

# SQLskills Immersion Event IE0: Accidental/Junior DBA

## Module 2: Installation and Configuration

Erin Stellato  
Erin@SQLskills.com



### Introduction

- **Proper installation and configuration of SQL Server is very important**
  - It affects security, performance, scalability, and maintainability
- **Some settings and components can be changed after initial installation, some cannot**
  - Spend the time up front to get it right the first time
- **Configuration occurs for both the OS and SQL Server**
- **Maintenance needs to occur for both the OS and SQL Server**
  - Service Packs, individual hotfixes, Cumulative Updates (SQL Server only)

## Security Considerations

- **A well thought-out installation includes security considerations, as the right decisions can prevent many problems**
  - Physical security
  - Service accounts
    - More secure, reliable environment
  - Database file locations
  - Installing only what's needed
    - Decreases resource usage
    - Reduces surface attack area
    - Reduces patching time

## Configuration Considerations

- **Many default configuration settings are incorrect**
  - OS
  - SQL Server instance
  - SQL Server database
- **Using “best-practice” values has many benefits**
  - Improved performance and scalability
  - Improved reliability
  - Decrease restore and recovery time (therefore increase availability)

## Maintenance Considerations

- **Maintenance is preventative, and provides many benefits**
  - Reduces the possibility of known problems
  - Provides an opportunity to walk through testing and planning
  - Provides an opportunity to use your HA infrastructure
- **Better support with third-party vendors, including those for:**
  - Hardware
  - Storage
  - Microsoft (OS and SQL Server)
  - Application

## SQL Server Version and Edition

- **Determining the version and Edition of SQL Server you will run is one of the first decisions that must be made**
  - Current release is 2019, you may have an earlier version available in-house
    - Note supported OS versions as part of your version decision
  - Different versions of SQL Server have different Editions available
- **Most important differences to understand are:**
  - Features
  - Limitations
  - Cost (licensing)

## Overview

- Installation best practices
- Configuration best practices
- Setting up database mail and alerts
- Database settings
- Database file layouts

## Before You Actually Install SQL Server...

- Verify hardware and storage for the solution
- Install and patch operating system
- Update BIOS and firmware
- Install/update device drivers
- Verify power management settings
- Provision Domain Accounts
  - Grant additional rights
- Perform final server configuration tasks
- Decide what SQL Server features to install

## Install and Patch Windows

- **Install Windows**
  - Verify the version and edition needed to support the SQL Server version and edition selected, as well as solution requirements (e.g. clustering)
  - Apply Windows service packs and appropriate patches/hotfixes
- **Install and configure Microsoft Update**
  - Ensure that you are notified when new updates are available
  - Do not allow Windows to automatically download and install updates
- **This may all be done by another team/team member, make sure you verify latest versions and updates**

## Update BIOS, Firmware, and Device Drivers

- **Verify the latest main system BIOS is installed**
  - You will need to check the vendor's site for the latest version
  - New servers often have an old BIOS version
- **Vendor tools can detect out-of-date firmware**
  - Dell Open Management Systems Administrator
  - HP System Insight Manager
  - IBM Director
- **Ensure that the latest firmware and drivers for all components are installed**
  - e.g. RAID controllers, HBAs, NICs, PCI-E storage cards
- **Choose the vendor supplied, system-specific drivers instead of generic Windows drivers when possible**

## Verify Power Management Settings

- **Windows**
  - Set the power plan to High Performance
    - Default is Balanced
  - This can have a negative effect on database server performance
  - Verify using free CPU-Z tool ([www.cpuid.com](http://www.cpuid.com))
  - May need to update hardware power management in the BIOS as well
- **BIOS**
  - Power management in the BIOS setup application should be set to OS control, or disabled
- **Virtual Machines**
  - Set at host level, not VM level
- **Cloud (e.g. Azure, AWS)**
  - No ability to control

## SQL Server Service Accounts

- **Determine accounts for services**
  - Database Engine, SQL Server Agent, SSIS, SSRS, SSAS, etc.
- **Recommended to use Managed Service Accounts (domain account), Group Managed Service Accounts (MSA for multiple servers), or Virtual Accounts (local account)**
- **Specific domain accounts can be created and used but have to be managed manually**
  - You can use one account for all services, or one account *per* service
  - Assign minimal permissions
    - SQL Server installation will grant necessary rights for each account
    - Accounts do not need to be Local Administrators on the server
    - Specific permissions are documented here: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-windows-service-accounts-and-permissions>
  - You will need the name and password for each account for the actual installation

## Rights for the Database Engine Service Account

- **Need to grant rights to the engine service account**
  - Enabled via Local Security Policy
  - Perform volume maintenance tasks (Instant File Initialization)
    - This can be done during installation in SQL Server 2016 and higher
  - Lock pages in memory

## Instant File Initialization

- Allows SQL Server 2005 and higher to use Windows technology to skip the zeroing process for data files (not log files)
- Requires “Perform Volume Maintenance Tasks” security permission granted to SQL Server service account or group
- Cannot be enabled from within SQL Server, except during 2016 installation
- **Use Local Security Policy Editor to grant the permission**
  - Administrative Tools -> Local Security Policy and then Local Policies -> User Rights Assignment
  - Defaults to Local Administrators Group but can grant this to a lower privileged account
- Instant initialization will happen automatically once SQL Server is restarted after granting the permission
- Message in SQL Server ERRORLOG in SQL Server 2014 SP2 and higher:  
Server Database Instant File Initialization: enabled

## Instant File Initialization in Action

- **Hardware configuration – Dell PowerEdge R720**  
32 cores, 64GB memory, RAID 10 array with 6 x 300 GB, 15k disks
- **Software configuration: SQL 2014 (similar results on all versions)**
  - With zero initialization
    - CREATE DATABASE with 20GB Data file = 7:12 minutes
    - ALTER DATABASE BY 10GB = 4:20 minutes
    - RESTORE 30GB DATABASE (EMPTY Backup) = 29:13 minutes
    - RESTORE 30GB DATABASE (11GB Backup) = 30:27 minutes
  - With instant file initialization
    - CREATE DATABASE with 20GB Data file = 8 seconds
    - ALTER DATABASE BY 10GB = < 1 second
    - RESTORE 30GB DATABASE (EMPTY Backup) = 14 seconds
    - RESTORE 30GB DATABASE (11GB Backup) = 1:32 minutes
- **Additional data:** <http://bit.ly/1Rzq6Yl>

## Lock Pages in Memory

- **Required for AWE on 32-bit SQL Servers**
- **Prevents paging of the buffer pool on 64-bit instances**
  - Does not require configuration of "AWE Enabled" in SQL Server
- **Cannot be enabled from within SQL Server**
- **Requires "Lock pages in memory" security permission granted to SQL Server service account or group**
  - SQL 2005+
  - Standard Edition also requires TF 845 prior to SQL 2012
- **Use Local Security Policy Editor to grant the permission**
  - Administrative Tools -> Local Security Policy and then Local Policies -> User Rights Assignment
- **Lock Pages in Memory used automatically once SQL Server is restarted after granting the permission**
  - Message in the SQL Server ERRORLOG: Using locked pages for buffer pool



## Final Server Configuration Tasks

- Change network settings to use a static IP address, if appropriate
- Add server to the Windows domain, if appropriate
- Enable Remote Desktop in Windows
  - Allows you to logon remotely when needed
- Antivirus software should be installed and configured with appropriate SQL Server exclusions
  - <http://support.microsoft.com/kb/309422>
- Make sure there are no pending reboots in Windows
- If using FILESTREAM, review the FILESTREAM Storage whitepaper:
  - <http://msdn.microsoft.com/en-us/library/hh461480.aspx>

