Standardized MS SQL Maintenance

Documentation

Table of Contents

Official documentation of one-script maintenance solution

1 Introduction

This document describes maintenance solution used for keeping SQL Server in good condition purposes as agreed maintenance standard used on all SQL Server environments which is deployed after SQL Server instance installation. By default it is configured on agreed schedules but can be manually customized after deployment to the instance directly in SQL Agent jobs.

All environment specific need can be meet by modifications to standard implementation, for more info regarding to customization of Ola Halengreen's scripts can be found at official sites.

Maintenance can be deployed to SQL Server versions >= 2005 (SQL Server 2000 not supported but some solution is in long term plans to have one script maintenance solution as this one)

2 Technical preview

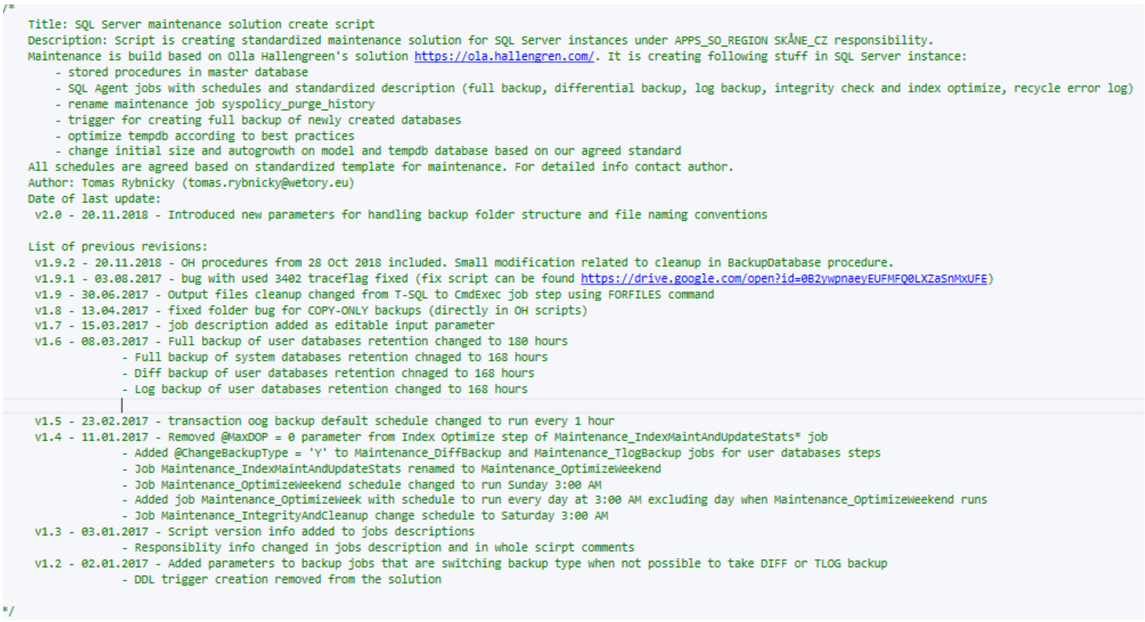
Whole solution is mostly based on popular Ola Hallengren’s maintenance procedures (visit here for more details https://ola.hallengren.com/). Olla’s procedures have been modified so as to fit needs described in standard specified in our team. Script is creating maintenance jobs as its main objection. All tasks performed on the target SQL Server instance are listed below.

Script perform following stuff:

- create OH procedures in master database
- create and drop helper procedures in master database
- create SQL Agent jobs (full backup, differential backup, log backup, integrity check and index optimize, recycle error log)
- schedule SQL Agent jobs based on agreed standard times
- rename maintenance job syspolicy_purge_history
- optimize tempdb according to best practices (this is optional)
- change initial size and Autogrowth on model and tempdb database based on our agreed standard

Some of above described actions are not executed automatically but can be fired by changing input parameters, for more info see section Deployment

All important info also included in script header, containing also some short update history.

