



Professional Association for SQL Server

Replacing SQL Maintenance Plans with Ola Hallengren's Maintenance Solution

Extract

SQL Maintenance Plans provide a quick method to create and schedule, simple database maintenance. As your database environment grows, the built-in tools in the SQL maintenance toolbox may prove insufficient. Ola Hallengren has developed a series of stored procedures which together provide a maintenance solution with the additional flexibility and features required to effectively manage your databases. I will demonstrate how to implement this solution and discuss maintenance strategy.



Professional Association for SQL Server

SQL Maintenance Plans

- Built in maintenance plans in SSMS since SQL 2005
- Wizard and Maintenance Plan Design provide a quick and easy way to manage databases
- Great for accidental/beginning DBA's
- Quick Temporary or One-Off maintenance tasks
- As your databases grow, as your organizations needs grow you may run into some of the limitations of maintenance plans



Professional Association for SQL Server

SQL Maintenance Plan Limitations

- Backup
 - No CHECKSUM Option when backing up a database
 - Cannot verify a backup with CHECKSUM, critical if not doing DBCC CHECKDB nightly (see addendum)
- Indexing
 - Rebuild Index and Reorganize Index Tasks can only be implemented with limited logic.
 - When configured, the tasks run regardless of fragmentation of the index (whether it needs it or not).
 - Lack of Granularity can lead to
 - Unnecessary log growth especially defragmenting large indexes,
 - Unnecessary locking of indexes (if online re-indexing is not available)
 - Unnecessarily long maintenance windows
 - Resource intensive
- Statistics Update
 - Table/View index statistics updated regardless of previous index maintenance
 - Rebuilding an index will automatically update index statistics
 - Ends up running twice = unnecessarily long maintenance windows
 - This stats update may not be as accurate



Professional Association for SQL Server

SQL Maintenance Plan Limitations

- Maintenance Cleanup
 - No logic other than deleting files with a specific extension based on age
- History Cleanup
 - No logic other than deleting history with a specific extension based on age
- Shrink Database Task !!!
 - Not a best practice
 - Database fragmentation
- Maintenance Plans: They don't like it when DBAs leave
 - <http://www.airbornegeek.com/2010/10/maintenance-plans-they-don%e2%80%99t-like-it-when-dbas-leave-2/>



Professional Association for SQL Server

A Better Solution

- Using T-SQL Scripts or Stored Procedures in combination with SQL Server Agent jobs
 - Offers 100% flexibility
 - OS Access
 - Portability
- You could develop your own but ...
 - As complexity grows so are the chances of errors
 - Requires deep understanding of SQL Server internals (steep learning curve)
 - Debugging can get very time-consuming
- Using a publicly available script that has been shared and vetted by a community of users
 - <http://ola.hallengren.com/>
 - <http://sqlfool.com/2009/06/index-defrag-script-v30/>
 - http://www.grics.qc.ca/YourSqlDba/index_en.shtml
 - <http://www.simple-talk.com/sql/database-administration/sql-server-tacklebox-free-ebook/>
- Could use Powershell Scripts but...
 - Same issues as above
 - Need to learn a brand new scripting language
 - Eventually you will probably have to learn, but till then....



Professional Association for SQL Server

Ola Hallengrens Maintenance Solution

- Ola Hallengren – DBA for large financial firm in Sweden
- Organizations that use his SQL maintenance solution
 - Citibank
 - Well Fargo
 - Target
 - Several Universities
 - City of Las Cruces, NM
 - Whole bunch of other companies and organizations
- Updated regularly and backwards compatible with older versions
- Where to get it? <http://ola.hallengren.com/>



Professional Association for SQL Server

Summary

- Covers most common database tasks
- Full, Differential, T-log backups of user and system databases
- Integration with 3rd party backup solutions
- Dynamic Index optimization reorganizes, rebuilds or ignores indexes based on fragmentation, page count. Online or Offline rebuild option use determination..
- Database Integrity Checks can check entire databases (DBCC CHECKDB) or granular (ie DBCC CHECKTABLE, CHECKALLOC, CHECKCATALOG, CHECKFILEGROUP)
- Maintenance Clean Up to delete backup and job history and clean up logs
- Log events to Maintenance database
- Option to create all the maintenance jobs automatically by running the MaintenanceSolution.sql script (simply schedule the jobs to run) - <http://ola.hallengren.com/downloads.html>
- Or implement scripts as needed to create a custom maintenance solution that is based on a solid framework