## Deciding What Features to Install

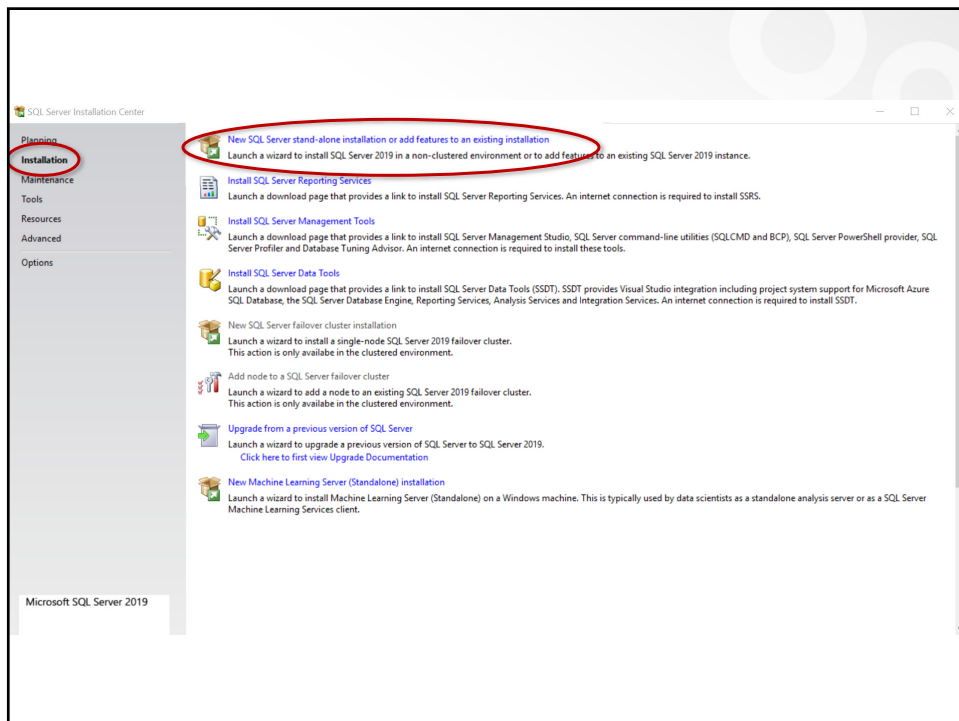
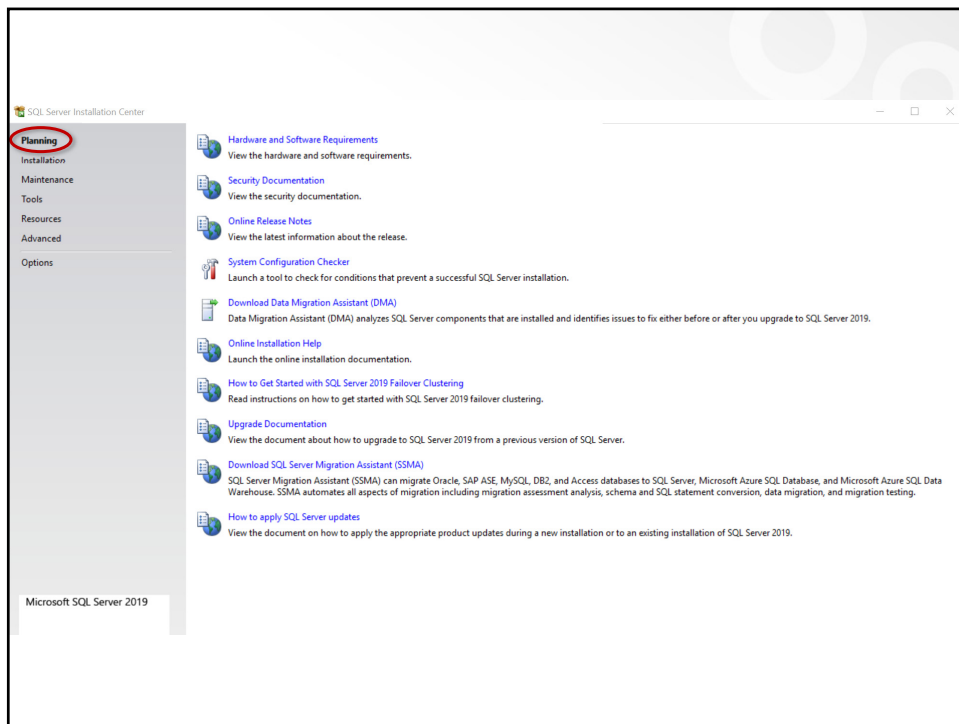
- Confirm what additional features will be installed (e.g. SSIS, SSRS)
  - This is dependent on what is needed by the applications the databases support
- Only install the features you need
  - This reduces your attack surface
  - It also reduces resource usage, and patch surface (think fewer reboots)
  - If you don't know what's needed, ask!
    - This may take several attempts
    - If needed, you can add features after the initial installation

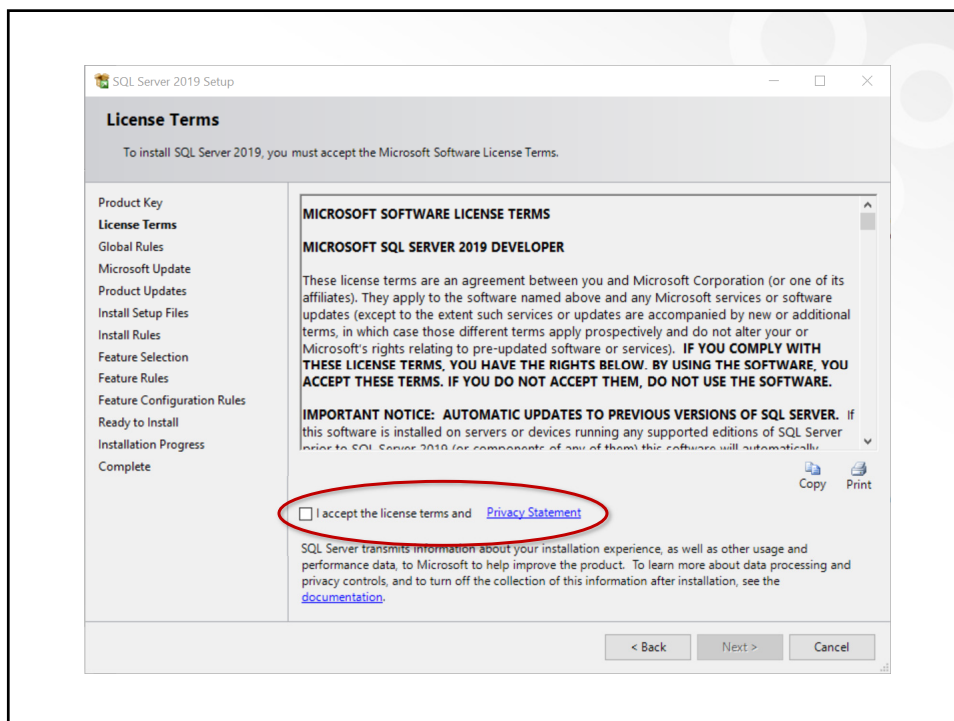
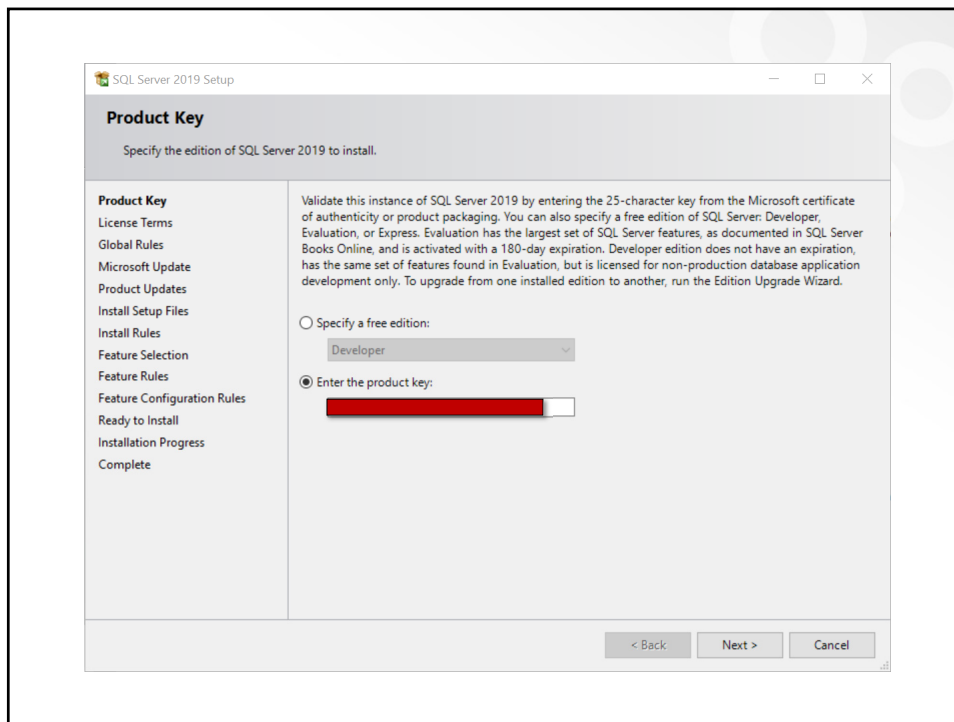
## Final Pre-Installation Tasks