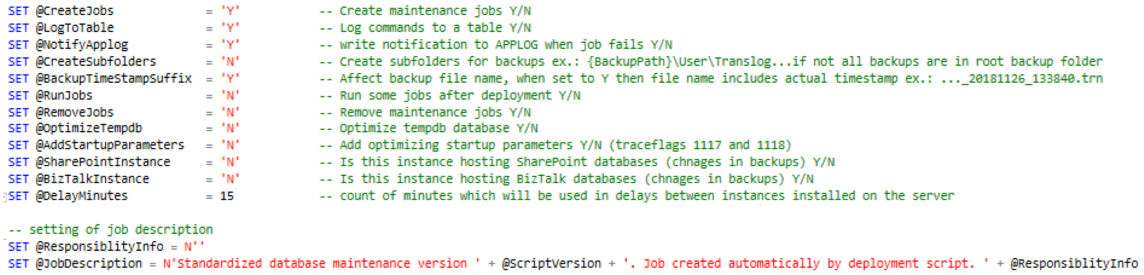


```
/*
   Title: SQL Server maintenance solution create script
   Description: Script is creating standardized maintenance solution for SQL Server instances under APPS_SO_REGION SKÅNE_CZ responsibility.
   Maintenance is build based on Olla Hallengreen's solution https://ola.hallengren.com/. It is creating following stuff in SQL Server instance:
       - stored procedures in master database
       - SQL Agent jobs with schedules and standardized description (full backup, differential backup, log backup, integrity check and index optimize, recycle error log)
       - rename maintenance job syspolicy_purge_history
       - trigger for creating full backup of newly created databases
       - optimize tempdb according to best practices
       - change initial size and autogrowth on model and tempdb database based on our agreed standard
   All schedules are agreed based on standardized template for maintenance. For detailed info contact author.
   Author: Tomas Rybnicky (tomas.rybnicky@wetory.eu)
   Date of last update:
    v2.0 - 20.11.2018 - Introduced new parameters for handling backup folder structure and file naming conventions

   List of previous revisions:
    v1.9.2 - 20.11.2018 - OH procedures from 28 Oct 2018 included. Small modification related to cleanup in BackupDatabase procedure.
    v1.9.1 - 03.08.2017 - bug with used 3402 traceflag fixed (fix script can be found https://drive.google.com/open?id=0B2ywpnaeyEUFMFQ0LXZaSnMxUFE)
    v1.9 - 30.06.2017 - Output files cleanup changed from T-SQL to CmdExec job step using FORFILES command
    v1.8 - 13.04.2017 - fixed folder bug for COPY-ONLY backups (directly in OH scripts)
    v1.7 - 15.03.2017 - job description added as editable input parameter
    v1.6 - 08.03.2017 - Full backup of user databases retention changed to 180 hours
                 - Full backup of system databases retention chnaged to 168 hours
                 - Diff backup of user databases retention chnaged to 168 hours
                 - Log backup of user databases retention changed to 168 hours

    v1.5 - 23.02.2017 - transaction oog backup default schedule changed to run every 1 hour
    v1.4 - 11.01.2017 - Removed @MaxDOP = 0 parameter from Index Optimize step of Maintenance_IndexMaintAndUpdateStats* job
                 - Added @ChangeBackupType = 'Y' to Maintenance_DiffBackup and Maintenance_TlogBackup jobs for user databases steps
                 - Job Maintenance_IndexMaintAndUpdateStats renamed to Maintenance_OptimizeWeekend
                 - Job Maintenance_OptimizeWeekend schedule changed to run Sunday 3:00 AM
                 - Added job Maintenance_OptimizeWeek with schedule to run every day at 3:00 AM excluding day when Maintenance_OptimizeWeekend runs
                 - Job Maintenance_IntegrityAndCleanup change schedule to Saturday 3:00 AM
    v1.3 - 03.01.2017 - Script version info added to jobs descriptions
                 - Responsiblity info changed in jobs description and in whole scirpt comments
    v1.2 - 02.01.2017 - Added parameters to backup jobs that are switching backup type when not possible to take DIFF or TLOG backup
                 - DDL trigger creation removed from the solution

*/
```

