

# AccessData LAB



## Microsoft SQL Server Performance Considerations

Version: 6.2.x



# AccessData Legal and Contact Information

Document date: March 22, 2017

## Legal Information

©2017 AccessData Group, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

AccessData Group, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, AccessData Group, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, AccessData Group, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, AccessData Group, Inc. reserves the right to make changes to any and all parts of AccessData software, at any time, without any obligation to notify any person or entity of such changes.

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

AccessData Group, Inc.  
588 West 400 South Suite 350  
Lindon, UT 84042  
USA

## AccessData Trademarks and Copyright Information

The following are either registered trademarks or trademarks of AccessData Group, Inc. All other trademarks are the property of their respective owners.

AccessData®	DNA®	PRTK®
AccessData Certified Examiner® (ACE®)	Forensic Toolkit® (FTK®)	Registry Viewer®
AD Summation®	Mobile Phone Examiner Plus®	Summation®
Discovery Cracker®	MPE+ Velocitor™	SilentRunner®
Distributed Network Attack®	Password Recovery Toolkit®	

# Microsoft SQL Server Performance Considerations

AccessData recommends you consult a Database Administrator (DBA) for setup and continuing operations of your SQL Server.

## Microsoft SQL Server Setup

It is important to set up SQL Server properly from the initial installation to keep the system running at optimal performance, prevent unrecoverable events, and plan for growth.

### Initial Setup

This section will discuss important settings in the initial setup of SQL 2012 R2. When SQL Server is initially installed, these are the default settings that are of importance, as well as what DBConfig will set them to.

	SQL Server Default	DBConfig Default
Memory	Max Memory Setting is initially set at over 2,000 TB	Max Memory Setting to 50% of the available RAM on the system
tempdb	tempdb is set at a single file initial size of 8 MB with a growth rate of 10%	tempdb is spread over 8 files and pre-sized to 2 GB with a growth setting of 64 MB
Model	Model Database has an initial size of 4 MB with a growth rate of 1 MB	Model Database Auto-growth is changed to 64 MB

There are several factors that affect the performance of the SQL Server. In the next section, we will discuss areas that affect performance in order of importance.

### SQL Server Processors

The processing of queries is the most important feature of a SQL Server. With Intel processors, you will want to do higher Base Clock Speed (not the Turbo number) and more physical cores for best performance. Other than Clock Speed, keep an eye on the L3 Cache size. This is the memory right next to the processor. If you have 2 processors with the same Base Clock Speed, the one with the higher L3 Cache will run faster.

The amount of physical cores required for your SQL Server will be determined by the number of Distributed Processing Engines (DPEs) in your environment. The method of determining the amount of physical cores

needed on the SQL Server is derived by adding up all of the physical cores of all of the DPEs in your environment, then multiplying that number by 25%-50%. The resulting number will be the amount of cores for your SQL Server.

For Example: You have 16 DPEs with 12 cores each. The SQL Server Core recommendation would be:

$$16(\text{DPE}) * 12(\text{Physical Core}) * .25 (25\%) = 48 \text{ to } 96 \text{ cores in the database}$$

In large installs (above 10 DPEs) you can trend toward 25% of the range as there would be more slack for the database to catch up.

**Important:** This recommendation is specific to the 6.2 version of AccessData's engine enhancements.

## RAM

Microsoft SQL Server's memory management performs well on its own as long as the instance's Max Memory is configured to leave some memory resources available for other applications and the operating system to operate. As stated above, this setting is altered at initial install, but this modification only occurs during the initial database setup. This can be adjusted after initial preparation by a database administrator and should be checked by your DBA.

The performance team at AccessData recommends allocating 4 GB of memory for each Physical Core on your SQL Server.

## Storage

Under normal conditions, SQL Server will read and write heavily during evidence loads. Before we get into storage considerations, let's go over a description of what databases exist in a standard install and their purpose.

**Note:** These recommendations assume you are using the latest hardware with minimal network latency and disk I/O bottlenecks.

There are 3 main areas where SQL Server will do heavy read and write activities to disk: Case Databases' Primary and Secondary data files (MDF, NDF), Case Databases' Transaction Logs (LDF), and the instance's tempdb. Ideally, these three file categories should all be located on separate physical disk arrays.

