I am sure they would answer this:

Now you might wonder if every month, you have to check every database's autogrow settings because you don't remember which ones you changed last month. I got good news for you! Greg Larsen from SimpleTalk posted a very cool script that shows all the databases that had the default autogrow settings! Here it is: [20]

```
-- Drop temporary table if it exists
IF OBJECT_ID('tempdb..#info') IS NOT NULL
       DROP TABLE #info;
-- Create table to house database file information
CREATE TABLE #info (
     databasename VARCHAR(128)
     ,name VARCHAR(128)
    ,fileid INT
    ,filename VARCHAR(1000)
    ,filegroup VARCHAR(128)
    ,size VARCHAR(25)
    ,maxsize VARCHAR(25)
    ,growth VARCHAR(25)
    ,usage VARCHAR(25));
-- Get database file information for each database
SET NOCOUNT ON;
INSERT INTO #info
EXEC sp_MSforeachdb 'use ?
select ''?'',name,  fileid, filename,
filegroup = filegroup_name(groupid),
''size'' = convert(nvarchar(15), convert (bigint, size) * 8) + N'' KB'',
''maxsize'' = (case maxsize when -1 then N''Unlimited''
else
convert(nvarchar(15), convert (bigint, maxsize) * 8) + N'' KB'' end),
''growth'' = (case status & 0x100000 when 0x100000 then
convert(nvarchar(15), growth) + N''%''
else
convert(nvarchar(15), convert (bigint, growth) * 8) + N'' KB'' end),
''usage'' = (case status & 0x40 when 0x40 then ''log only'' else ''data
only'' end)
from sysfiles';
-- Identify database files that use default auto-grow properties
SELECT databasename AS [Database Name]
      ,name AS [Logical Name]
      ,filename AS [Physical File Name]
      ,growth AS [Auto-grow Setting] FROM #info
WHERE (usage = 'data only' AND growth = '1024 KB')
   OR (usage = 'log only' AND growth = '10%')
ORDER BY databasename
-- get rid of temp table
DROP TABLE #info;
```

[20] https://www.simple-talk.com/sql/database-administration/sql-server-database-growth-and-autogrowth-settings/



	Database Name	Logical Name	Physical File Name	Auto-grow Setting
1	master	mastlog	G:\Program Files\Microsoft SQL Server\MSSQL11.SQ...	10%
2	msdb	MSDBLog	G:\Program Files\Microsoft SQL Server\MSSQL11.SQ...	10%
3	SP2013_AdminContent	SP2013_AdminContent	F:\ContentDB\SP2013_AdminContent.mdf	1024 KB
4	SP2013_AdminContent	SP2013_AdminContent_log	E:\ContentDBLOgs\SP2013_AdminContent_log.ldf	10%
5	SP2013_Config	SP2013_Config	F:\ContentDB\SP2013_Config.mdf	1024 KB
6	SP2013_Config	SP2013_Config_log	E:\ContentDBLOgs\SP2013_Config_log.ldf	10%
7	SP2013_MetaData	SP2013_MetaData	F:\ContentDB\SP2013_MetaData.mdf	1024 KB
8	SP2013_MetaData	SP2013_MetaData_log	E:\ContentDBLOgs\SP2013_MetaData_log.ldf	10%
9	SP2013_MySites_Content	SP2013_MySites_Content	F:\ContentDB\SP2013_MySites_Content.mdf	1024 KB

As you can see, I didn’t do my maintenance yet and I got some work to do.

You can also check the autogrow settings directly from your Windows 7/8 PC. There is a Great post on SharePoint-Community.Net done by Jesper M. Christensen that explains how to do it. Check it out here: Checking SQL database db and log file sizes and growth with PowerShell - from your Windows 7/8