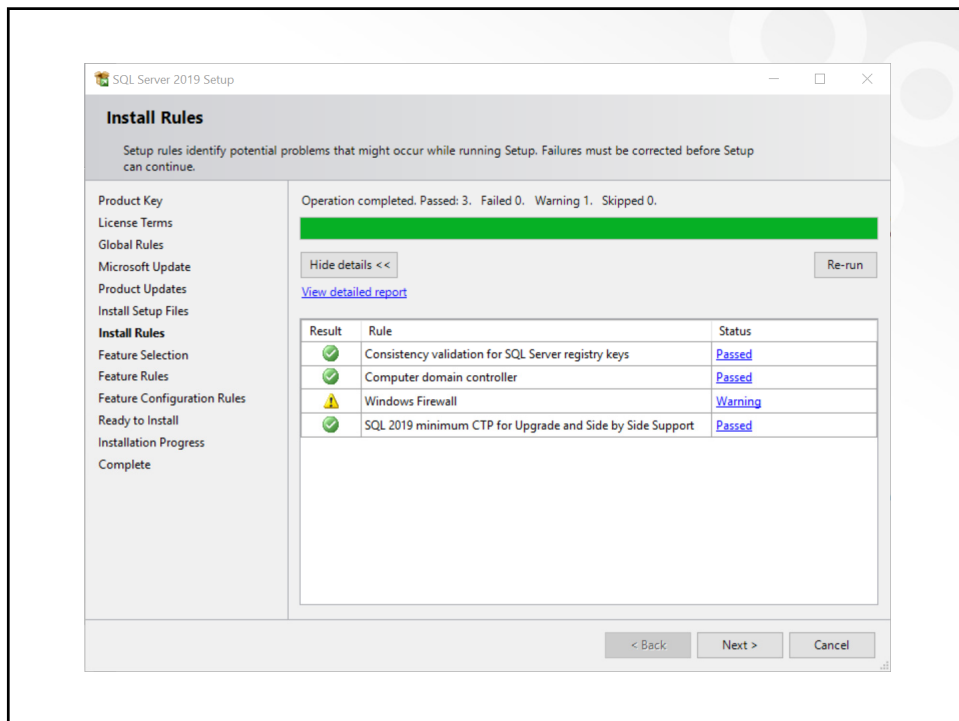
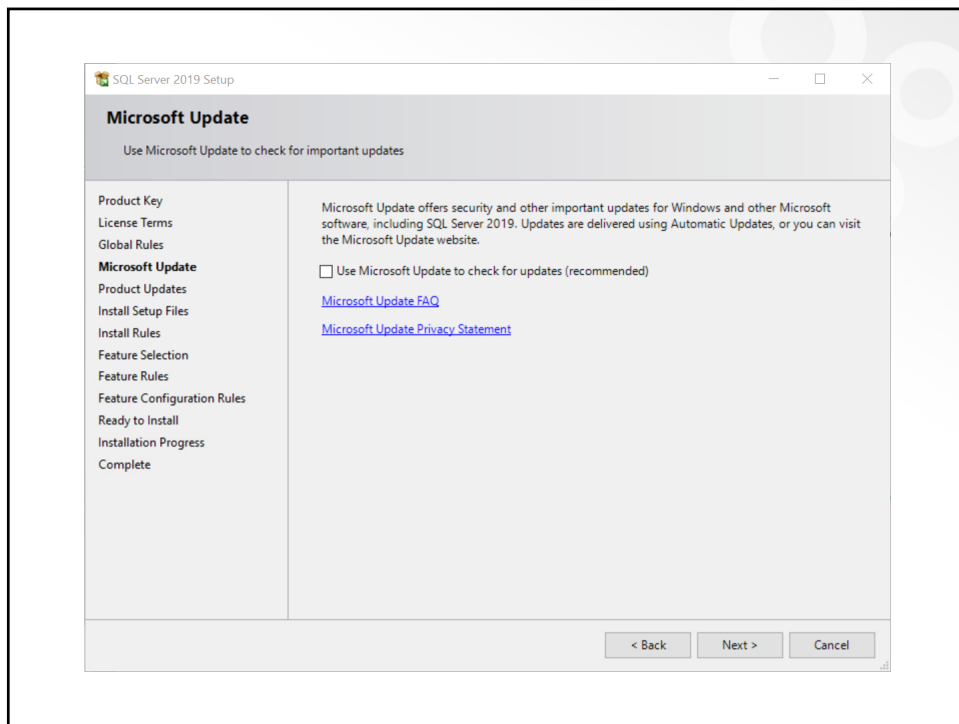
- **Establish disk and folder naming conventions**
  - Determine default data and backup directories to be used by the instance
- **System Configuration Checker**
  - Looks for conditions that would prevent a successful SQL Server installation
  - Creates a report that can be reviewed later if needed

## What You Decide *During* Installation

- **Stand-alone instance vs. FCI**
- **Features**
- **Instance type (DEFAULT or Named)\***
  - Only one instance can be the default
  - Any non-default instances must be named
- **Service Accounts**
- **Collation\***
  - This will be used by system databases and by all new databases by default
- **Authentication Mode**
  - Windows vs. Mixed
  - Windows is preferred due to integrated security, but many third-party applications will use SQL logins and require mixed mode
- **Data Directories**
- **FILESTREAM**







SQL Server 2019 Setup

## Feature Selection

Select the Developer features to install.

Product Key  
License Terms  
Global Rules  
Microsoft Update  
Product Updates  
Install Setup Files  
Install Rules  
**Feature Selection**  
Feature Rules  
Feature Configuration Rules  
Ready to Install  
Installation Progress  
Complete

Looking for Reporting Services? [Download it from the web](#)

Features:

Instance Features

- ☐ Database Engine Services
- ☐ SQL Server Replication
- ☐ Machine Learning Services and Language Extensions
  - ☐ R
  - ☐ Python
  - ☐ Java
- ☐ Full-Text and Semantic Extractions for Search
- ☐ Data Quality Services
- ☐ PolyBase Query Service for External Data
- ☐ Java connector for HDFS data sources
- ☐ Analysis Services

Shared Features

- ☐ Machine Learning Server (Standalone)
  - ☐ R
  - ☐ Python
- ☐ Data Quality Client
- ☐ Client Tools Connectivity
- ☐ Integration Services
- ☐ Scale Out Master
- ☐ Scale Out Worker
- ☐ Client Tools Backwards Compatibility
- ☐ Client Tools SDK
- ☐ Distributed Replay Controller
- ☐ Distributed Replay Client
- ☐ SQL Client Connectivity SDK
- ☐ Master Data Services

Redistributable Features

Select All Unselect All

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

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

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

Feature description:

Includes the Database Engine, the core service for storing, processing and securing data. The Database Engine provides controlled access and rapid transaction processing and also provides rich support for sustaining high availability. The Database Engine also provides support for the utility control point in the SQL Server Utility. Only Database Engine Services and Analysis Services can be clustered.

Prerequisites for selected features:

Disk Space Requirements

Drive C: 1003 MB required, 507629 MB available

< Back Next > Cancel

SQL Server 2019 Setup

## Instance Configuration

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

Product Key  
License Terms  
Global Rules  
Microsoft Update  
Product Updates  
Install Setup Files  
Install Rules  
Feature Selection  
Feature Rules  
**Instance Configuration**  
Server Configuration  
Database Engine Configuration  
Feature Configuration Rules  
Ready to Install  
Installation Progress  
Complete

☐ Default instance

☒ Named instance: SQL2019

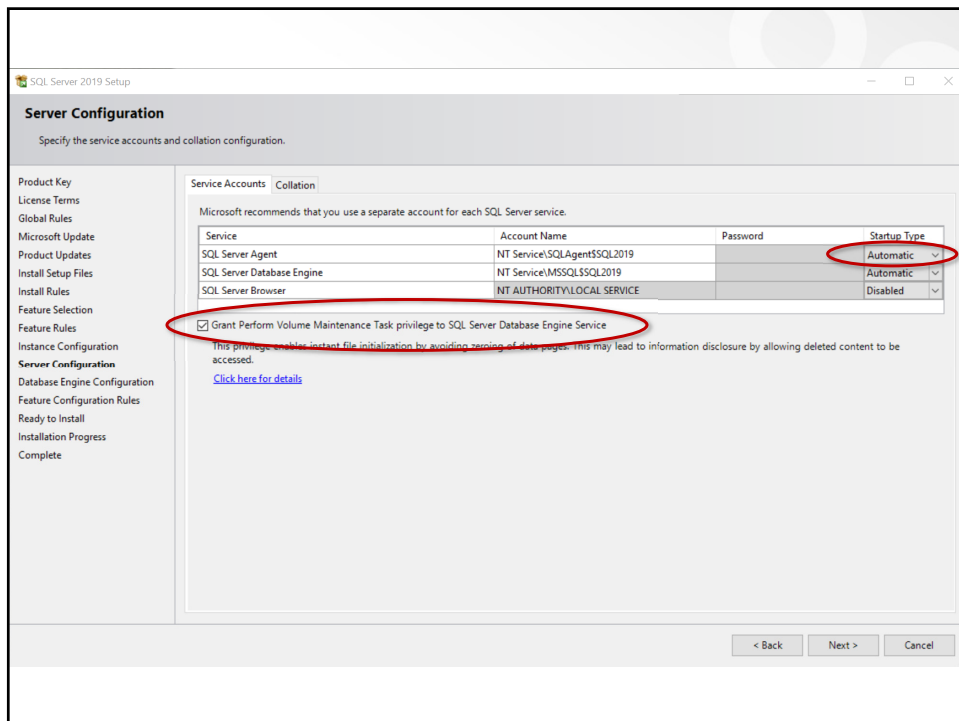
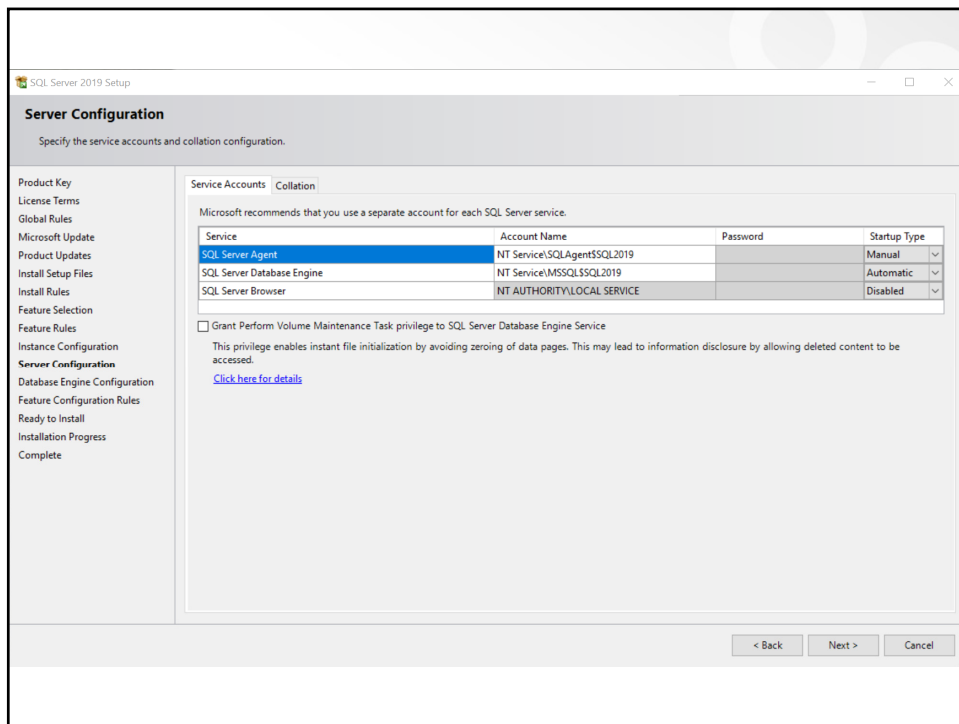
Instance ID: SQL2019

SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL15.SQL2019

Installed instances:

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

< Back Next > Cancel



SQL Server 2019 Setup

## Server Configuration

Specify the service accounts and collation configuration.