Professional Association for SQL Server

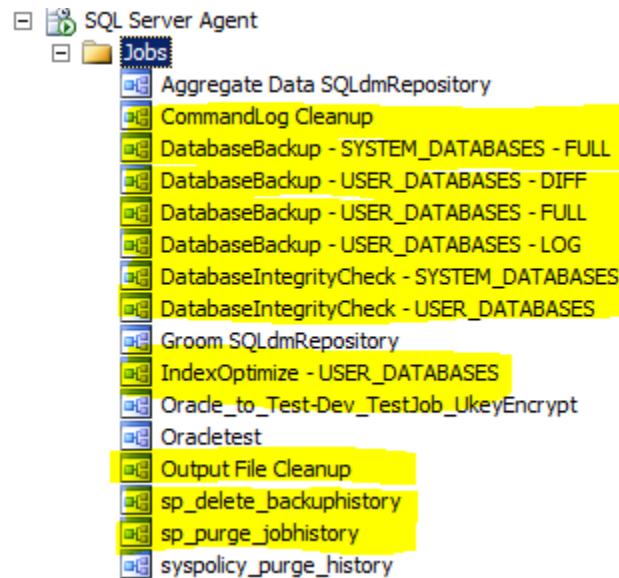
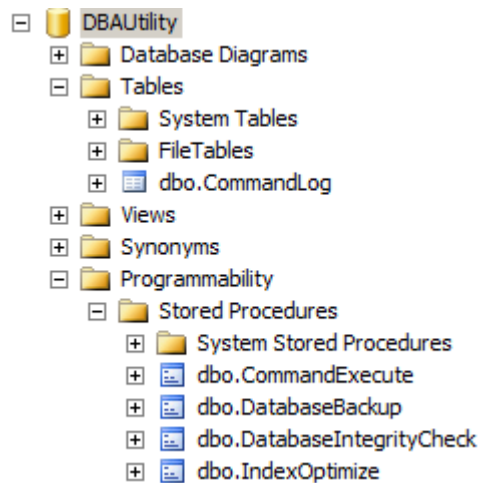
One Step Implementation

- Compatible with SQL 2005 and up
- Open query window in SSMS and Paste the MaintenanceSolution.sql script
- Note – the script will create several stored procedures and a Command log table. By default these objects are created in the MASTER database. Recommend using a DBA utility database. (Edit Script)
- Edit the Script
 - USE [DBAUtility]
 - SET @BackupDirectory = N'C:\SQLBackups' -- Specify the backup root directory.
- Execute the query



Professional Association for SQL Server

Objects Created by Solution



Professional Association for SQL Server

One Step Implementation

- Go into [SQL Server Agent] / [Jobs] and start the jobs that have been created. Verify that these jobs are completing successfully. Verify that the backup files are being created. Check the output files in the error log directory.
- Schedule the jobs.



Professional Association for SQL Server

Issues with One Step

- Everything will work but...
- Not if you have a non-default install of SQL Server
 - Ex. Output File Cleanup job
- NO SCHEDULES CONFIGURED
- No strategy for scheduling jobs (how often should DBCC be run?)
- No alerts or notifications are configured (recommend)
- Backups are deleted after 24 hours!
- Backing up to local drive not recommended
- No built in option to backup to multiple locations
- Implementation won't work for SQLExpress (there is a workaround)
- You need to review the default settings (parameters) of each job and understand what they mean



Professional Association for SQL Server

Under the Hood

- To make the most of the solution you need
 - To understand how it works
 - Apply an effective schedule (for your environment)
- Script is divided into 7 sections
 - Essential Settings
 - Creating the Command Log table
 - Creating the DatabaseBackup stored procedure
 - Creating the DatabaseIntegrityCheck stored procedure
 - Creating the IndexOptimize stored procedure
 - Creating the maintenance jobs