Figure 1 - script header

3 Deployment

Only thing you have to do is to copy script from its official storage on Google Drive. Copy script to SQL Server Management Studio and set parameters if needed, by default parameters are set as following (all of them are pretty self-commented). You can change them based on environment you are deploying to.



```
SET @CreateJobs              = 'Y'          -- Create maintenance jobs Y/N
SET @LogToTable              = 'Y'          -- Log commands to a table Y/N
SET @NotifyApplog            = 'Y'          -- write notification to APPLOG when job fails Y/N
SET @CreateSubfolders        = 'N'          -- Create subfolders for backups ex.: {BackupPath}\User\Translog...if not all backups are in root backup folder
SET @BackupTimeStampSuffix   = 'Y'          -- Affect backup file name, when set to Y then file name includes actual timestamp ex.: ..._20181126_133840.trn
SET @RunJobs                 = 'N'          -- Run some jobs after deployment Y/N
SET @RemoveJobs              = 'N'          -- Remove maintenance jobs Y/N
SET @OptimizeTempdb          = 'N'          -- Optimize tempdb database Y/N
SET @AddStartupParameters    = 'N'          -- Add optimizing startup parameters Y/N (traceflags 1117 and 1118)
SET @SharePointInstance      = 'N'          -- Is this instance hosting SharePoint databases (chnages in backups) Y/N
SET @BizTalkInstance         = 'N'          -- Is this instance hosting BizTalk databases (chnages in backups) Y/N
SET @DelayMinutes            = 15           -- count of minutes which will be used in delays between instances installed on the server

-- setting of job description
SET @ResponsiblityInfo = N''
SET @JobDescription = N'Standardized database maintenance version ' + @ScriptVersion + '. Job created automatically by deployment script. ' + @ResponsiblityInfo
```

Figure 2 Script parameters

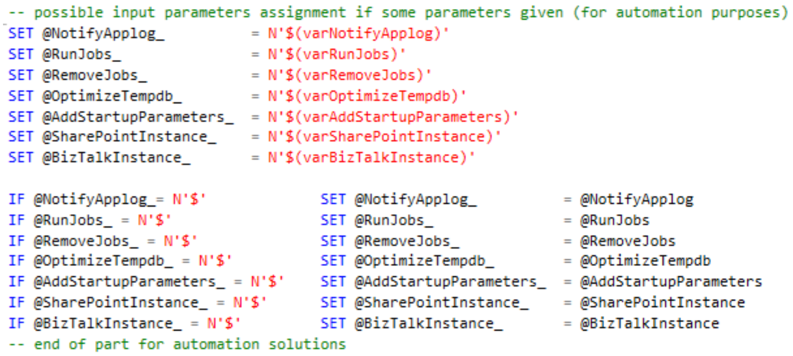
You can also change script body if there is something that you do not like there, but only by your own responsibility!

You can deploy to single server or to more servers using registered servers, where script will be executed in parallel on all registered instances (never mind if running on one or more servers).

One important thing is that script is creating delayed schedules for more instance running on same machine. Because of reducing system resources load during run at same time on all instances. You can set delay by parameter **@DelayMinutes**. Then every installed instance is given index based on value in registry key **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server** under value of key name **IntalledInstances.**

3.1 Running from another script

From V1.3 there is support for parametrizing run of the script for executing it from another script (batch, PowerShell…). There need to be possibility to set mentioned parameters for execution from “outside”. Any of the parameters are required for running the script, if no value for parameter given default values from declaration will be used. See below screen for more info about possible parameters



```
-- possible input parameters assignment if some parameters given (for automation purposes)
SET @NotifyApplog_          = N'$(varNotifyApplog)'
SET @RunJobs_               = N'$(varRunJobs)'
SET @RemoveJobs_            = N'$(varRemoveJobs)'
SET @OptimizeTempdb_        = N'$(varOptimizeTempdb)'
SET @AddStartupParameters_  = N'$(varAddStartupParameters)'
SET @SharePointInstance_    = N'$(varSharePointInstance)'
SET @BizTalkInstance_       = N'$(varBizTalkInstance)'

IF @NotifyApplog_= N'$'               SET @NotifyApplog_          = @NotifyApplog
IF @RunJobs_ = N'$'                   SET @RunJobs_               = @RunJobs
IF @RemoveJobs_ = N'$'                SET @RemoveJobs_            = @RemoveJobs
IF @OptimizeTempdb_ = N'$'            SET @OptimizeTempdb_        = @OptimizeTempdb
IF @AddStartupParameters_ = N'$'      SET @AddStartupParameters_  = @AddStartupParameters
IF @SharePointInstance_ = N'$'        SET @SharePointInstance_    = @SharePointInstance
IF @BizTalkInstance_ = N'$'           SET @BizTalkInstance_       = @BizTalkInstance
-- end of part for automation solutions
```