Product Key  
License Terms  
Global Rules  
Microsoft Update  
Product Updates  
Install Setup Files  
Install Rules  
Feature Selection  
Feature Rules  
Instance Configuration  
**Server Configuration**  
Database Engine Configuration  
Feature Configuration Rules  
Ready to Install  
Installation Progress  
Complete

Service Accounts | Collation

Database Engine:

SQL\_Latin1\_General\_CP1\_CI\_AS

Customize...

Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data

< Back
Next >
Cancel

SQL Server 2019 Setup

## Database Engine Configuration

Specify Database Engine authentication security mode, administrators, data directories, TempDB, Max degree of parallelism, Memory limits, and Filestream settings.

Product Key  
License Terms  
Global Rules  
Microsoft Update  
Product Updates  
Install Setup Files  
Install Rules  
Feature Selection  
Feature Rules  
Instance Configuration  
Server Configuration  
**Database Engine Configuration**  
Feature Configuration Rules  
Ready to Install  
Installation Progress  
Complete

Server Configuration | Data Directories | TempDB | MaxDOP | Memory | FILESTREAM

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:

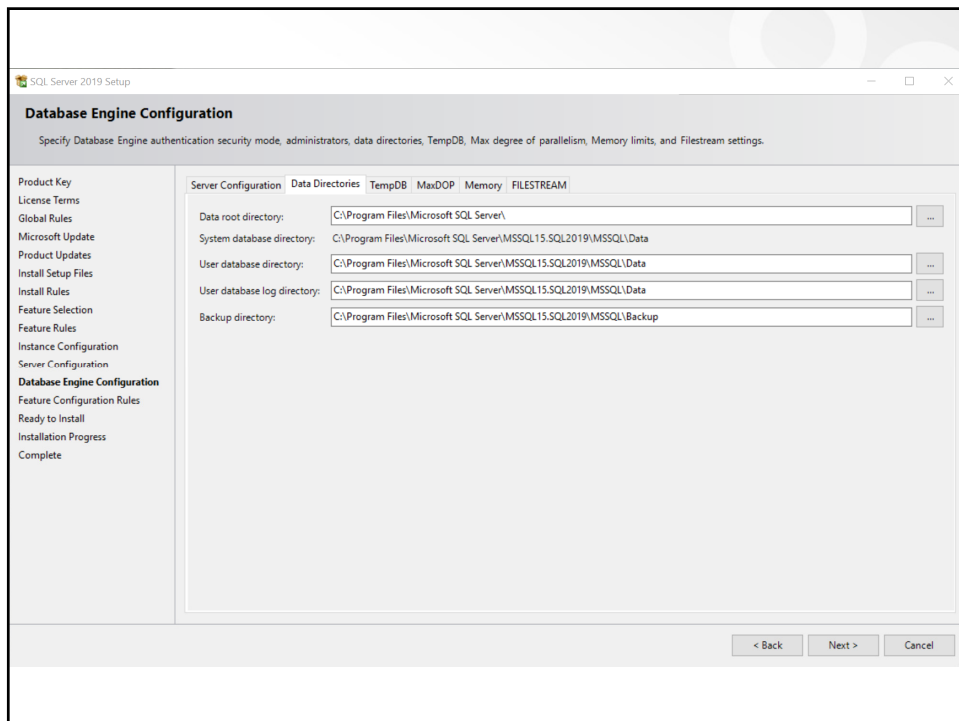
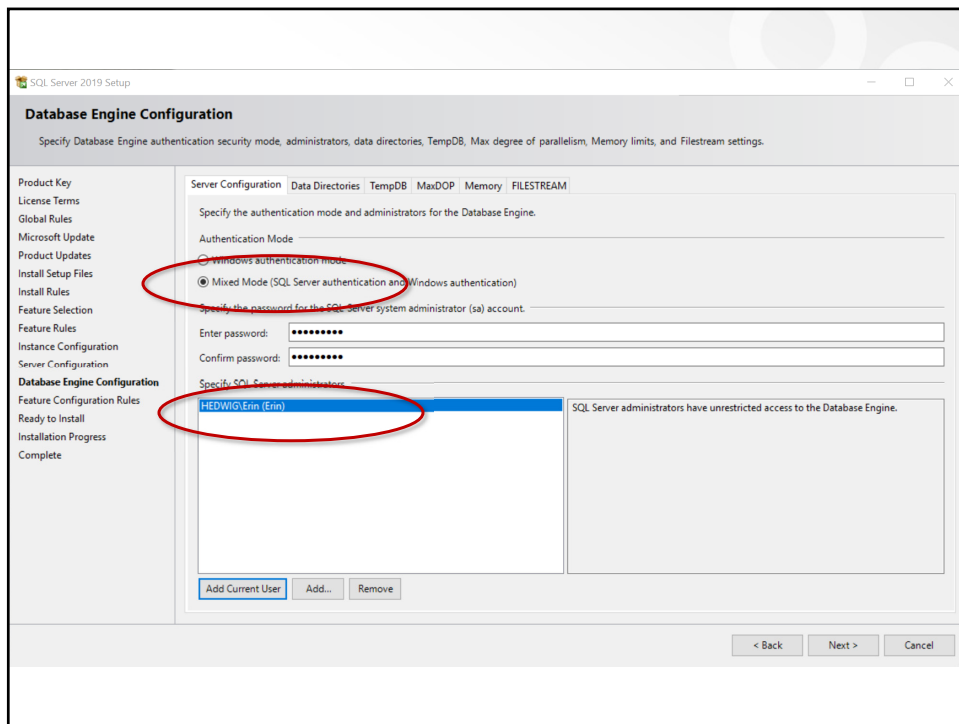
Specify SQL Server administrators

SQL Server administrators have unrestricted access to the Database Engine.

Add Current User
Add...
Remove

< Back
Next >
Cancel





SQL Server 2019 Setup

## Database Engine Configuration

Specify Database Engine authentication security mode, administrators, data directories, TempDB, Max degree of parallelism, Memory limits, and Filestream settings.

Product Key  
License Terms  
Global Rules  
Microsoft Update  
Product Updates  
Install Setup Files  
Install Rules  
Feature Selection  
Feature Rules  
Instance Configuration  
Server Configuration  
**Database Engine Configuration**  
Feature Configuration Rules  
Ready to Install  
Installation Progress  
Complete

Server Configuration | Data Directories | TempDB | MaxDOP | Memory | FILESTREAM

Data root directory: C:\Program Files\Microsoft SQL Server\  
System database directory: C:\Program Files\Microsoft SQL Server\MSSQL15.SQL2019\MSSQL\Data  
User database directory: H:\Databases\2019  
User database log directory: H:\Databases\2019  
Backup directory: H:\Backups

< Back Next > Cancel

SQL Server 2019 Setup

## Database Engine Configuration

Specify Database Engine authentication security mode, administrators, data directories, TempDB, Max degree of parallelism, Memory limits, and Filestream settings.