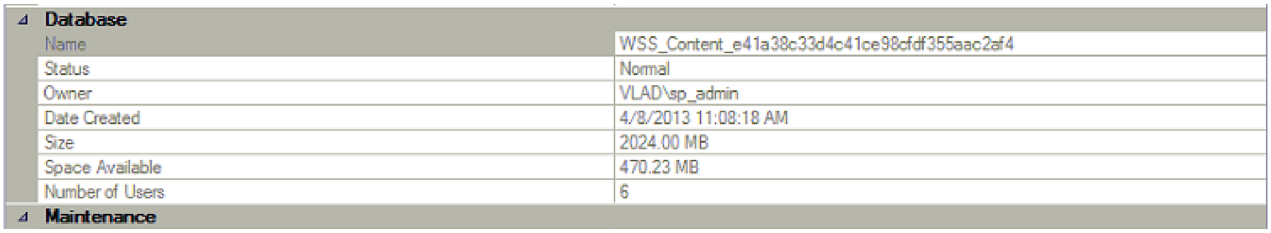
The Autogrow, Your insurance Policy

Now, I talked to you about Autogrow in two sections already and hopefully you understand that it’s important to have it set correctly. Well, I honestly hope you never have a need to autogrow your databases! Yes, you read it right. Autogrow is not a permanent solution, it’s simply an insurance policy if your databases reaches your Initial Size.

If you have autogrow turned off, and your SharePoint Content Database reaches initial size you set, SQL will make it read only. All the settings were to make sure that even if your Content DB reaches initial size, your SQL will continue to operate fast by autogrowing a big enough chunk to last a while. So basically, when you got that nice sports car at the beginning of the chapter, you insured it right? You wouldn’t want a guy with a 1998 Honda Civic hit you in the parking and run away while you have to pay probably 10 000$ + for a plastic bumper. Even if you hope you never use your insurance policy, it’s there

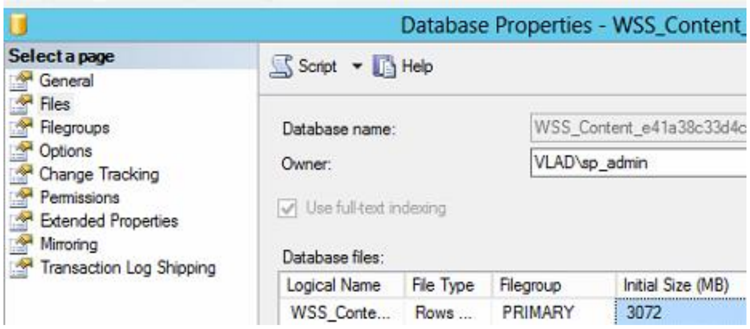
But how do we prevent our autogrow to kick in and save us? We have to pre-size the databases. Remember I told you to make your initial size big enough to accommodate how much data you will have in one year. I could have said 1 month, 6 months, but I suggested one year so you won’t have to check if you need to presize your databases too often. So, if you planned successfully, you only need to do this once per year, however checking it every 2-3 months will make sure no unexpected things happen. Let’s see how we pre-size a database!

In SQL Manager, select a content database and check out the properties



Database	
Name	WSS_Content_e41a38c33d4c41ce98cfdf355aac2af4
Status	Normal
Owner	VLAD\sp_admin
Date Created	4/8/2013 11:08:18 AM
Size	2024.00 MB
Space Available	470.23 MB
Number of Users	6
Maintenance	

In this database I see that it takes 2024 MB on disk, and it has 470MB free. That automatically tells me that during the year, it had to autogrow twice (1024 Initial + 500 + 500 autogrow). That means, in one year the content database actually got to ~ 1.5GB instead of the 1GB I originally planned. So for my next year, I will prepare for another 2GB of content. I see that my Database is currently at 2024 MB (~ 1.5 Taken), so by going into the Files Tab, I change my Initial Size to 3072 and Click OK.



Database name: WSS_Content_e41a38c33d4c

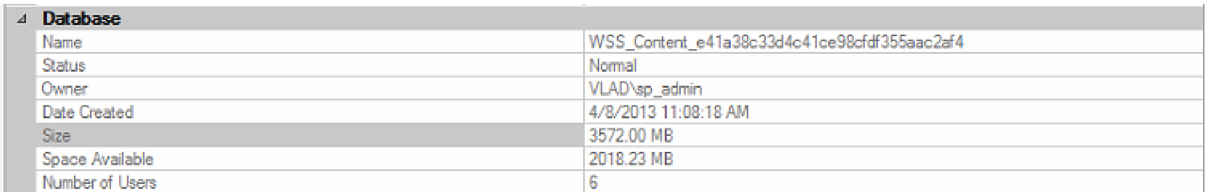
Owner: VLAD\sp_admin

☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)
WSS_Conte...	Rows ...	PRIMARY	3072

Now we go back in the General Tab in Database Properties and we see that I successfully got my database to 3572.00 MB and I got almost 2GB of free space ready for users to collaborate fast!



