dbaTDPMon

TROUBLESHOOT DATABASE PERFORMANCE & MONITORING

Version 2015.9

This utility is a bespoke database / system maintenance and health-check solution for SQL Server.
It runs for versions from SQL Server 2000 until 2012. It was not tested on SQL Server 2014 and 2016, but no issues should be found. Some features are only available for SQL Server 2005 and beyond.
Utility consists in plain T-SQL code. A database is needed in order to store objects used by this utility.
Task automation is performed using SQL Agent jobs, pre-scheduled.
Both maintenance-plan and health-check modules can be used as an "agentless" management system.

Why dbaTDPMon?

- implement database maintenance best practices (including system databases)
- automate daily health checks / HTML reporting
- can be used to administrate multiple instances from a central point
- fully customizable / various options / time limit / email alerting
- check full documentation for all details

Custom Maintenance Plan
- ***Backup***
  - use checksum (+2k5) and verify the backup file
  - retention can be set to days or backup files count
  - automatically trigger a full database backup prior to taking a transaction log / differential backup, if needed
- ***Consistency Checks***
  - can be run at database or table level
  - checks are spread over an entire week (configurable)
- ***Index Maintenance***
  - reorganize/rebuild decision can be based on logical fragmentation or page density
  - use "drive table" to limit the number of analyzed indexes
  - 2 algorithms available: online/offline index rebuild or disable/rebuild (managing dependencies)
  - may force ghost records cleanup
- ***Heap Tables Maintenance***
  - rebuild decision based on extent fragmentation, page density and forwarded records
- ***Statistics Maintenance***
  - use "drive table" to limit the number of analyzed statistics
  - update decision is made based on statistics age and changes made
- ***System Maintenance***
  - scheduled errorlog cycle
  - purge history

Daily Health Checks
- online/offline instances and databases health state
- report failed SQL Agent jobs / disk space issues
- report outdated backups and checkdb
- analyze errorlogs
- and many more...

Table of Contents

a. maintenance-plan

1. Database Backup (2k / 2k5 / 2k8 / 2k8r2 / 2k12)
- file name template: `ServerName_DBName_yyyymmdd_hhmmss_BackupType.Ext`
  - o BackupType: FULL, DIFF, LOG
  - o file extensions: BAK / TRN
- weekly system databases full backup
- daily user databases differential backup / weekly user databases full backup / hourly log backup
- checksum, compression (default, if available)
- if log/diff backup and no full, a full is taken
- backups are verified (with checksum if available, 2k5+)
- backup location set in `dbo.appConfiguration` (one per instance)
  - o if not set, can be sent as a stored procedure parameter value
- each database on its own folder
- default retention 7 days (set up in `dbo.appConfiguration`)
  - o may change to last 7 backups (at least one full included)
  - o cleanup is performed using 2 methods:
    - ▪ xp_delete_file;
    - ▪ del file (list taken from msdb) / enable/disable xp_cmdshell
- when performing cleanup, if a full backup is deleted, orphan diff and log backups are also deleted

SQL Server Agent Jobs:
- dbaTDPMon - Database Backup – Full
  - o daily at 00:00:00, system database on Saturday only
  - o job sends full execution log over email (+2k5)
    - ▪ backupset information included in email
- dbaTDPMon - Database Backup – Log
  - o daily, every hour

2. User Database Maintenance (2k / 2k5 / 2k8 / 2k8r2 / 2k12)
- Daily: Kill Orphan Connections (+2k5)
- Daily: Allocation Consistency Checks
  - ▪ DBCC CHECKALLOC / DBCC CHECKCATALOG
  - ▪ *daily, except Saturday*
- Weekly: Tables Consistency Checks
  - ▪ DBCC CHECKTABLE / DBCC CHECKIDENT
  - ▪ *run with DATA_PURITY (if applicable), EXTENDED_LOGICAL_CHECKS*
  - ▪ *user and system tables*
  - ▪ *on Sunday / only objects with pages allocated*
- Weekly: Reference Consistency Checks
  - ▪ DBCC CHECKCONSTRAINTS
  - ▪ *on Sunday*
- Weekly: Database Consistency Check
  - ▪ DBCC CHECKDB WITH PHYSICAL_ONLY
  - ▪ *on Saturday*
- Weekly: Perform Correction to Space Usage
  - ▪ DBCC UPDATEUSAGE
  - ▪ *on Monday / only objects with pages allocated*
- Daily: Rebuild or Reorganize Indexes
  - ▪ ALTER INDEX REORGANIZE – logical fragmentation in [5,30] +1000 pages
  - ▪ ALTER INDEX REBUILD – logical fragmentation in [30, 100] and +1000 pages
  - ▪ use online (if applicable); (additional disk space may be required to perform the operation)
    - • additional "space efficient" algorithm available: disable index (and its dependencies – non-clustered/XML indexes, foreign-keys), rebuild index and rebuild disabled dependencies