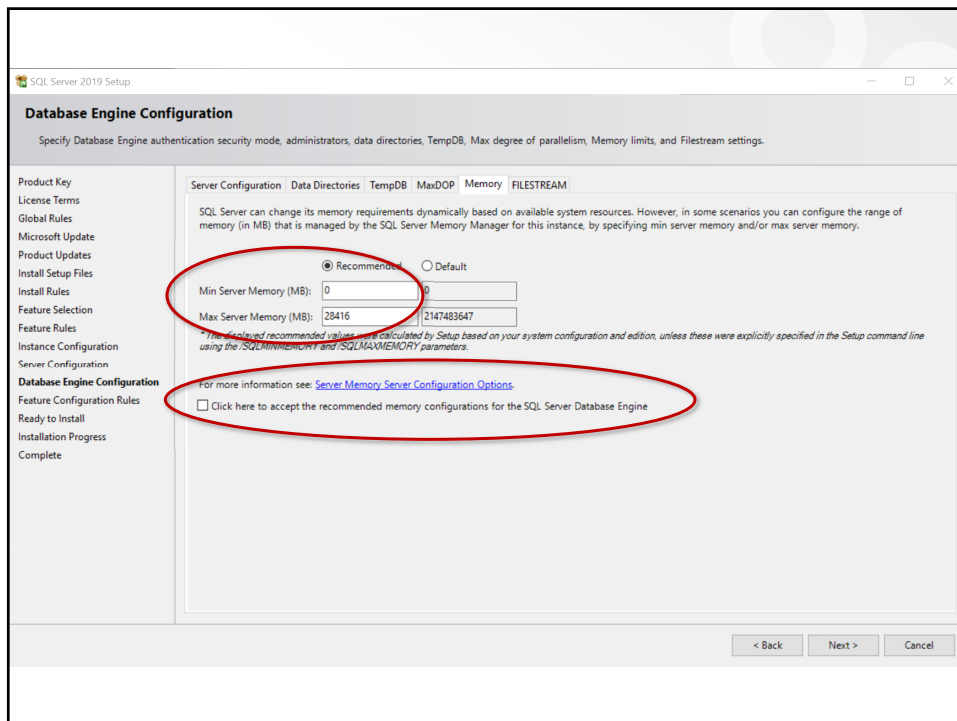
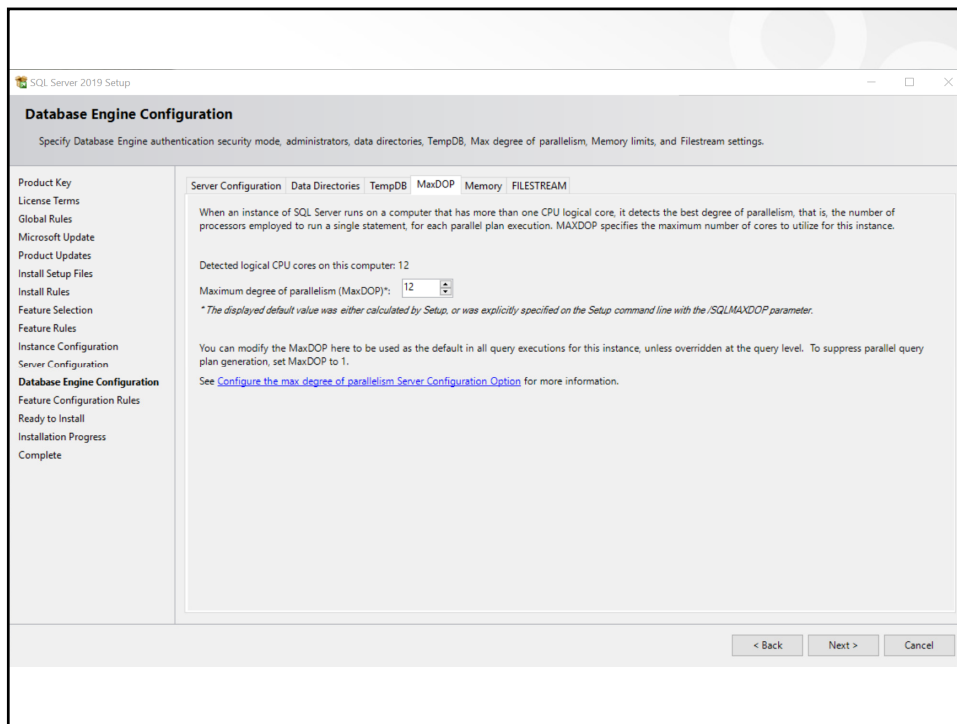
Product Key  
License Terms  
Global Rules  
Microsoft Update  
Product Updates  
Install Setup Files  
Install Rules  
Feature Selection  
Feature Rules  
Instance Configuration  
Server Configuration  
**Database Engine Configuration**  
Feature Configuration Rules  
Ready to Install  
Installation Progress  
Complete

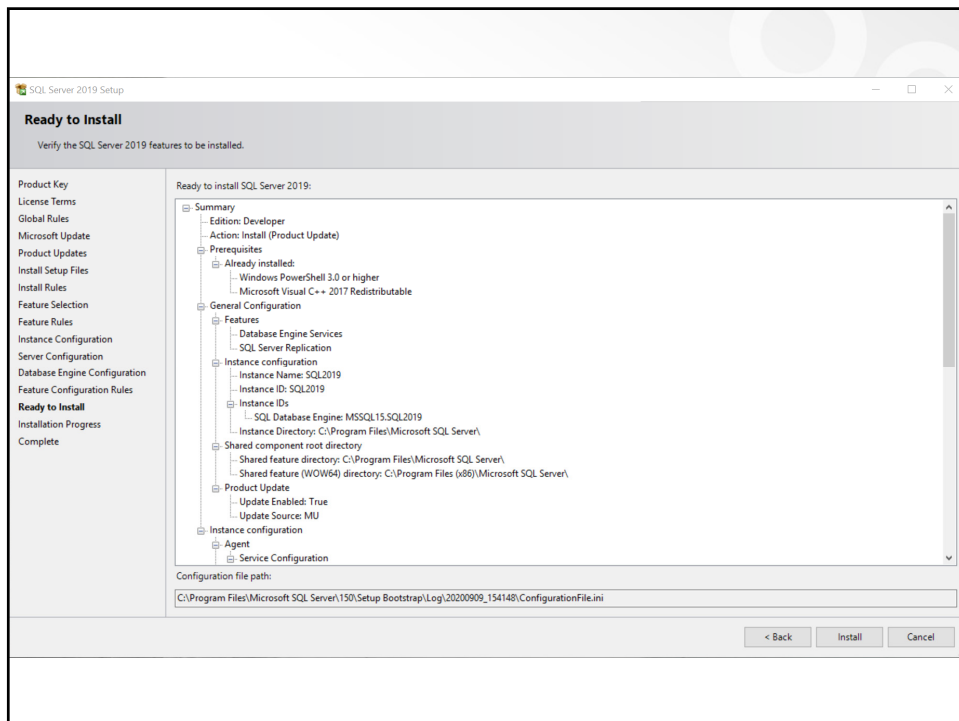
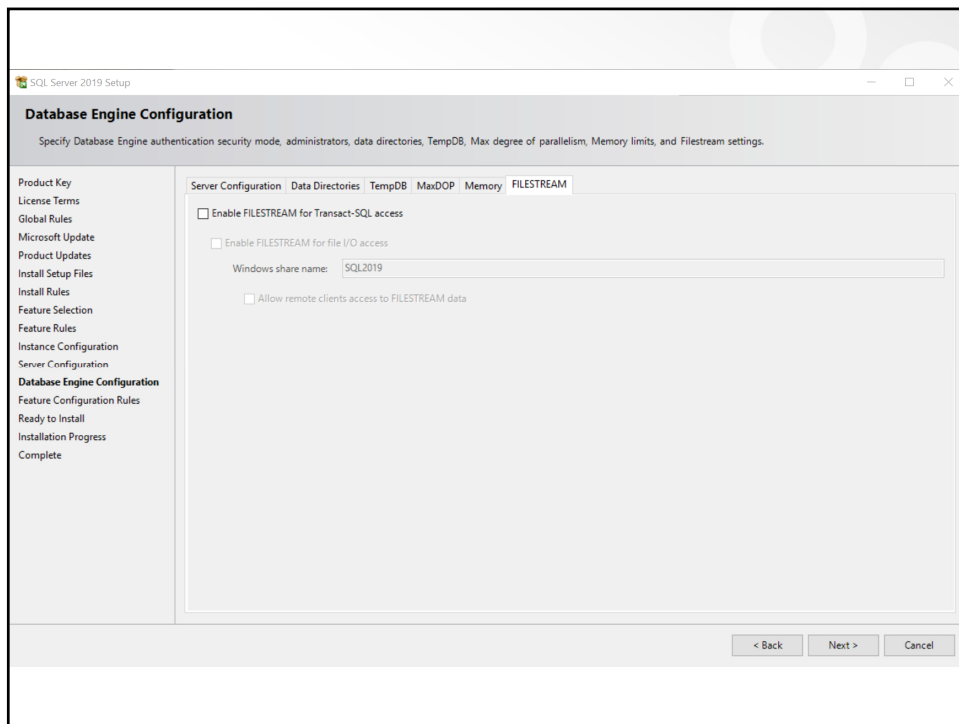
Server Configuration | Data Directories | TempDB | MaxDOP | Memory | FILESTREAM

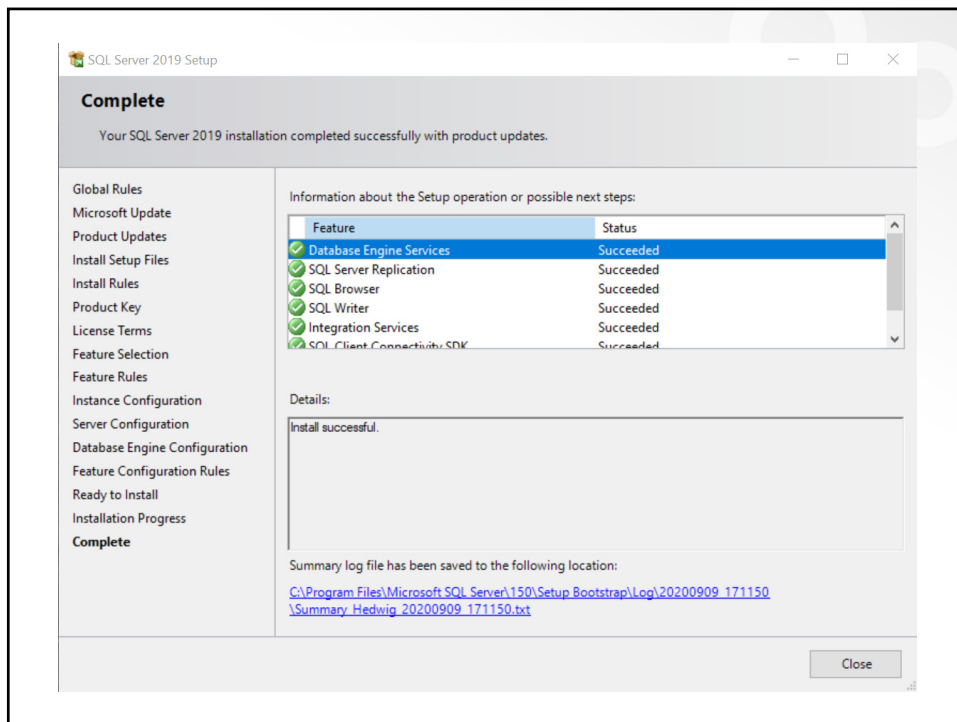
TempDB data files: tempdb.mdf, tempdb\_mssql\_#.ndf  
Number of files: 8  
Initial size (MB): 8 Total initial size (MB): 64  
Autogrowth (MB): 64 Total autogrowth (MB): 512  
Data directories: C:\Program Files\Microsoft SQL Server\MSSQL15.SQL2019\MSSQL\Data  
Add...  
Remove