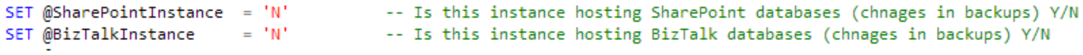
Figure 3 Parameters for "outside" execution

3.2 Exceptions in environments

Parameters: @SharePointInstance, @BizTalkInstance

Some application specific SQL environments requires some exception in SQL maintenance, for example BizTalk related SQL databases are handled by backups from BizTalk application. Or SharePoint databases can be handled by 3rd party tool like DocAve. There are prepared switch parameters at the script beginning to modify maintenance accordingly.

Now there are only two exceptions for mentioned BizTalk and SharePoint environments as described below.



```
SET @SharePointInstance  = 'N'        -- Is this instance hosting SharePoint databases (chnages in backups) Y/N
SET @BizTalkInstance     = 'N'        -- Is this instance hosting BizTalk databases (chnages in backups) Y/N
```

Figure 4 Environment exception flags

If instance is marked as **@SharePointInstance** then only system databases are handled by this maintenance solution. User databases are handled by DocAve in this situation.

If instance is marked as **@BizTalkInstance** then exceptions for standard BizTalk databases are included in backup step of user databases. Then parameter in script is set as following

@Databases = 'USER_DATABASES, -BAMPrimaryImport%, -BAMAlertsApplication%, -BAMAlertsNSMain%, -BizTalkDTADb%, -BizTalkMgmtDb%, -BizTalkMsgBoxDb%, -BizTalkRuleEngineDb%, -SSODB%, -BAMStarSchema%, -BAMAnalysis%, -BAMArchive%, -BizTalkAnalysisDb%'

This exclusion is done based on official BizTalk documentation https://msdn.microsoft.com/en-us/library/aa578342.aspx

3.3 Optimize TempDB

Parameters: @OptimizeTempdb

There are some optimization recommendations regarding TempDB performance. Based on MS best practices following TempDB configurations should be followed for optimal performance of TempDB database:

- database files should be stored on separate disk drives to provide exclusive IO operations just for communication with this database on disk drives (this is not object of maintenance solution and should be done by design of environment)
- database files should be divided into number of files equally to count of logical processors but maximally up to 8, then advantages of parallel accessing become disadvantage because of internal management of SQL engine
- database auto growth and initial size should be the same on all files mentioned in previous step. We have chosen following values for our standard. Initial size is set to **512 MB** and file growth is set to **256 MB** which are agreed values not causing any overload when performing auto growth of files and similar issues

.

More info about TempDB optimization can be found on official MS web pages https://msdn.microsoft.com/en-us/library/ms190768.aspx.

3.4 Optimize model settings

The **model** database is used as the template for all databases created on an instance of SQL Server. Because **tempdb** is created every time SQL Server is started, the **model** database must always exist on a SQL Server system. The entire contents of the **model** database, including database options, are copied to the new database. Some of the settings of **model** are also used for creating a new **tempdb** during start up, so the **model** database must always exist on a SQL Server system.

We are changing model database configuration to more efficient and optimal values in our standardized solution. We are changing initial file size and file growth attributes of this database from default values to agreed values providing some performance improvements on newly created databases with default configuration in comparison to default values set after installation of SQL Server instance. Values change described in table below.

Attribute	**Default value**	**Changed value**
Autogrowth for data file	By 1 MB, Unlimited	By 128 MB, Unlimited
Autogrowth for log file	By 10 percent, Unlimited	By 64 MB, Unlimited
Initial size for data file	5 MB	128 MB
Initial size for log file	2 MB	64 MB