Database	
Name	WSS_Content_e41a38c33d4c41ce98cfdf355aac2af4
Status	Normal
Owner	VLAD\sp_admin
Date Created	4/8/2013 11:08:18 AM
Size	3572.00 MB
Space Available	2018.23 MB
Number of Users	6

Shrinking the log files

The database transaction log files seems to keep on growing - especially if you do not back these up. This is because every transaction is put in these and not “flushed” until you do a backup. When you perform a backup the file gets “empty” and is ready for new transactions. Performing a manual Shrink creates fragmentation in the database and that decreases performance significantly. Creating a good maintenance plan is advised!

Maintenance Plan

You will have to create a maintenance plan to backup databases, rebuild indexes and checking the logical and physical integrity of your database. I will not go in details in this topic because it is not in the scope of the whitepaper however, do know that putting a great maintenance plan will save you problems down the road!

Thank you

Thank you very much for reading this Whitepaper! I really hope this has been informative and that will help you achieve maximum performance in your SharePoint Farm. For any questions or opinions, either send me an email or tweet @vladcatrinescu. Also, don't forget to join www.SharePoint-Community.Net, the Best SharePoint place to hang out!

Works Cited

"Best practices for SQL Server in a SharePoint Server farm." <http://technet.microsoft.com/en-US/library/hh292622(v=office.15).aspx>.

"Capacity management and sizing overview for SharePoint Server 2013." <http://technet.microsoft.com/en-US/library/ff758647(v=office.15).aspx>.

"Database maintenance for SharePoint 2010 Products." <http://technet.microsoft.com/en-us/library/cc262731>.

"Defining Auto-Growth Settings for Databases." <http://msdn.microsoft.com/en-us/library/gg634626.aspx>.

"Hardware and software requirements for SharePoint 2013." <http://technet.microsoft.com/en-ca/library/cc262485.aspx>.

"Instant Initialization - What, Why and How? - Kimberly L. Tripp." Kimberly L. Tripp. 26 Mar. 2013 <http://www.sqlskills.com/blogs/kimberly/instant-initialization-what-why-and-how/>.

"SQL Collation settings for SharePoint." Darren Marsden. 26 Mar. 2013 <http://www.darrenmarsden.com/post/2012/03/24/SQL-Collation-settings-for-SharePoint.aspx>.

"SQL Server Database Growth and Autogrowth Settings." SQL Server Database Growth and Autogrowth Settings. 09 Apr. 2013 <https://www.simple-talk.com/sql/database-administration/sql-server-database-growth-and-autogrowth-settings/>.

"Storage and SQL Server capacity planning and configuration."

"Supportability regarding SQL collation for SharePoint Databases and TempDB." Supportability regarding SQL collation for SharePoint Databases and TempDB. 26 Mar. 2013 <http://support.microsoft.com/kb/2008668>.

"Tempdb Configuration Best Practices in SQL Server." Tempdb Configuration Best Practices in SQL Server. 09 Apr. 2013 <http://www.mssqltips.com/sqlservertip/1432/tempdb-configuration-best-practices-in-sql-server/>.

"Tempdb Database." <http://msdn.microsoft.com/en-ca/library/ms190768.aspx>.

"Tuning SQL Server 2012 for SharePoint 2013 Jump Start." Tuning SQL Server 2012 for SharePoint 2013. Microsoft Virtual Academy. 09 Apr. 2013 <http://www.microsoftvirtualacademy.com/training-courses/tuning-sql-server-2012-for-sharepoint-2013-jump-start>.

"You're Doing it Wrong: 5 Factors That Affect Database Performance." SQLRockstar Thomas LaRock. 09 Apr. 2013 <http://thomaslarock.com/2012/08/youre-doing-it-wrong-5-factors-that-affect-database-performance/>.

Vlad Catrinescu
www.absolute-sharepoint.com
www.sharepoint-community.net