TempDB log file: templog.ldf  
Initial size (MB): 8 Setup could take longer with large initial size.  
Autogrowth (MB): 64  
Log directory: C:\Program Files\Microsoft SQL Server\MSSQL15.SQL2019\MSSQL\Data

< Back Next > Cancel







## Additional Installation Decisions

- **SQL Server 2016+**
  - Whether to assign the Perform Volume Maintenance right to the service account (enable IFI)
  - tempdb configuration
- **SQL Server 2019**
  - MAXDOP
  - Max Server Memory

## Overview

- Installation best practices
- Configuration best practices
- Setting up database mail and alerts
- Database settings
- Database file layouts

## Post-Installation Configuration (1)

- **Security**
  - Enable Remote DAC
  - Remove sysadmin from BUILTIN\Administrators (before SQL Server 2008)
- **Instance settings**
  - Max server memory (MB)
  - Max degree of parallelism\*
  - Cost threshold for parallelism
  - Optimize for ad hoc workloads
  - Compress backup (SQL Server 2008 and higher)
  - Perform backup checksum (SQL Server 2014 and higher)
  - Trace flags

## Post-Installation Configuration (2)

- **Configure network-related settings**
  - Configure ports
  - Set up firewall rules
  - Test connectivity
- **Configure service settings**
  - SQLAgent (automatic)
    - Exception is Failover Cluster Instance, the SQLAgent will be set to Manual and should be left as such
- **Configure the ERRORLOG**
- **Configure tempdb**

## Enable Remote DAC

- DAC = Dedicated Administrator Connection
- Back door into SQL Server when it's locked up and not responding
- By default it is only available from the server
- Enable Remote DAC to access the instance from any machine

```
sp_configure 'remote admin connections', 1;  
GO  
RECONFIGURE;  
GO
```

- Listens on port 1434 by default
- Only one connection using DAC at a time
- Connect via SSMS by prefixing instance name with Admin:
- Things to be aware of for named instances:  
<https://www.sqlskills.com/ie0/testingdac>

## Remove sysadmin from BUILTIN\Administrators

- Prior to SQL Server 2008, the sysadmin privilege was assigned to BUILTIN\Administrators and any Windows administrator had full sysadmin rights to the SQL Server instance
- Even with this privilege removed (default for SQL Server 2008 and higher installations), administrators can still gain access to SQL Server by starting SQL Server in single-user mode
- Further, using PsExec (SysInternals tool), administrators can connect under NT AUTHORITY\SYSTEM, which has sysadmin privileges
  - Only through SQL Server 2008 R2
- In SQL Server 2012 and higher, the NT AUTHORITY\SYSTEM group no longer has sysadmin privileges
- Do not delete the NT AUTHORITY\SYSTEM group, or remove sysadmin from it, as specific functions require it to exist

## Max Server Memory (MB) (1)

- Default value is 2147483647
- Required on 64-bit systems to prevent memory pressure and out of memory conditions
  - If you do not set a max, SQL Server will consume as much memory as it can
- This value only applies to the buffer pool, it does not set the total amount of memory used by SQL Server
  - Additional memory for thread stack and multipage allocators
  - Memory for thread stack = (max worker threads) x (stack size)
  - x32 = 512KB; x64 = 2MB; ia64 = 4MB



## Max Server Memory (MB) (2)

- (Total system memory) minus (memory for thread stack) minus (OS memory requirements ~ 2-4GB) minus (memory for other applications)
- How we typically do it:
  - 1GB for the OS
  - + 1GB for each 4GB between 4-16GB
  - + 1GB for each 8GB above 16GB in the server
- If in doubt, err on the lower side
- Then monitor Memory\Available Mbytes > 500MB and adjust as necessary
- Post: How much memory does my SQL Server actually need?
  - <https://www.sqlskills.com/ie0/instancememory>

## Max degree of parallelism

- Defaults to 0
  - Any parallel query can use up to  $n$  processors for execution, where  $n$  is the number of processors available to SQL Server
- MAXDOP limits:
  - Maximum concurrently executing tasks on schedulers
  - Maximum operators able to execute concurrently in the same plan
- It is recommended to specify a value other than 0
  - Some applications may recommend a value of 1 (e.g. SharePoint)
  - For non-NUMA systems, set MAXDOP no higher than the number of physical cores, with a maximum value of 8
    - If you have only 4 CPUs, can set to 0 or 2 (it depends on workload)
  - For NUMA systems, set MAXDOP equal to the number of physical cores in a single processor
- Can be set per-database as of SQL Server 2016
  - Overrides instance level setting if value is greater than 0

## Cost Threshold for Parallelism

- **Defaults to 5**
  - When the cost value for a query is above 5, SQL Server may create and execute a parallel plan for a query
- **This value is ignored if there is only 1 CPU available to SQL Server, or if MAXDOP = 1**
- **Typically recommended to increase this to a value of 25**
- **If desired, examine the plan cache to see if this value needs to be adjusted**
  - Post: Tuning Cost Threshold for Parallelism From the Plan Cache
    - <https://www.sqlskills.com/ie0/tuningcostthreshold>

## Optimize for Ad Hoc Workloads

- **Not enabled by default**
- **Available as of SQL Server 2008**
- **The first time a query is executed, a “compiled plan stub” is saved in the cache, which is significantly smaller in size than the full plan**
- **If the query is executed again, the full plan is saved in cache**
- **Enable this option to reduce the size of single-use plans in the plan cache**
  - Most applicable for ad hoc workloads
  - May not be supported with some third-party applications
  - Use query from Kimberly’s post to see if it would provide benefit to your system:
    - <https://www.sqlskills.com/ie0/adhocworkloadanalysis>

## Compress backup

- Not enabled by default
- Added for Enterprise Edition in SQL Server 2008
- Available for Standard Edition in SQL Server 2008 R2 onward
- Enable this to decrease the size of backups
  - Backup and restore duration will also decrease
- Compression does require more CPU, but this is typically not a bottleneck for most servers
- Confirm that the compression ratio for backups is worth the additional CPU cost (> 20% compression)
  - Compression is not as effective when TDE is used with SQL Server 2008 through 2014
  - Improvements in 2016, but MAXTRANSFERSIZE **must be** greater than 65536
    - <https://blogs.msdn.microsoft.com/sqlcat/2016/06/20/sqlsweet16-episode-1-backup-compression-for-tde-enabled-databases/>

## Backup checksum

- Not enabled by default
- Available in SQL Server 2014 and higher
- Enable this option to verify page checksums as pages are read for the backup
  - It will also create a checksum for the entire backup
- Use trace flag 3023 to enable for the instance prior to SQL Server 2014

## Trace Flags

- **Change the behavior of the SQL Server engine**
- **Can be version specific, and some provide different behavior depending on version (e.g. 4199)**
- **Can be applied for a specific session, or globally**
  - For global use, enter as a Startup Parameter in SQL Server service through SQL Server Configuration Manager
- **Always test behavior of a trace flag before implementing in a production environment**
- **Recommended trace flags:**
  - 3226 – prevents writing successful backup information to the ERRORLOG
  - 1118 – removes most single page allocations
    - Controlled through SET MIXED\_PAGE\_ALLOCATION database option in SQL Server 2016 and higher
- **More information:**
  - <https://www.sqlskills.com/blogs/erin/sqlskills-101-trace-flags/>



55

© SQLskills, All rights reserved.  
<https://www.sqlskills.com>

## Demo

Finding and changing instance settings

## Configure Ports

- **Configure via SQL Server Configuration Manager**
- **Default instance ports**
  - SQL Server will listen on TCP 1433
  - Dedicated Administrative Connection listens on TCP 1434
  - Service Broker defaults to TCP 4022
  - SQL Server Debugger/RPC uses TCP 135
  - SQL Browser Service uses UDP 1434 and TCP 2382
  - SQL Analysis Services uses TCP 2383
  - Reporting Services uses TCP 80 or TCP 443 for SSL
  - Database Mirroring and Availability Groups use TCP 5022 and 7022

## Configure Ports and Firewall Rules

- **Named instances should be assigned a fixed port manually in the SQL Server Configuration Manager (default is dynamic port assignment)**
  - Set a static port under IP All
  - Verify port number selected is not reserved
    - <http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>
- **Block external access to ports used by SQL Server and only allow access internally through firewall rules for specific ports**
  - Windows Firewall snap in or netsh command line
  - <http://support.microsoft.com/kb/968872>
- **Test connectivity from a *client* workstation after everything has been configured to validate firewall configuration**

## Configure Service Settings

- SQL Agent defaults to manual start within services
- Change the SQL Agent Start Mode to Automatic via SQL Server Configuration Manager
- Changes to the SQL Server services, and stops/starts/restarts, should *always* be done via Configuration Manager; do not use Windows Services
  - e.g. service account, start mode
  - Exception: When SQL Server is clustered, use Cluster Manager for restarts, etc.
- Specify Startup Parameters via Configuration Manager as well
  - e.g. -T3226

## Configuring the Error Log

- By default, error log rolls over when instance restarts, and only six (6) files are retained at a time
- This can cause large files due to infrequent restarts, or a loss of historical information due to frequent restarts
- To maintain a history, set up a job to cycle the error log on a regular basis, and increase the number of files available
- More information:
  - <https://www.sqlskills.com/blogs/erin/sqlskills-101-the-sql-server-errorlog/>