- Case Databases will experience heavy random read and writes. AccessData recommends locating the Case Databases' Primary and Secondary data files on either a RAID 5 or RAID 10 storage array; AccessData recommends RAID 10.
- Transaction Logs will experience sequential writes. AccessData recommends locating the Transaction Logs on either a RAID 1 or RAID 10 storage array.
- tempdb will experience heavy random read and writes. Because tempdb is temporary storage, AccessData recommends locating the tempdb on a RAID 0 storage array or SSD for best performance. However, this recommendation allows for no fault tolerance, so if fault tolerance is required, AccessData recommends locating the tempdb on a RAID 10 storage array.

## System Databases

The following are the different types of databases and their roles within SQL Server.

## master DB

Records the master database records of all of the system-level information for an instance of SQL Server. It is also the database that stores information regarding the existence of all other databases within an instance and the locations of those databases' files. Consequently, a SQL instance cannot start if the master database is damaged or otherwise unavailable.

## msdb

The msdb database is used by SQL Server Agent for scheduling alerts and jobs and by other functionality such as Microsoft SQL Server Management Studio and Database Mail.

## model DB

Model Database is used as a template for new databases when created within an instance. AccessData currently does not support alterations of model database settings, so changes to this database will not be reflected in new cases.

## tempdb

tempdb is a globally-available workspace for holding temporary objects or intermediate result sets. Examples of what is stored here would be temporary tables and variables.

- While there is constant debate regarding how to best set the actual count, size, and growth rate of tempdb data files, no one disputes the relationship tempdb has to performance.
- AccessData will create 8 tempdb data files pre-sized to 2 GB within a SQL instance. The number of data files and their size should be reviewed after large runs of data through AccessData's evidence processor. You will want to increase the amount of tempdb data files until you reach a point where your disk queues stay under 1; with fewer tempdb data files you will not be maxing out the I/O. Upon review of the disk queues, if they are sustained above 2 or more you will want to decrease the number of tempdb data files, increase the speed of the storage where tempdb data files are stored, or segregate the I/O onto a different array away from other high I/O activities. A healthy SQL Server will use the existing tempdb data files without any growth; uneven growth of the tempdb data files is a symptom of a problem.
- Pre-sizing a database allocates all the disk space as requested in an effort to prevent growth. AccessData currently allocates 2048 MB per tempdb data file on initial configuration.
- The default growth rate of the tempdb data files is 10%. Starting at 2048 MB, growth of 20 MB wouldn't cause delay in operations; however, the larger the data file, the more time it will take to allocate space. For example, if your database is at 10 GB, now you are allocating 1 GB with each successive growth taking more time. It is rare for this setting to change performance on properly pre-sized tempdb data files.

## User Databases

### ADG

The ADG database (sometimes referred to as the "Shared" database) is the primary application database, storing settings, users, permissions, and many other items necessary to run your application.

The ADG database can also be used to host cases, but such a utilization carries with it a number of potential limitations, including performance concerns, disaster recovery complications, and general ease of use. AccessData recommends that for optimal performance the ADG database should not be used to host cases. If you have cases hosted inside of the ADG database, they can be moved with a backup/restore operation. Please contact AccessData Support for detailed steps and assistance.

It is critical to perform maintenance on your ADG Database. There are 2 main areas where maintenance should be done:

- SQL Maintenance, as mentioned below
- Application Maintenance  
This includes clearing the system logs and activity logs. Allowing these to grow indefinitely will eventually cause performance problems. The information contained in the activity log is redundant, as the case audit log will show what was done on a case. If you wish to back it up, there is a CSV save option at the bottom of the page. The information contained in the system log relates to system status information. This is also redundant to the logs contained as text files on the services server.

## Case Databases

The Case Databases, as the name suggests, contain information specific to a single case or project within the system.

- SQL Server Maintenance, as mentioned below, should be performed regularly.
- Consider pre-sizing your database prior to loading a large piece of evidence. We have published guidelines on the expected growth of a database which was one-third the size of the evidence. This number will change based on the data ingested, requiring the one-third to be adjusted. If you wish to re-size your database to 2 TB, 10% auto-growth will attempt to allocate 2 GB if the database has to grow.