Professional Association for SQL Server

Essential Settings

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

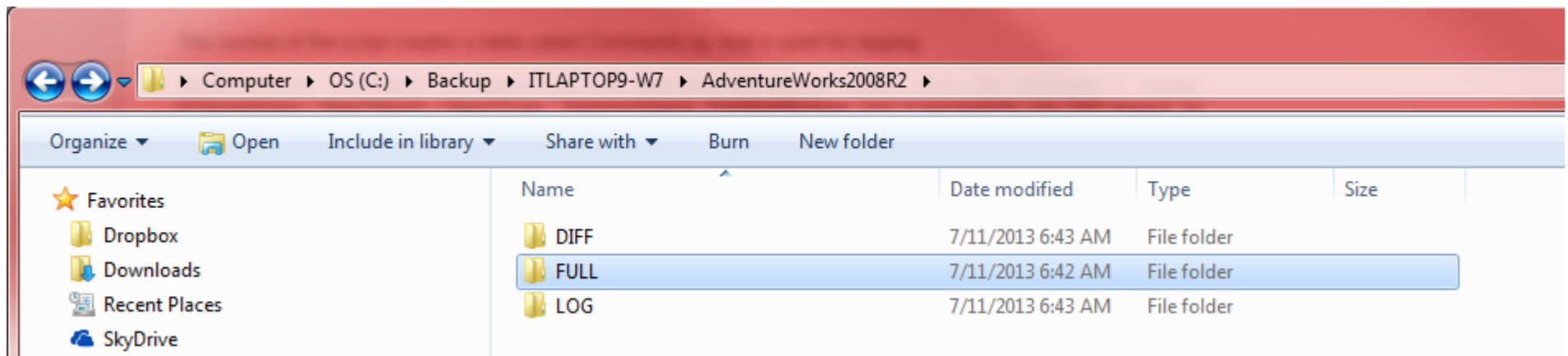
- The default database the solution will install to is MASTER. Recommend creating a separate DBA database for this and other utilities
- Default Jobs Created (no schedule)
 - Use as is
 - Can use as a template (delete later) to create your own jobs
- Edit the Backup Directory (consider network share)
- The output file directory will null is actually set to the default SQL Server error log location
 - What if you have a non-default install? Ex programs installed on D:
- Logging to the LogTable always recommended (you should leave alone)



Professional Association for SQL Server

Essential Settings

- Note – the first time the backup job is run, a subdirectory is created for the instance and subdirectories for each database are created under them. Furthermore, separate subdirectories for full, differential and log backups are created in each database folder (if you run these backup jobs).



Professional Association for SQL Server

CommandExecute SPROC

- Internal use only (don't need to mess with)
- The core functionality of the solution relies on the CommandExecute stored procedure. It is required for the other three stored procedures to work.
- Purpose to ensure consistent error handling and logging
- Useful general way of executing strings of TSQL commands. It records each call, together with the time it was done, to the SQL Server log, and, once the command is executed, it logs whether it was successful and how long it took.



Professional Association for SQL Server

DatabaseBackup SPROC

- 24 Different parameters (open up stored procedure, open job), Only first three parameters are required (if you setup your own job)
- See Ola's website for examples, explanations are a bit thin

	Name	Description		Name	Description
1	Databases	Select databases. User_Databases, System_Databases, <yourDatabase> <DB1,DB2,DB...>	13	MaxTransferSize	Specify the largest unit of transfer, in bytes, to be used between SQL Server and the backup media.
2	Directory	Specify the backup root directory, which can be a local directory or a network share. @Directory = NULL will use default setting when installing the solution	14	NumberOfFiles	Specify the number of backup files. (non – NTFS backup volume?)
3	BackupType	Specify the type of backup: full, differential, or transaction log.	15	CompressionLevel	Set the Quest LiteSpeed for SQL Server, Red Gate SQL Backup, or Idera SQL safe backup compression level.
4	Verify	Verify the backup. (Recommend always set to “Y”)	16	Description	Enter a description for the backup. (Will be written to log)
5	CleanupTime	Specify the time, in hours, after which the backup files are deleted. (Edit to suit your retention policy)	17	Threads	Specify the Quest LiteSpeed for SQL Server, Red Gate SQL Backup, or Idera SQL safe backup number of threads.
6	Compress	Compress the backup. @Compress = NULL will take default server setting	18	Throttle	Specify the Quest LiteSpeed for SQL Server maximum CPU usage, as a percentage.
7	CopyOnly	Perform a copy-only backup.	19	Encrypt	Encrypt the backup. (3rd party encryption types, TDE??)
8	ChangeBackupType	Change the backup type if a differential or transaction-log backup cannot be performed.	20	EncryptionType	Specify the type of encryption. (3rd party encryption types)
9	BackupSoftware	Specify third-party backup software; otherwise, SQL Server native backup is performed.	21	EncryptionKey	Enter the key that is used to encrypt the backup.
10	Checksum	Enable backup checksums. (Highly recommend,, use to validate backups RESTORE VERIFYONLY WITH CHECKSUM v DBCC)	22	ReadWriteFileGroups	Perform a backup of the primary filegroup and any read/write filegroups.
11	BlockSize	Specify the physical blocksize in bytes.	23	LogToTable	Log commands to the table CommandLog.
12	BufferCount	Specify the number of I/O buffers to be used for the backup operation.	24	Execute	Execute commands.