## tempdb

- **tempdb is one of four system databases, and is a shared resource for the instance**
- **Traditional tempdb uses include:**
  - User objects such as table variables (@), #temp tables/indexes, global ##temp tables/indexes, regular tables
  - Internal objects such as worktables, workfiles from spills to disk, intermediate sort results, version store
- **Use of tempdb can vary greatly between applications (those created in-house and third-party)**
- **It may be beneficial to separate tempdb from other databases due to I/O contention and put on high-performance I/O subsystem**
  - Entirely depends on I/O subsystem and workload involved
  - Often placed on SSDs, which are best suited to random I/O, but will generally give a performance boost to an I/O subsystem that's overwhelmed

## tempdb Contention

- **Some query workloads cause multiple concurrent threads to repeatedly create/drop small temp tables and/or worktables**
- **Causes PAGELATCH contention on allocation bitmaps and system catalogs**
  - Post: Optimizing tempdb configuration with SQL Server 2012 Extended Events
    - <https://www.sqlskills.com/ie0/optimizetempdbxe>
- **Contention can occur in *all* versions**
- **This can sometimes (rarely) happen in user databases with VERY high-end allocation workloads**

## Alleviating tempdb Contention<sup>(1)</sup>

- **Trace flag 1118 removes use of mixed extents (single-page allocations) in all databases**
  - Removes SGAM contention and should be enabled by default
  - <http://support.microsoft.com/kb/328551>
  - TF no longer needed in SQL Server 2016 and higher

## Alleviating tempdb Contention<sup>(2)</sup>

- **Create multiple data files to spread and reduce contention**
  - < 8 cores, use #files = #cores
  - > 8 cores, use 8 files, increasing by chunks of 4
    - Start with this recommendations and monitor for contention
    - Be careful of having too many data files
    - In some cases, increasing the size of the data files may also help
      - <https://support.microsoft.com/en-us/help/4099472>
    - Note: this does not apply to log files
- **In SQL Server 2019, the Memory-Optimized tempdb Metadata feature can provide relief when contention exists on tempdb system tables**
  - <https://bit.ly/3igqKZZ>
- **Recommendations to reduce allocation contention in SQL Server tempdb database**
  - <http://support.microsoft.com/kb/2154845>
  - <https://bit.ly/33gzvNh>



## tempdb Sizing

- Initial file size for tempdb after installation is 8 MB, and it reverts to the last *specifically* set size after a server restart
- Manually specify the size to avoid excessive auto-growth after restart
  - Not easy to figure out initial size as many operations can use tempdb space
- General practice is to create tempdb, run your production workload and then see how big tempdb gets
  - In addition to the general query workload, include index maintenance operations
  - Set tempdb size to resulting size
- Enable appropriate auto-growth, equal on all files
- Tempdb sizing can be done during installation as of SQL Server 2016
- Enable instant file initialization for instance
- Books Online: Capacity Planning for Tempdb
  - <http://msdn.microsoft.com/en-us/library/ms345368.aspx>

## Demo

Simple tempdb contention

## Overview

- Installation best practices
- Configuration best practices
- Setting up database mail and alerts
- Database settings
- Database file layouts

## Database Mail

- **Required for SQL Server to send mail**
  - It is used for notifications and alerts about errors and failed Agent jobs
- **Follow these steps to enable Database Mail**
  - Enable the Database Mail XPs
  - Use the Database Mail Configuration Wizard to set properties
  - Create a Database Mail account and profile
    - You must have access to an e-mail server that supports SMTP
    - You will need to know your outgoing mail server (SMTP) information
      - You may need assistance from a network administrator
    - Books Online: Configure Database Mail
      - <http://technet.microsoft.com/en-us/library/hh245116.aspx>
- **May require firewall rule to allow port 25 or 587**
- **May require disabling anti-virus rules that block SMTP access**

## Creating a New SQL Operator

- **Create at least one SQL operator**
  - Operators are aliases for people or groups that can receive electronic notification when jobs have completed or alerts have been raised
  - The SQL Server Agent service supports the notification of administrators through operators
  - Operators enable the notification and monitoring capabilities of SQL Server Agent

## Creating SQL Server Agent Alerts

- **Create SQL Server Agent alerts for critical errors**
  - Severity 19 through Severity 24 errors
  - Error 825 – Read-retry required
  - Glenn's posts on creating and configuring alerts
    - <https://www.sqlskills.com/ie0/configurealerts>
    - <https://www.sqlskills.com/ie0/configurealerts2>
  - Errors 854, 855, and 856 – Hardware memory corruption errors reported in Windows Server 2012 when running SQL Server 2012 and higher
    - <http://support.microsoft.com/kb/2967651>
- **You want to be notified about critical errors as soon as possible**
  - Make sure to configure a response for each alert
- **A SQL Server Agent alert can trigger an email, a net send, or a page**
  - An email to a distribution group is the preferred method
  - Net send and paging are deprecated in SQL Server 2012 and higher

## Demo

Setting up SQL Agent alerts

## Overview

- Installation best practices
- Configuration best practices
- Setting up database mail and alerts
- Database settings
- Database file layouts

## Default Database Settings

- When a new database is created, it uses the model system database as its “template”
- If any database option is changed in model, when you create a new database its option will be set to the same value
  - Be careful about making changes to model
- Don't assume that any database settings are changed, or not changed, by vendor database installations
- There are numerous options you can configure for a database
  - You will probably never change most options from the default value
- We will only cover the options that should be changed, should *not* be changed, or are most often changed

## Default Database Settings: Best Practices (1)

- **Collation**
  - Defaults to the collation set for model unless changed during database creation
  - Potential performance issues if collations do not match
  - Collation mismatches can also stop code execution
- **Recovery model**
  - Three options:
    - Simple, Bulk-logged, Full
    - Default for model is Full
  - Affects:
    - How some database transactions are logged
    - Whether transaction log backups are required
    - Restore options
  - Discussed in more detail in Module 4

## Default Database Settings: Best Practices (2)

- **Compatibility level**
  - Determines the SQL Server version with which the database is compatible
    - This has nothing to do with being able to restore/attach to a lower version
  - Restoring a database to a newer version does not update the compatibility level
  - An older compatibility level may limit what queries you can execute against a database
  - Noteworthy in SQL Server 2014 due to significant changes in the cardinality estimator with compatibility level 120
  - Noteworthy in SQL Server 2016 and higher due to more changes in the cardinality estimator in addition to other enhancements, for example:
    - Trace flag 2371 enabled
    - Statistics sampling is multi-threaded
    - Queries for memory-optimized tables can have parallel plans
    - Adaptive joins and batch mode memory grant feedback
    - <https://msdn.microsoft.com/en-us/library/bb510680.aspx>

## Default Database Settings: Best Practices (3)

- **Auto Create Statistics**
  - Enabled by default
  - Leave enabled
  - Creates single column statistics when the optimizer needs them
- **Auto Update Statistics**
  - Enabled by default
  - Leave enabled, but treat it as a safety measure
  - Implement maintenance jobs to actively manage statistics

## Additional Database Settings (1)

- **Auto Update Statistics Asynchronously**
  - Disabled by default
  - If enabled, the original query plan is used and updates for invalidated statistics will happen after query execution
  - If disabled, updates for invalidated statistics will happen before query execution, and a new query plan will be generated
- **Parameterization**
  - SIMPLE by default
  - Only change to FORCED if thorough testing has shown benefits
- **Page Verify**
  - CHECKSUM by default for SQL Server 2005 and higher
  - Do not change

## Additional Database Settings (2)

- **Auto-shrink**
  - Disabled by default
  - Never use auto-shrink; it should always be disabled
  - This is probably the only setting there is NOT an 'it depends' answer for
  - This should be one of the first things that gets checked when you take over a new database
  - Data file shrink (same code as auto-shrink) is almost guaranteed to introduce index fragmentation within the data files
  - The database will most likely just auto-grow again, and then auto-shrink, auto-grow in a cycle that wastes resources
  - You can't control when it kicks in and affects performance
  - Blog post: Auto-shrink – turn it off!
    - <https://www.sqlskills.com/ie0/autoshrinkoff>

## Changes to Database Settings in SQL Server 2016

- **SQL Server 2016 added new syntax that allows you to configure the following options for each database:**
  - Max Degree of Parallelism
  - Cardinality Estimator version
    - Use TF 9481 to use old CE at the instance-level
  - Parameter sniffing
    - Use TF 4136 to set instance-wide
  - Query optimization hotfixes
    - Use TF 4199 to set instance-wide
- **The syntax also allows you to clear procedure cache for a database**
  - Can clear a specific plan handle in SQL Server 2019+
- **Settings can be different on the primary vs. the secondaries in an Availability Group**

## Additional Database Settings in SQL Server 2017

- **Identity cache**
- **Interleaved Execution for multi-statement table-valued functions**
  - Requires compatibility mode 140
- **Batch mode memory grant feedback**
  - Requires compatibility mode 140
- **Batch mode adaptive joins**
  - Requires compatibility mode 140



## Additional Database Settings in SQL Server 2019

- **T-SQL Scalar UDF inlining**
  - Requires compatibility mode 150
- **Automatically elevate supported operations to resumable**
- **Row mode memory grant feedback**
  - Requires compatibility mode 150
- **Batch mode on rowstore**
  - Requires compatibility mode 150
- **Deferred compilation for table variables**
  - Requires compatibility mode 150
- **Accelerated plan forcing**
- **Lightweight query profiling**
- **Display error message: String or binary data would be truncated**
- **Collect last query plan statistics**
- **Determine whether row-level security affects execution plan**