## Web Application Databases (Only for Summation and AD eDiscovery)

### Workflow

- Standard SQL Maintenance, as mentioned below

### Workflow40

- Standard SQL Maintenance, as mentioned below
- Work Order Table Maintenance

If you have had a problem or have a system that has been active for an extended amount of time, your work order tables may have exceeded the size necessary to be easily retrieved into the user interface. The clean-up tool can help with two of these problems: A rogue re-index job that continues to generate orders, and a rogue user that has piled up bulk print jobs.

### AD eDiscovery

- Standard SQL Maintenance, as mentioned below

### Infrastructure

- Standard SQL Maintenance, as mentioned below

# Microsoft SQL Server Maintenance

This section assumes your database is set up using the best practice of one case per database.

## Standard Maintenance Suggestion (Full Database Recovery)

The default recovery mode and default for SQL Server upon install is Full Database Recovery Mode. This mode records every transaction performed to a database. When combined with a proper backup maintenance plan, Full Recovery Mode provides the ability to restore to any point from the time of the Full Backup record to the most recent activity in the database. This allows for an increased ability to recover on the fly, but compels the use of an effective backup maintenance plan, as the transaction logs would otherwise continue to grow indefinitely. This maintenance package is a good option for those that do not know how heavy a load their SQL Server is experiencing under daily use. An assumption made is that the backups are stored in a recoverable location in the event of a disaster.

**Note:** Maximum Data Loss in the event of a disaster is four hours.

### Recommended Tasks for Regular Backup:

Action	Time
Full Backup <ul style="list-style-type: none"><li>Full backup will allow you to restore database</li></ul>	Weekly
Incremental Backup	Daily
Transaction Log Backup <ul style="list-style-type: none"><li>Performing this action along with Full Backup will keep your transaction log size from growing</li></ul>	Every Four Hours
Reorganization of Indexes	Daily
Rebuild Indexes	Weekly
Update Statistics	Daily
Maintenance Cleanup Tasks <ul style="list-style-type: none"><li>Use the Maintenance Cleanup Tasks to remove old files related to maintenance plans above. This includes text reports created by maintenance plans and database backup files.</li></ul>	Weekly

## Frequency of Review and Adjustments

The frequency of the tasks should be reviewed and adjusted after implementation. Use the following information to determine what is best for your needs.

### 24 Hours

Answer the following questions:

1. Have the transaction logs grown too large for your storage?
2. Have the transaction logs grown to larger than 30%-50% of the database size?
3. Has the user performance increased or decreased?
4. Have the jobs overlapped when running?

### 1 Week

Answer the following questions:

1. Did all the backups complete?
2. Did your Cleanup Task remove excess tLog backups and incremental backups?
3. Did the rebuild indexes complete?
4. How long did each job take over the week?

This is a good point to document while you go through this process.

### Monthly

Answer all the questions above.

- ❖ Review.

## Advanced Maintenance Suggestion (Full Database Recovery)

All of the above applies to this suggestion and assumptions are the same as above. These recommendations also include some advanced scripts found at <https://ola.hallengren.com>.

**Note:** Maximum Data Loss in the event of a disaster is four hours.

### Recommended Tasks for Advanced Backup:

Action	Time
Full Backup <ul style="list-style-type: none"><li>Full backup will allow you to restore database</li></ul>	Weekly
Incremental Backup	Daily
Transaction Log Backup <ul style="list-style-type: none"><li>Performing this action along with Full Backup will keep your transaction log size from growing</li></ul>	Every 15 Minutes
Reorganization of Indexes <ul style="list-style-type: none"><li>General rule of thumb is to reorganize if fragmentation is below 30%, rebuild if it is above 30%. This can be done easily with scripts from <a href="https://ola.hallengren.com">https://ola.hallengren.com</a>.</li></ul>	Four Hours
Rebuild Indexes <ul style="list-style-type: none"><li>This task is most needed after large changes to the database in a short timeframe; for example, after adding data using the AD Lab evidence processor. You can trigger a rebuild inside SQL Server to auto rebuild the indexes after a large job completes.</li></ul>	Four Hours
Update Statistics	Four Hours
Maintenance Cleanup Tasks <ul style="list-style-type: none"><li>Use the Maintenance Cleanup Tasks to remove old files related to maintenance plans above. This includes text reports created by maintenance plans and database backup files.</li></ul>	Weekly