Database Integrity SPROC

- 10 Different parameters (open up stored procedure, open job), Only first parameter is required (if you setup your own job)
- See Ola's website for examples, explanations are a bit thin

	Name	Description
1	Databases	Select databases.
2	CheckCommands	Specify the integrity check commands to be performed: checkdb, checkfilegroup, checktable, checkalloc and/or checkcatalog.
3	PhysicalOnly	Limit the checks to the physical structures of the database. (For Quick Check, If scheduling should have time to do full)
4	NoIndex	Do not check nonclustered indexes. (, If scheduling should have time to do full)
5	ExtendedLogicalChecks	Perform extended logical checks. (By default this is enabled, checks to see if values stored match datatype of column, NOT AVAILABLE IN BUILT-IN MAINTENANCE PLAN)
6	TabLock	Use locks instead of an internal database snapshot.
7	FileGroups	Select filegroups.
8	Objects	Select objects.
9	LogToTable	Log commands to the table CommandLog. (Always Leave)
10	Execute	Execute commands.



Professional Association for SQL Server

DatabaseOptimize SPROC

- 22 Different parameters (open up stored procedure, open job), Only first parameter is required rest are set to defaults (if you setup your own job), See Ola's website for examples, explanations are a bit thin

	Name	Description		Name	Description
1	Databases	Select databases.	12	LOBCompaction	Compact pages that contain large object (LOB) columns, when reorganizing indexes. (NVARCHAR(MAX), VARBINARY)
2	FragmentationLow	Specify index maintenance operations to be performed on a low-fragmented index. (Default < 5, See options below)	13	UpdateStatistics	Update statistics. (ALL, INDEX, COLUMNS, NULL (is Default). You want schedule job set to ALL weekly)
3	FragmentationMedium	Specify index maintenance operations to be performed on a medium-fragmented index. (Default >5, < 30 results in Index Reorganize, See options below)	14	OnlyModifiedStatistics	Update statistics only if any rows have been modified since the most recent statistics update. (If you set above, then save time by setting this Y)
4	FragmentationHigh	Specify index maintenance operations to be performed on a high-fragmented index. (Default >30 results in Index Rebuild, See options below)	15	StatisticsSample	Indicate, as a percentage, how much of a table is gathered when updating statistics. (Server default used if NULL, ensure auto update on statistics is left on if so)
5	FragmentationLevel1	Set the lower limit, as a percentage, for medium fragmentation (Default 5)	16	StatisticsResample	Update statistics with the most recent sample.
6	FragmentationLevel2	Set the lower limit, as a percentage, for high fragmentation. (Default 30)	17	PartitionLevel	Maintain partitioned indexes on the partition level.
7	PageCountLevel	Set a size, in pages; indexes with fewer pages are skipped for index maintenance. (Default 1000 pages /8KB per page/ 128 page per MB, MS whitepaper)	18	TimeLimit	Set the time, in seconds, after which no commands are executed. (if working in a constrained maintenance window)
8	SortInTempdb	Use tempdb for sort operations when rebuilding indexes. (only in ENT or DEV)	19	Indexes	Select indexes. (Default is all indexes)
9	MaxDOP	Specify the number of CPUs to use when rebuilding indexes. (Are you reindexing during operations???)	20	Delay	Set the delay, in seconds, between index commands.
10	FillFactor	Percentage, how full the pages should be made when rebuilding indexes. (Server Default 0, should change to avoid splits. See link)	21	LogToTable	Log commands to the table CommandLog. (Set to Y)
11	PadIndex	Apply the percentage of free space that the FillFactor parameter specifies to the intermediate-level pages of the index. (May want to consider if have Very large indexes. See link)	22	Execute	Execute commands.

