

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 pieces 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**
 - 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 you must make**
 - Current release is 2012, you may have an earlier version available in-house
 - 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
- Provision Domain Accounts
 - Grant additional rights
- Install and patch operating system
- Update BIOS and firmware
- Install/update device drivers
- Verify power management settings
- Perform final server configuration tasks
- Decide what SQL Server features to install

Domain Accounts and Additional Rights

- **Obtain domain accounts for services**
 - You may use one account for all services, or one account *per* service
 - Separate accounts is recommended
 - Assign minimal permissions
 - SQL Server installation will grant necessary rights for each account
 - Accounts do not need to be Local Administrators on the server
 - You will need the name and password for each account for the actual installation
- **Grant rights to the SQL Server service account**
 - Perform volume maintenance tasks (Instant File Initialization)
 - Lock pages in memory

Instant File Initialization

- Allows SQL Server 2005 and higher to use Windows technology to skip the zeroing process for new data files
- If you can, turn it on
- Cannot be enabled from within SQL Server
- Requires “Perform Volume Maintenance Tasks” security permission granted to SQL Server service account or group
- 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
 - Can grant this to a lower privileged account
- Instant initialization happens automatically once SQL Server is restarted after granting the permission
- Only applies to data files

Instant File Initialization in Action

- **Hardware configuration – Dell Precision 670**
Dual Proc (x64) with Dual Core, 4 GB Memory RAID 1+0 array w/4-142 GB, 15000rpm disks
- **Software configuration: SQL 2005**
 - With zero initialization
 - 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
 - With instant file initialization
 - CREATE DATABASE with 20 GB Data file = 1.3 seconds
 - ALTER DATABASE BY 10 GB = 0.4 seconds
 - RESTORE 30 GB DATABASE (EMPTY Backup) = 5 seconds
 - RESTORE 30 GB DATABASE (11 GB Backup) = 19:42 minutes



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 log: Using locked pages for buffer pool

Install and Patch Windows

- **Install Windows**
 - Verify version and edition to support SQL Server version and edition, 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 allow Windows to automatically download and install 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

- **BIOS**
 - Power management in the BIOS setup application should be set to OS control, or disabled
- **Windows**
 - Set the power plan to High Performance
 - Default is Balanced
 - This can have a negative effect on database server performance
 - Verify using CPU-Z (www.cpuid.com)
 - May need to update hardware power management in the BIOS as well

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>
- Manually optimize your C: drive
- 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 features will be installed (SSIS, SSRS, etc.)**
 - This will be dependent on what's needed by the applications the databases will 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 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

Decide these items before you begin installation!

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 (< SQL Server 2008)
- **Instance settings**
 - Max server memory (MB)
 - Max degree of parallelism
 - Cost threshold for parallelism
 - Compress backup
 - Optimize for Ad hoc Workloads

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 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:

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 > 150-300MB and adjust as necessary
- Post: How much memory does my SQL Server actually need?
 - <http://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
 - For NUMA systems, set MAXDOP equal to the number of physical cores in a single processor

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
 - <http://www.sqlskills.com/ie0/tuningcostthreshold>

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 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)

Optimize for Ad hoc Workloads

- Not enabled by default
- Available as of SQL Server 2005
- 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:
 - <http://www.sqlskills.com/ie0/adhocworkloadanalysis>

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

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
 - Except when SQL Server is clustered, then use Cluster Manager for restarts, etc.
- Specify Startup Parameters via Configuration Manager as well
 - -T3226
 - Do not log successful backup messages to 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
- **Usage 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

In-Memory 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
 - <http://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 In-Memory Contention

- **Trace flag 1118 removes use of mixed extents (single-page allocations) in all databases**
 - Removes SGAM contention
 - Should be enabled by default
 - <http://support.microsoft.com/kb/328551>
- **Create multiple data files to spread and reduce contention**
 - Bob Ward (CSS) at PASS 2011:
 - < 8 cores, use #files = #cores
 - > 8 cores, use 8 files, increasing by chunks of 4
 - Start with Bob's recommendations and monitor for contention
 - Be careful of having too many data files
- **Note: this does not apply to log files**