## Standard Maintenance Suggestion (Simple Database Recovery)

**Note:** Maximum Data Loss in the event of a disaster is twenty-four hours.

### Recommended Tasks for Standard Backup:

Action	Time
Full Backup	Weekly
<ul style="list-style-type: none"><li>• Full backup will allow you to restore database</li></ul>	
Incremental Backup	Daily
Reorganization of Indexes	Daily
Rebuild Indexes	Weekly
Update Statistics	Daily
Maintenance Cleanup Tasks	Weekly
<ul style="list-style-type: none"><li>• Use the Maintenance Cleanup Tasks to remove old files related to maintenance plans, including text reports created by maintenance plans and database backup files.</li></ul>	

## Advanced Maintenance Suggestion (Simple Database Recovery)

All of the above applies to this suggestion and assumptions are the same as above. These recommendations also include some advanced scripts found at <https://ola.hallengren.com>.

**Note:** Maximum Data Loss in the event of a disaster is twenty-four hours.

### Recommended Tasks for Advanced Backup:

Action	Time
Full Backup <ul style="list-style-type: none"><li>Full backup will allow you to restore database</li></ul>	Weekly
Incremental Backup	Daily
Reorganization of Indexes <ul style="list-style-type: none"><li>General rule of thumb is to reorganize if fragmentation is below 30%, rebuild if it is above 30%. This can be done easily with scripts from <a href="https://ola.hallengren.com">https://ola.hallengren.com</a>.</li></ul>	Four Hours
Rebuild Indexes <ul style="list-style-type: none"><li>This task is most needed after large changes to the database in a short timeframe; for example, after adding data using the AD Lab evidence processor. You can trigger a rebuild inside SQL Server to auto rebuild the indexes after a large job completes.</li></ul>	Four Hours
Update Statistics	Four Hours
Maintenance Cleanup Tasks <ul style="list-style-type: none"><li>Use the Maintenance Cleanup Tasks to remove old files related to maintenance plans above. This includes text reports created by maintenance plans and database backup files.</li></ul>	Weekly

## DBConfig

Below are the Validate commands and their descriptions.

Validate, in most circumstances, will find and correct problems found in the database or sequences. Some of these are the result of old issues or database changes that have occurred between versions. Along with other maintenance, running all of the Validate options should be done at least once a month.

### DBConfig Validate Commands

Command	Description
ClearEmptyDatabases	Remove ADG databases that have no cases or shared schema present.
FixSequences	Check Sequence values and reset them to Max(Column) + 1 if needed. Also on MSSQL, move Sequence files to proper location if needed.
RebuildMissingIndexes	Rebuild any missing index, primary keys, or unique key constraints.
FailAllHungJobs	Set any jobs that are not complete to failed.
ValidatePermissionsAndVersions	Check for users/groups/roles that are not current (Ex: ADG55 when it should be ADG56, etc.) Also, see if all case versions are the same.
RemoveAbandonedObjectData	Remove any "object" data that does not have a primary value in the cmn_objects table. (This can happen when EP fails and does not properly clean up.)
ApplyAddendums	Apply any data addendums (data patches). Currently there is only one; a remedy for out of sync Head Of Family flags.
FixFTKFilterMRUDefaultDates	Changes what looks like an empty or negative date to be a valid date in the table ftk_FilterRecentAttrs table.  <b>[TFS 18494] There seems to be an error in either restore or migration that puts bad data in the column. The fix simply validates the date (it sets it to 01-01-0001). This has only been observed on Oracle databases.</b>

## DBControl

The DBControl-Validate Fix command can be run as often as needed. This Validate Fix command does not do the same validation mentioned above. DBControl has a subset of the features DBConfig has. This can be run more often than the DBConfig, as the DBControl has the ability to validate a single case instead of the entire attached database, as well as any shared databases.