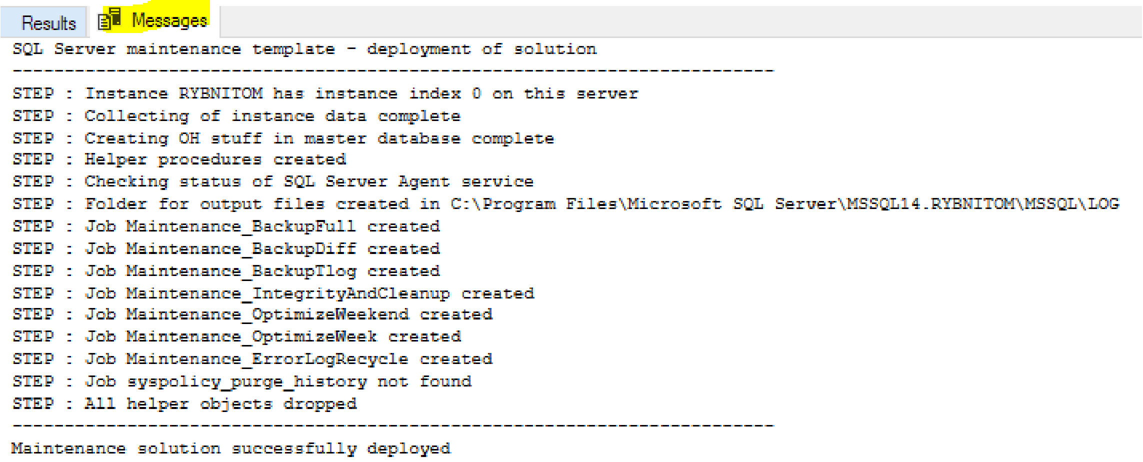
Table model configuration

For more info about model database see official MSDN article https://msdn.microsoft.com/en-us/library/ms186388.aspx.

4 After deployment

4.1 Direct results

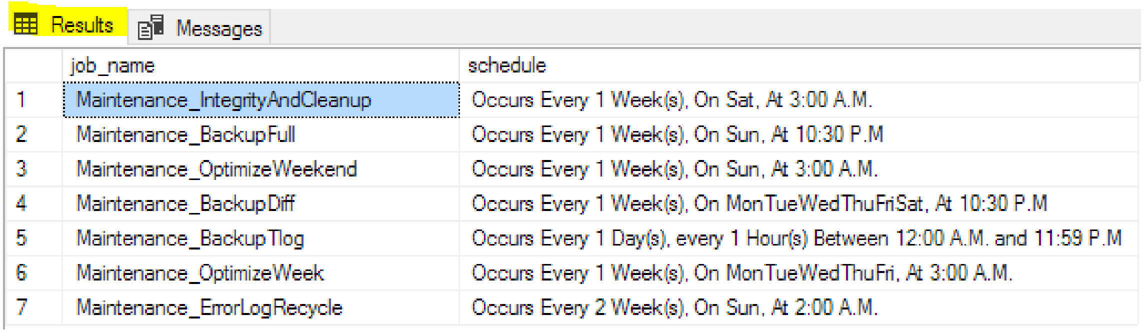
After proper execution you can check messages for detailed steps which have been done over instance and also for possible related error messages.



```
SQL Server maintenance template - deployment of solution
-----------------------------------------------------------------------
STEP : Instance RYBNITOM has instance index 0 on this server
STEP : Collecting of instance data complete
STEP : Creating OH stuff in master database complete
STEP : Helper procedures created
STEP : Checking status of SQL Server Agent service
STEP : Folder for output files created in C:\Program Files\Microsoft SQL Server\MSSQL14.RYBNITOM\MSSQL\LOG
STEP : Job Maintenance_BackupFull created
STEP : Job Maintenance_BackupDiff created
STEP : Job Maintenance_BackupTlog created
STEP : Job Maintenance_IntegrityAndCleanup created
STEP : Job Maintenance_OptimizeWeekend created
STEP : Job Maintenance_OptimizeWeek created
STEP : Job Maintenance_ErrorLogRecycle created
STEP : Job syspolicy_purge_history not found
STEP : All helper objects dropped
-----------------------------------------------------------------------
Maintenance solution successfully deployed
```

Figure 5 Script messages

And in some cases you may be asked for schedules or you can use it for self-review of schedules for deployed jobs or you can attach to install change etc. there is result table with job names and schedules.