## Additional Database Settings in Azure SQL DB

- **Automatically elevate supported operations to online**
- **Optimize for ad hoc workloads**
- **Collect execution statistics for natively compiled T-SQL modules**
- **Automatically drop temporary tables**
- **Configure pause duration for resumable index activity**
- **Wait for Sch-M lock on a low priority queue for asynchronous statistics updates**

## Demo

Finding and changing database settings

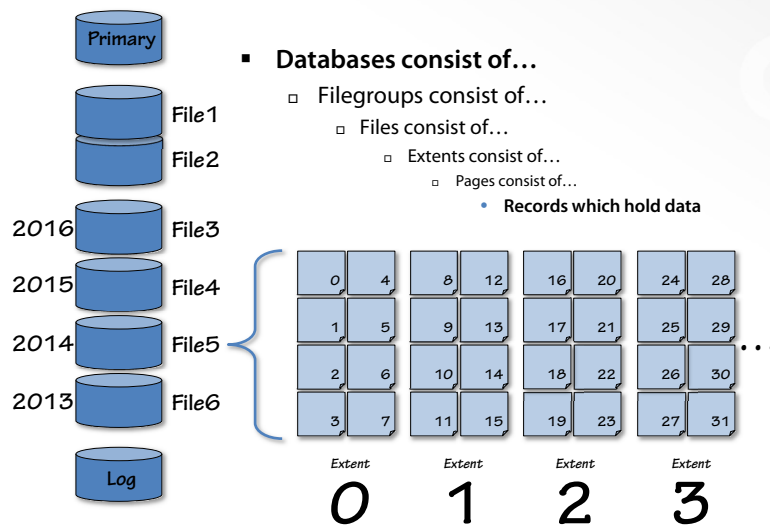
## Cardinality Estimator and Upgrades

- The cardinality estimator (CE) component was significantly changed in SQL Server 2014
  - First redesign since SQL Server 7.0
- Initially tied to compatibility mode in SQL Server 2014 (120)
- Moved to a database-scoped configuration in SQL Server 2016
- If you are upgrading from SQL Server 2012 or below, *it is absolutely essential* that you test query performance
  - There are many combinations of settings which can cause changes query performance
  - The Importance of Database Compatibility Level in SQL Server
    - <https://www.sqlskills.com/blogs/glenn/database-compatibility-level-in-sql-server/>
  - Avoiding SQL Server Upgrade Performance Issues
    - <https://www.sqlskills.com/blogs/glenn/avoid-sql-server-upgrade-performance-issues/>

## Overview

- Installation best practices
- Configuration best practices
- Setting up database mail and alerts
- Database settings
- Database file layouts

## Database Components



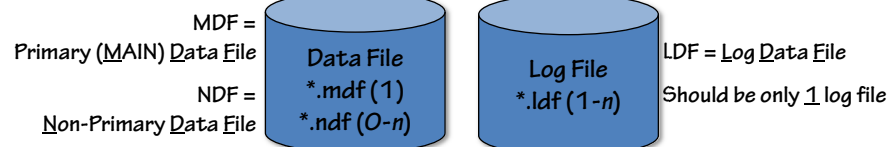
## Database Structure

Up to 32,767 databases per instance



Up to 32,767 files per database  
(Total is for both data and log files)

Minimum of 2 files – one for data, one for log



## Filegroups and Files

- **A filegroup is a logical container**
  - Filegroups contain one or more files
  - A table or index is completely contained in a single filegroup
- **Every database has at least one filegroup – the PRIMARY filegroup**
  - Contains at least one file (.mdf)
  - Additional data files can be added to the primary filegroup (.ndf)
- **One or more secondary filegroups can be defined**
  - Each secondary filegroup can have one-to-many data files
- **There are many reasons for creating multiple filegroups and multiple files**
  - e.g. manageability, performance, availability

## Notes on Auto-Growth

- **When a file is full, it will grow to provide more space**
  - By default, new space is zero-initialized, which takes time
  - Allocation pauses while file growth takes place
  - Very often the auto-growth default is not changed, which leads to massive amounts of auto-growth
    - 1MB in SQL 2005+, 10% in previous versions (10% still for log files)
- **Best practice to manually manage file sizes**
  - Manually growing files allows YOU to decide which files to grow, by how much, and when
  - Allows file fragmentation to be minimized
  - Monitor file usage and alert when threshold reached
- **Auto-grow should always be left enabled 'just in case'**
  - If it's off, and no-one monitors space, the workload could stop
  - Monitor auto-growth using Extended Events (SQL Trace prior to 2012)
  - Always set file growth to a fixed amount, not a percentage

## Demo

Viewing and altering filegroups, files, and auto-growth

## Key Takeaways

- **Installation requires preparation and often requires coordination with other teams (server, storage, etc.)**
  - Considerations include:
    - Security
    - Maintenance
    - Configuration
- **Proper configuration is necessary at the server, instance, and database level**
  - There are many settings that do *not* need to be changed
    - Understand what needs to be altered and why

## Resources

- **Installation**
  - SQL Server: Installing and Configuring SQL Server 2016 (Pluralsight course)
    - <https://www.pluralsight.com/courses/sqlserver-2016-installing-configuring>
  - Provisioning a New SQL Server Instance – Part One
    - <https://www.sqlskills.com/IE0/newinstall1>
  - Provisioning a New SQL Server Instance – Part Two
    - <https://www.sqlskills.com/IE0/newinstall2>
  - Provisioning a New SQL Server Instance – Part Three
    - <https://www.sqlskills.com/IE0/newinstall3>
  - SQL Server: Installing and Configuring SQL Server 2016 (Pluralsight course)
    - <https://app.pluralsight.com/library/courses/sqlserver-2016-installing-configuring/table-of-contents>

## Resources

### ▪ Security

- Configure Windows Service Accounts and Permissions
  - [http://msdn.microsoft.com/en-us/library/ms143504.aspx#Serv\\_Perm](http://msdn.microsoft.com/en-us/library/ms143504.aspx#Serv_Perm)
- How to make unwanted access to SQL Server 2005 by an operating system administrator more difficult
  - <http://support.microsoft.com/kb/932881>
- SQL Server 2005 connectivity and Volume Shadow Copy Services (VSS)
  - <http://support.microsoft.com/kb/919023>

### ▪ Configuration

- SQL Collation and related performance impact, viewing collation in query plans
  - [http://blogs.msdn.com/b/sql\\_pfe\\_blog/archive/2013/02/04/sql-collation-and-related-performance-impact-viewing-collation-in-query-plans.aspx](http://blogs.msdn.com/b/sql_pfe_blog/archive/2013/02/04/sql-collation-and-related-performance-impact-viewing-collation-in-query-plans.aspx)



93

© SQLskills, All rights reserved.  
<https://www.sqlskills.com>

## Resources

### ▪ tempdb

- KB: How to use Detach and Attach functions to move SQL Server databases
  - <http://support.microsoft.com/kb/224071>
- WP: Working with tempdb in SQL Server 2005
  - <http://www.microsoft.com/technet/prodtechnol/sql/2005/workingwithtempdb.msp>
- Blog post: Misconceptions Around TF 1118
  - <https://www.sqlskills.com/ie0/t1118misconceptions>
- Blog post series on tempdb
  - <https://www.sqlskills.com/ie0/tempdbseries>
- Books Online:
  - Troubleshooting Insufficient Disk Space in Tempdb
    - <http://msdn.microsoft.com/en-us/library/ms176029.aspx>
  - Capacity Planning for Tempdb
    - <http://msdn.microsoft.com/en-us/library/ms345368.aspx>



94

© SQLskills, All rights reserved.  
<https://www.sqlskills.com>

## Appendix

### Reasons to Create Multiple Filegroups

- **Segregate or distribute I/O**
  - When you create a table or index you can specify a filegroup
  - The data is written to the file(s) in the filegroup, which can exist on one drive, or spread across multiple drives
  - Any number of objects can exist within a filegroup (1...n)
- **Separate read/write and read-only data**
  - Filegroups with read-only data can be placed on slower disks
  - Filegroups with read/write data can use Tier 1 disks to optimize performance
  - Also permits file and filegroup backups which can help manage backup size, and backup/restore time
- **Partial database availability**
  - In the event of a disaster, the entire database does not need to be restored in order for it to be brought online
  - Faster recovery
  - Enterprise Edition only



## Considerations When Creating Filegroups

- **It is not recommended to create one filegroup per table**
  - This can cause significant delays when opening the database
- **Tables and indexes can be placed in separate filegroups**
  - However, if the filegroup(s) that store any indexes are unavailable, data cannot be inserted/updated/deleted
  - In addition, if specific activities in the application require access to multiple tables, which exist in multiple filegroups, problems can exist if not all filegroups are online
- **Filegroups create more administrative overhead**
  - Filegroup design is part of the solution design – not *just* the database design – don't take it lightly

## Reasons to Create Multiple Files

- **Distribute reads and writes across multiple files**
  - SQL Server uses two allocation algorithms to decide where data should be written when multiple files exist
  - This can improve performance
  - Investigating the proportional fill algorithm
    - <https://www.sqlskills.com/blogs/paul/investigating-the-proportional-fill-algorithm/>

## Considerations When Creating Files

- **At a minimum, create 2-4 files per filegroup**
  - This provides better performance than a single data file
- **Create files that are the same size (and pre-sized so they do not have to auto-grow)**
- **Set fixed auto-growth values that are the same for all files**
  - Think of auto-growth as a safety net
- **Capacity planning considerations**