- analyzed index types are: clustered, non-clustered, XML (primary/secondary), spatial
- may force ghost cleanup process, if option is enabled (ghost records for indexes which got rebuilt are excluded) (*see Ghost Records clean-up algorithm*)
- rebuild/reorganize decision is based on logical fragmentation (page density analysis is also available, but not a default option) (*see Index Reorganize / Rebuild algorithm*)
- Daily: Rebuild Heap Tables (+2k5)
  - ALTER TABLE REBUILD (+2k8)
    - heap tables with disabled unique indexes will be excluded: rebuild table also means index rebuild, and unique indexes may enable unwanted constraints
  - alternative algorithm: disable all indexes, create PK as GUID, drop PK, rebuild all indexes
  - rebuild decision is based on extent fragmentation, page density and forwarded records percentage (*see Heap Rebuild algorithm*)
- Daily: Update Statistics
  - update statistics where age is older than 7 days (default) or where age is newer than 7 days (default) but percent of changes made is higher than 1 (default)
  - statistics with no changes made are not updated regardless of age (*see Statistics update algorithm*)
- Weekly: Shrink Database (TRUNCATEONLY)
  - DBCC SHRINKDATABASE WITH TRUNCATEONLY
  - *on Monday*
- Monthly: Shrink Log File
  - DBCC SHRINKFILE WITH TRUNCATEONLY
  - *first Saturday of the month*

SQL Server Agent Jobs:
- dbaTDPMon - Database Maintenance - Users DBs
  - *daily at 02:00:00*
  - job sends execution log over email (+2k5)

3. System & System Database Maintenance (2k / 2k5 / 2k8 / 2k8r2 / 2k12)
- master - Cycle errorlog file (monthly)
  - *on the 1st of each month*, run `master.dbo.sp_cycle_errorlog`
- master - Consistency Checks (weekly)
  - DBCC CHECKDB
  - *on Saturday*
- msdb - Consistency Checks (weekly)
  - DBCC CHECKDB
  - *on Saturday*
- model - Consistency Checks (weekly)
  - DBCC CHECKDB
  - *on Saturday*
- tempdb - Consistency Checks (weekly)
  - DBCC CHECKDB
  - *on Saturday*
- msdb - Backup History Retention – keep only the last 6 months
  - daily, `msdb.dbo.sp_delete_backuphistory`
- msdb - Job History Retention – keep only the last 12 months
  - daily, `msdb.dbo.sp_purge_jobhistory`
- msdb - Maintenance Plan History Retention – keep only the last 6 months
  - daily, `msdb.dbo.sp_maintplan_delete_log`
- msdb - Purge Old Mail Items (+2k5) – keep only the last 6 months
  - daily, `msdb.dbo.sysmail_delete_mailitems_sp`
- msdb - Purge Old Mail Logs (+2k5) – keep only the last 6 months
  - daily, `msdb.dbo.sysmail_delete_log_sp`
- msdb - Replication Alerts Retention (+2k5) – keep only the last 6 months
  - daily, `msdb.dbo.sysreplicationalerts`

- master – Index & Statistics Maintenance (weekly)
  - ALTER INDEX REORGANIZE [5,30) / +1000 pages
  - ALTER INDEX REBUILD (+30 and +1000pages) or [5,30) and -1000 pages
  - statistics update (7 days age plus 1% data changes)
  - *on Sunday*
- msdb - Index & Statistics Maintenance (weekly)
  - ALTER INDEX REORGANIZE [5,30) / +1000 pages
  - ALTER INDEX REBUILD (+30 and +1000pages) or [5,30) and -1000 pages
  - statistics update (7 days age plus 1% data changes)
  - *on Sunday*

SQL Server Agent Jobs:
- dbaTDPMon - Database Maintenance - System DBs
  - *daily at 00:00:00*
  - job sends execution log over email (+2k5)