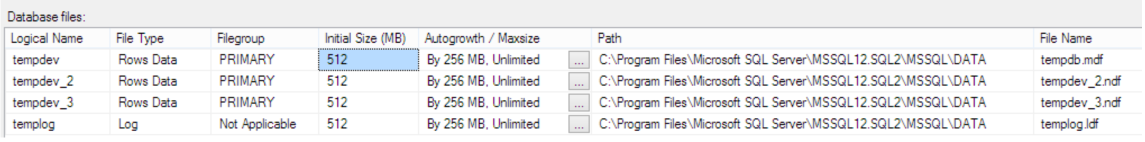


	job_name	schedule
1	Maintenance_IntegrityAndCleanup	Occurs Every 1 Week(s), On Sat, At 3:00 A.M.
2	Maintenance_BackupFull	Occurs Every 1 Week(s), On Sun, At 10:30 P.M
3	Maintenance_OptimizeWeekend	Occurs Every 1 Week(s), On Sun, At 3:00 A.M.
4	Maintenance_BackupDiff	Occurs Every 1 Week(s), On MonTueWedThuFriSat, At 10:30 P.M
5	Maintenance_BackupTlog	Occurs Every 1 Day(s), every 1 Hour(s) Between 12:00 A.M. and 11:59 P.M
6	Maintenance_OptimizeWeek	Occurs Every 1 Week(s), On MonTueWedThuFri, At 3:00 A.M.
7	Maintenance_ErrorLogRecycle	Occurs Every 2 Week(s), On Sun, At 2:00 A.M.

Figure 6 Script results

4.2 System databases

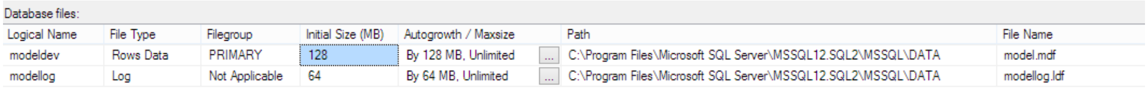
You can also can see tempdb and model database file properties that it was changed if parameter **@OptimizeTempdb** is set to Y (what means YES).



Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize		Path	File Name
tempdev	Rows Data	PRIMARY	512	By 256 MB, Unlimited	...	C:\Program Files\Microsoft SQL Server\MSSQL12.SQL2\MSSQL\DATA	tempdb.mdf
tempdev_2	Rows Data	PRIMARY	512	By 256 MB, Unlimited	...	C:\Program Files\Microsoft SQL Server\MSSQL12.SQL2\MSSQL\DATA	tempdev_2.ndf
tempdev_3	Rows Data	PRIMARY	512	By 256 MB, Unlimited	...	C:\Program Files\Microsoft SQL Server\MSSQL12.SQL2\MSSQL\DATA	tempdev_3.ndf
templog	Log	Not Applicable	512	By 256 MB, Unlimited	...	C:\Program Files\Microsoft SQL Server\MSSQL12.SQL2\MSSQL\DATA	templog.ldf

Figure 7 TempDB files optimization (3 CPUs)



Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize		Path	File Name
modeldev	Rows Data	PRIMARY	128	By 128 MB, Unlimited	...	C:\Program Files\Microsoft SQL Server\MSSQL12.SQL2\MSSQL\DATA	model.mdf
modellog	Log	Not Applicable	64	By 64 MB, Unlimited	...	C:\Program Files\Microsoft SQL Server\MSSQL12.SQL2\MSSQL\DATA	modellog.ldf

Figure 8 Model DB optimization

4.3 SQL Agent Jobs

This section describes SQL Agent jobs deployed to the target SQL Server instance by executing mentioned script. All jobs are part of SQL Server maintenance since deployed and all colising Maintenance plans or user defined SQL AGen jobs should be disabled or removed if there are any to prevent duplicate maintenance or backup tasks done.

Following list contain short description for all steps in all jobs that are part of solution.For more info about procedures executed in steps, read offical Ola Halengreen’s documentation.

Every step also contains link to official Microsoft documentation of SQL Server maintenance tasks and related command and procedures used to run them.

Maintenance_BackupDiff

- **TRACEON 3042** – enabling incremental backup file growth
  https://msdn.microsoft.com/en-us/library/ms188396.aspx
- **Full Backup - System databases** – full backup of all system databases
  https://msdn.microsoft.com/en-us/library/ms186289.aspx
- **Diff Backup - User databases –** differential backup of all user databases
  https://msdn.microsoft.com/en-us/library/ms175526.aspx
- **TRACEOFF 3042 –** disabling incremental backup file growth
  https://msdn.microsoft.com/en-us/library/ms188396.aspx
- **Integrity Physical - All databases –** DBCC CHECKDB with option PHYSICAL_ONLY
  https://msdn.microsoft.com/en-us/library/ms176064.aspx