DatabaseOptimize SPROC

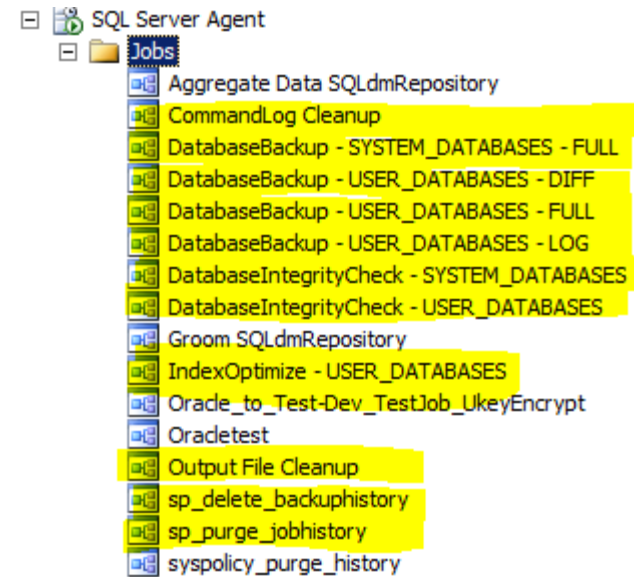
- INDEX_REBUILD_ONLINE means rebuild indexes online, INDEX_REBUILD_OFFLINE means rebuild indexes offline, and INDEX_REORGANIZE means reorganize indexes
- Can specify multiple index operations for a fragmentation group; e.g. 'INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE'. Because it is not always possible to rebuild an index online in SQL Server (If You don't have DEV or ENT edition. So 'INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE' tells the script to rebuilt the index online if possible, otherwise offline.
- All these parameters allow the stored procedure to analyze each index and perform maintenance on it individually. Thus providing speed, maximizing optimization and maximizing efficiency while using as few resources as possible.

Value	Description
INDEX_REBUILD_ONLINE	Rebuild index online.
INDEX_REBUILD_OFFLINE	Rebuild index offline.
INDEX_REORGANIZE	Reorganize index.
INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE	Rebuild index online. Rebuild index offline if online rebuilding is not supported on an index.
INDEX_REBUILD_ONLINE,INDEX_REORGANIZE	Rebuild index online. Reorganize index if online rebuilding is not supported on an index.
INDEX_REORGANIZE,INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE	Reorganize index. Rebuild index online if reorganizing is not supported on an index. Rebuild index offline if reorganizing and online rebuilding are not supported on an index.
NULL	Do not perform index maintenance. This is the default for a low-fragmented index.



The Jobs

- Jobs all use default settings
- None have been scheduled
- Edit the jobs (not the stored procedures) or create new ones
- Don't have to use all (or any) of the jobs (delete as needed)
- All jobs use the sqlcmd utility (in Tools\Binn folder) to execute the stored procedures.
- Why?
 - Sqlcmd allow T-SQL statements to be executed from command line
 - Can use either SQLAgent or Task Scheduler (if using SQLExpress)
- **NOTE: sqlcmd string must all be on one line (else won't work and give no error)**
- Jobs use Tokens `$(ESCAPE_SQUOTE(token))`, similar to `@@function`. Example `$(ESCAPE_SQUOTE(SRVR))`. See BOL for list of tokens



SQLCMD

- Open one of the DatabaseBackup, IntegrityCheck or IndexOrganize job
- Note Type

Step name:
DatabaseBackup - USER_DATABASES - FULL

Type:
Operating system (CmdExec)

Run as:
SQL Server Agent Service Account

Process exit code of a successful command: 0

Command:
sqlcmd -E -S \$(ESCAPE_SQUOTE(SRVR)) -d DBA -Q "EXECUTE [dbo].[DatabaseBackup] @Databases = 'AdventureWorks2008R2', @Directory = N'C:\Backup', @BackupType = 'FULL', @Verify = 'Y', @CleanupTime = 24, @Checksum = 'Y', @LogToTable = 'Y'" -b

Open...