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**
 - General query workload
 - Index maintenance operations
- **Set tempdb size to resulting size**
- **Enable appropriate auto-growth, equal on all files**
- **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 post Configuring Alerts for High Severity Problems
 - <http://www.sqlskills.com/ie0/configurealerts>
- **You want to be notified about these 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

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
 - 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
 - An older compatibility level may limit what queries you can execute against a database
- **Auto Shrink**
 - Disabled by default
 - Do not ever enable it
 - Creates fragmentation within the database, extremely slow process

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

- **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

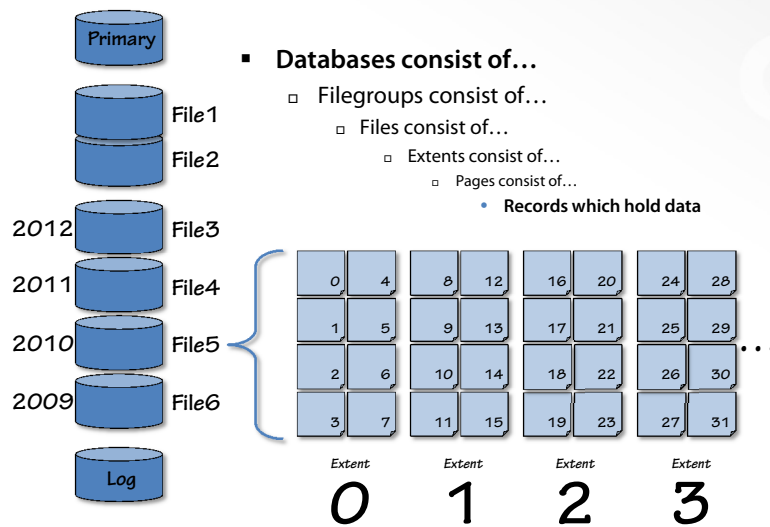
Demo

Finding and Changing Database Settings

Overview

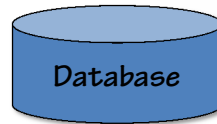
- Installation best practices
- Configuration best practices
- Setting up database mail and alerts
- Database settings
- Database file layouts

Database Components



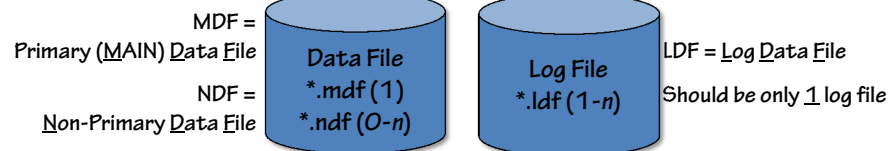
Database Structure

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



Files and Filegroups

- **A filegroup contains 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)
 - Multiple data files can be added to the filegroup (.ndf)
- **Multiple 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**
 - 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 SQL Trace or Extended Events
 - Always have file growth to be a fixed amount, not a percentage

Auto-Shrink

- **Never use auto-shrink**
 - 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
- **Why is auto-shrink bad?**
 - 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!
 - <http://www.sqlskills.com/ie0/autoshrinkoff>
- **Again, always make sure auto-shrink is turned off**

Resources

■ Installation

- Provisioning a New SQL Server Instance – Part One
 - <http://www.sqlskills.com/IE0/newinstall1>
- Provisioning a New SQL Server Instance – Part Two
 - <http://www.sqlskills.com/IE0/newinstall2>
- Provisioning a New SQL Server Instance – Part Three
 - <http://www.sqlskills.com/IE0/newinstall3>
- SQL Server 2012: Installation and Configuration (Pluralsight course)
 - http://www.sqlskills.com/IE0/installconfigure_ps



57

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

Resources

■ Security

- Configure Windows Service Accounts and Permissions
 - 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



58

© SQLskills, All rights reserved.
<http://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
 - <http://www.sqlskills.com/ie0/t1118misconceptions>
- Blog post series on tempdb
 - <http://www.sqlskills.com/ie0/tempdbseries>



Resources

■ tempdb

- 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>



Appendix

Reasons to Create Multiple Filegroups

- **Segregate or distribute IO**
 - 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

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**