Maintenance_BackupFull

- **TRACEON 3042** – enabling incremental backup file growth
  https://msdn.microsoft.com/en-us/library/ms188396.aspx
- **Full Backup - System databases**– full backup of all system databases
  https://msdn.microsoft.com/en-us/library/ms186289.aspx

- **Full Backup - User databases**– full backup of all user databases https://msdn.microsoft.com/en-us/library/ms186289.aspx
- **TRACEOFF 3042–** disabling incremental backup file growth https://msdn.microsoft.com/en-us/library/ms188396.aspx

Maintenance_BackupTlog

- **TRACEON 3042**– enabling incremental backup file growth https://msdn.microsoft.com/en-us/library/ms188396.aspx
- **Tlog Backup - System databases –** backup of transaction log for all system databases (in full recovery model) https://msdn.microsoft.com/en-us/library/ms191429.aspx
- **Tlog Backup - User databases –** backup of transaction log for all user databases (in full recovery model) https://msdn.microsoft.com/en-us/library/ms191429.aspx
- **TRACEOFF 3042–** disabling incremental backup file growth https://msdn.microsoft.com/en-us/library/ms188396.aspx

Maintenance_ErrorLogRecycle

- **Cycle Error Log –** close existing SQL Log and create new one https://technet.microsoft.com/en-us/library/ms182512(v=sql.110).aspx

Maintenance_IntegrityAndCleanup

- **Integrity Check –** DBCC CHECKDB with full scans over all objects in databases https://msdn.microsoft.com/en-us/library/ms176064.aspx
- **Command Log Cleanup –** removing records older than 30 days from [master].[dbo].[CommandLog] table
- **Output Files Cleanup -** removing files older than 30 days from output files folder (<<log directory>> + \Maintenace_OutputFiles\)
- **History Cleanup -** removing records older than 30 days from system tables

Maintenance_OptimizeWeek

- **Index Optimize –** rebuilding or reorganizing indexes based on their fragmentation (0% - 30%, 30% - 60%, 60% - 100%). Excluding offline index rebuilds to prevent table locks. https://technet.microsoft.com/en-us/library/ms190910(v=sql.105).aspx
- **Update Statistics –** updating of query optimization statistics on a table or indexed view https://msdn.microsoft.com/en-us/library/ms187348.aspx

Maintenance_OptimizeWeekend

- **Index Optimize –** rebuilding or reorganizing indexes based on their fragmentation (0% - 5%, 5% - 30%, 30% - 100%). Including offline index rebuilds. https://technet.microsoft.com/en-us/library/ms190910(v=sql.105).aspx
- **Update Statistics –** updating of query optimization statistics on a table or indexed view https://msdn.microsoft.com/en-us/library/ms187348.aspx

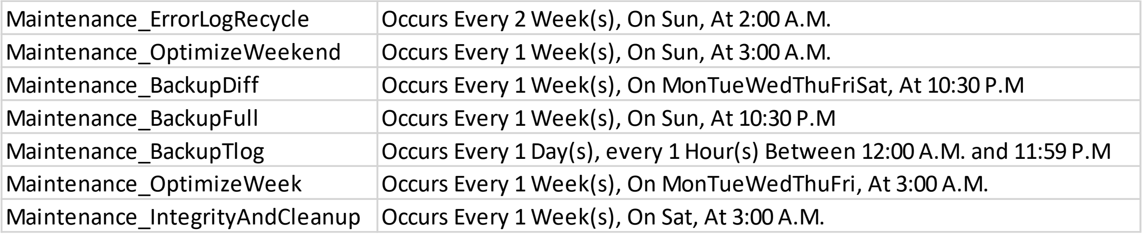
Maintenance_SyspolicyPurgeHistory (just renamed syspolicy_purge_history job)

- **Verify that automation is enabled.**
- **Purge history.**
- **Erase Phantom System Health Records.**

All suggested SQL Server database maintenance tasks are pretty well described at this site https://technet.microsoft.com/en-us/library/ms140255(v=sql.105).aspx. This maintenance solution is trying to find some accurate way to meet all database needs and include standardization definitions and schedules for use in company which is taking care of many SQL Server environments and want to keed all instances in good health condition.