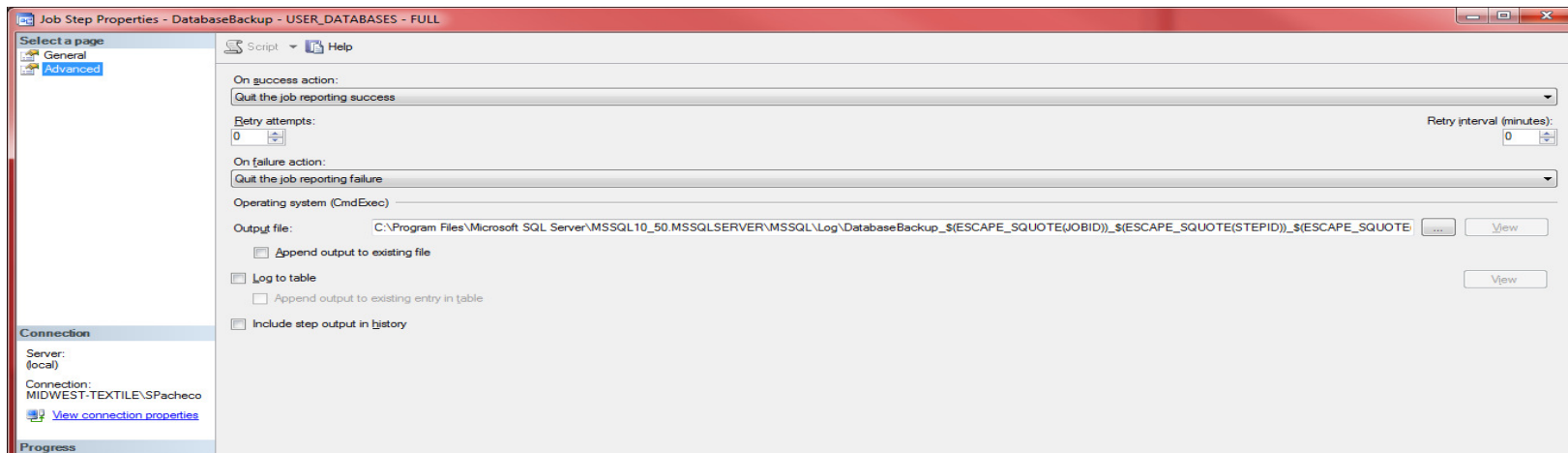
```
sqlcmd -E -S $(ESCAPE_SQUOTE(SRVR)) -d DBA -Q "EXECUTE [dbo].[DatabaseBackup]  
@Databases = 'AdventureWorks2008R2', @Directory = N'C:\Backup', @BackupType = 'FULL',  
@Verify = 'Y', @CleanupTime = 24, @Checksum = 'Y', @LogToTable = 'Y'" -b
```

- -E trusted command
- -S Servername (which is replaced by token)
- -d database (in this case DBA)
- -Q sqlcmd syntax says that the following quoted content (T-SQL code) is to be executed (SPROC using variables)
- -b "on error batch abort". On error, sqlcmd exits returning to a command prompt. An ERRORLEVEL value is returned which is recorded in the log file. If an error does happen, the stored procedure will continue to execute until it is done.



Professional Association for SQL Server

Output File



- The output file will record valuable information
- On the job properties click Advanced
- The default location is determined by the output file which is built using tokens
- C:\Program Files\Microsoft SQL Server\MSSQL10_50\MSSQLSERVER\MSSQL\Log\DatabaseBackup_\$(ESCAPE_SQUOTE(JOBID))_\$(ESCAPE_SQUOTE(STEPID))_\$(ESCAPE_SQUOTE(STRDT))_\$(ESCAPE_SQUOTE(STRTTM)).txt
- The JOBID, STEPID, STRDT (Start Date), STRTTM (Start Time) “uniqueify” the filename
- Open error file C:\Program Files\Microsoft SQL Server\MSSQL10_50\MSSQLSERVER\MSSQL\Log



Professional Association for SQL Server

Cleanup

- CommandLog Clean up – deletes rows from the Commandlog table older than 30 days (default setting)
- Output file Cleanup - deletes output log files older than 30 days (default setting)
- sp_delete_backuphistory – MSDB maintenance, deletes backup history from the msdb database that is over 30 days (default setting) old using the system stored procedure sp_delete_backuphistory
- sp_purge_jobhistory – MSDB maintenance, deletes job (SQL Server Agent jobs) history from the msdb database that is over 30 days (default setting) old using the system stored procedure sp_purge_jobhistory.
- Each of these jobs can be run on a schedule of your choosing (daily, weekly, monthly etc)
- Can run separately as is, or combine them as separate steps in one job running on one schedule.
- NOTE on combining jobs into separate steps, you need to include an error handling step. To configure the job to run all steps and report failure if any steps fail you need to add a step at the end that runs the following T-SQL

```
IF EXISTS(
SELECT * FROM msdb.dbo.sysjobhistory
WHERE UPPER(master.dbo.fn_varbinto hexstr(job_id)) = UPPER('$(ESCAPE_SQUOTE(JOBID))')
AND msdb.dbo.agent_datetime(run_date, run_time) >= msdb.dbo.agent_datetime($(ESCAPE_SQUOTE(STRTD)), $(ESCAPE_SQUOTE(STRTTM)))
AND run_status <> 1
)
BEGIN
    RAISERROR('One or more job steps have failed.', 16, 1)
END
```