4.4 SQL Agent Jobs schedules

All schedules are set up during deployment as agreed standardized and optimal schedules for related SQL Server maintenance task. If SQL Server instance require different schedules because of some reason it can be modified after deployement same as whole maintenance solution **can be (and also should be) modified** to fit exact needs of SQL Server instance it is deployed to.



Maintenance_ErrorLogRecycle	Occurs Every 2 Week(s), On Sun, At 2:00 A.M.
Maintenance_OptimizeWeekend	Occurs Every 1 Week(s), On Sun, At 3:00 A.M.
Maintenance_BackupDiff	Occurs Every 1 Week(s), On MonTueWedThuFriSat, At 10:30 P.M
Maintenance_BackupFull	Occurs Every 1 Week(s), On Sun, At 10:30 P.M
Maintenance_BackupTlog	Occurs Every 1 Day(s), every 1 Hour(s) Between 12:00 A.M. and 11:59 P.M
Maintenance_OptimizeWeek	Occurs Every 1 Week(s), On MonTueWedThuFri, At 3:00 A.M.
Maintenance_IntegrityAndCleanup	Occurs Every 1 Week(s), On Sat, At 3:00 A.M.

Figure 9 - SQL Agent jobs schedules

4.5 Database backup retention settings

Following drawing is describing available retore times that are feasible when standardized SQL Server maintenance is used for taking care of database backup reoutines.

Drawing does not describe point-in-time restores for databases in full recevory model. Backup of transaction log is running every 1 hour by default (can be adjusted for frequently changed databases)

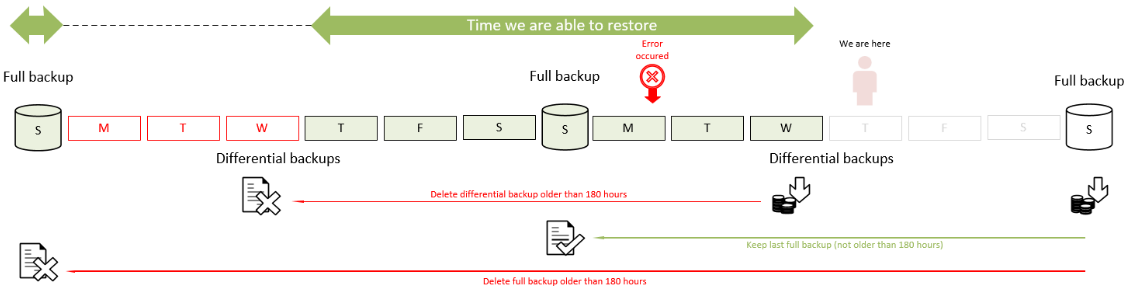

Time we are able to restore
Error occured
We are here
Full backup
Full backup
Full backup
S
M
T
W
T
F
S
S
M
T
W
T
F
S
S
Differential backups
Differential backups
Delete differential backup older than 180 hours
Keep last full backup (not older than 180 hours)
Delete full backup older than 180 hours

Figure 11 - backup retention settings

5 Possible problems

There was testing of the solution ongoing for several weeks for debugging and tuning purposes and all known problems has been fixed already, but as everything also this script can cause some issues in different environments.

I’m assuming only following possible issues:

- problems with old backup files removal
- problems with compression on not supported SQL versions/editions (partially fixed)

And some other possible problems can be related to OH stuff in the solution so, please be so kind and try to check this FAQ https://ola.hallengren.com/frequently-asked-questions.html first before asking me directly.

5.1 Reporting issues

Please report all found issues, current version of the solution is the first one and require some debugging to be “perfect”. Here are some contacts you can write to via email or LYNC:

- tomas.rybnicky@wetory.eu (T-SQL)

6 Change history

Version	Date	Author	Change history
v1.1	06.04.2016	Tomáš Rybnický	First version of this document
v1.2	11.05.2016	Tomáš Rybnický	Documentation updated
v1.3	19.05.2016	Tomáš Rybnický	Automation support
v1.4	03.01.2017	Tomáš Rybnický	Scope of document changed
v1.5	11.01.2017	Tomáš Rybnický	Major updates optimization of SQL Server
v1.6	14.03.2017	Tomáš Rybnický	Backup retention settings
v1.7	15.03.2017	Tomáš Rybnický	Job description as editable parameter
v2.0	26.11.2018	Tomáš Rybnický	Added parameters for folder structure and file naming