- See <http://ola.hallengren.com/frequently-asked-questions.html>



Professional Association for SQL Server

Scheduling Strategy

How should I schedule jobs?

- The answer depends on your maintenance window, the size of your databases, the maximum data loss you can tolerate, and many other factors. Here are some guidelines that you can start with, but you will need to adjust these to your environment.
- User databases:
 - Full backup one day per week
 - Differential backup all other days of the week
 - Transaction log backup every hour
 - Integrity check one day per week
 - Index maintenance one day per week
- System databases:
 - Full backup every day
 - Integrity check one day per week
- I recommend that you run a full backup after the index maintenance. The following differential backups will then be small. I also recommend that you perform the full backup after the integrity check. Then you know that the integrity of the backup is okay.
- Cleanup:
 - `sp_delete_backuphistory` one day per week
 - `sp_purge_jobhistory` one day per week
 - CommandLog cleanup one day per week
 - Output file cleanup one day per week



Professional Association for SQL Server

Customization

- Many ways to customize depends on your needs/environment
- I have many small (under 20 GB databases) so I decided to combine Backup, IndexOptimize and DBCC in one job as separate steps (with the error handling step at the end). I run the job daily for all user_databases
- I have similar job for system_databases (leaving out indexing)
- We have use Microsoft Data Protection Manager (DPM) to backup offsite. Restores take long time and are not straightforward. We have lots of space on our VM's for storage. I keep the backups local but only 2 days worth.
- I skipped the output file but am reconsidering
- My way works for me but it probably wouldn't for...
 - Large Databases with large indexes...
 - Insufficient space for backups
 - A small maintenance window (24/7 shop)
 - Etc...



Professional Association for SQL Server

Addendum

- Why Use BACKUP WITH CHECKSUM OPTION - Starting from SQL Server 2005, we have an option to enable **PAGE CHECKSUM** on the database level using the PAGE_VERIFY database option. Going over in detail about PAGE CHECKSUM is out of the scope [detailed blog post coming soon with additional tests and interesting results] for this post, but in essence it helps to uncover corruption caused by the IO subsystem. With regular T-SQL BACKUP, one can use the CHECKSUM clause with the BACKUP command and it helps in the following ways.
 - Validate the PAGE CHECKSUM in all of the data pages where CHECKSUM/TORN_PAGE_DETECTION value is available.
 - Even if the PAGE CHECKSUM values are NOT present, adding this clause helps to compute the combined CHECKSUM of all data pages which is stored on the backup media. This is valuable if you want to check if the backup itself is NOT corrupted prior to restoring from the backup.
 - While BACKUP CHECKSUM is NO replacement for the extensive DBCC CHECKDB to validate corruption issues, but has a place of its own and should be run in between the regular schedule of the DBCC CHECKDB as a best practice.
 - So far so good for the T-SQL but a mega fail on the **maintenance plan** as it doesn't have any option to use this feature. Or at-least I couldn't figure out yet if it is hidden somewhere.
 - Can then Validate backups using RESTORE VERIFYONLY FROM DISK = N'C:\BackupFolder\BackupFile.bak' With CHECKSUM
 - Use RESTORE VERIFYONLY if can't afford to do full DBCC CHECKDB nightly
- References
 - <http://msdn.microsoft.com/en-us/library/ms188902.aspx>
 - <http://www.sqlservercentral.com/Forums/Topic940767-1550-1.aspx>
 - <https://www.simple-talk.com/sql/database-administration/database-maintenance-plans-in-ssms-an-overview/>
 - <https://www.simple-talk.com/sql/database-administration/automate-and-improve-your-database-maintenance-using-ola-hallengrens-free-script/>



Professional Association for SQL Server



Professional Association for SQL Server

Thank You from PASS

www.sqlpass.org