4. Configuration options

Table `dbo.appConfigurations` contains parameters that can be used for configuration:

name	value	description
Default project code	NULL	Project code to be used when not specified as a parameter, useful when receiving emails for multiple customers
Alert repeat interval (minutes)	60	Spam filter; interval for resending an alert when raised
Default lock timeout (ms)	5000	How long a SQL statement will wait for needed locks before timing out
Default backup location	NULL	Full disk path on where to place backup files (can be a UNC path) Filled in by default at installation with the current instance's default backup directory, if set.
Default backup retention (days)	7	
Database Mail profile name to use for sending emails	NULL	Database mail profile to be used for email notifications Filled in by default at installation with the first found database mail profile.
Default recipients list - Job Status (semicolon separated)	NULL	List of email addresses to which the generated job status will be sent as attachment
Default recipients list - Alerts (semicolon separated)	NULL	List of email addresses to which the generated alerts will be sent
Notify job status only for failed jobs	TRUE	Email will be sent only when a job fails, with job log and error details
Log action events	TRUE	Log all internal actions to `dbo.logEventMessages` table
Log events retention (days)	15	Internal event messages retention period
Ignore alerts for: Error 1222 - Lock request time out period exceeded	TRUE	Do not generate alerts when timeout errors are encountered
Change retention policy from RetentionDays to RetentionBackupsCount	TRUE	Change backup retention from days to backup count. At least one full backup will be kept to allow differential backups restore.
Force cleanup of ghost records	false	If ghost records count exceeds the threshold, will force the cleanup mechanism by running `sp_clean_db_free_space`
Ghost records cleanup threshold	131072	Threshold for running the ghost records cleanup

Default behavior:
- emails are sent only in case of failures
- all actions (changes to objects, job executions, index fragmentation or backupset information) are logged as events in `dbo.logEventMessages`

5. Upper Level Stored Procedures

5.1 Database Consistency Checks

The stored procedure below manages the way database consistency checks are performed:

```
[dbo].[usp_mpDatabaseConsistencyCheck]    @sqlServerName      [sysname] = @@SERVERNAME,
                                          @dbName             [sysname],
                                          @tableSchema        [sysname]      = '%',
                                          @tableName          [sysname]      = '%',
                                          @flgActions         [smallint]     =  12,
                                          @flgOptions         [int]          =   3,
                                          @executionLevel     [tinyint]      =   0,
                                          @debugMode          [bit]          =   0
```

Parameter Name	Description
@sqlServerName	type = sysname default value = @@SERVERNAME The instance name on which the database consistency checks will be performed. Can be either a local instance or a linked server.
@dbName	type = sysname The database name for which consistency checks will be performed. It must be a valid database name. Wildcards or database lists are not supported.
@tableSchema	type = sysname default value = % Object schema to be analyzed. This parameter is only used for DBCC CHECKTABLE, CHECKCONSTRAINTS, CHECKIDENT, UPDATEUSAGE and DBCC CLEANTABLE. If this parameter is specified, only schemas with name matching it will be analyzed. Wildcards are supported. Lists are not supported.
@tableName	type = sysname default value = % Object name to be analyzed. This parameter is only used for DBCC CHECKTABLE, CHECKCONSTRAINTS, CHECKIDENT, UPDATEUSAGE and DBCC CLEANTABLE. Only objects (tables and materialized views) with name matching the parameter value and with reserved pages will be analyzed. Wildcards are supported. Lists are not supported.

@flgActions	type = smallint
	default value = 12 (8 | 4)
	
	Select which database consistency checks should be performed.
	For multiple actions, parameter value is the sum of selected actions (each value represents a bit):
	   1 - perform database consistency check (DBCC CHECKDB)
	       o for versions greater or equal with than 2005 and when option 1 is not enabled DATA_PURITY option is used only when dbi_dbccFlags <> 2 (DBCC DBINFO)
	       o for versions greater or equal with than 2008R2 and when option 1 is not enabled EXTENDED_LOGICAL_CHECKS is always used
	   2 - perform table consistency check (DBCC CHECKTABLE)
	       o for versions greater or equal with than 2005, DATA_PURITY option is always used
	       o for versions greater or equal with than 2008R2 and when option 2 is not enabled EXTENDED_LOGICAL_CHECKS is always used
	   4 - perform consistency check of disk space allocation structures (DBCC CHECKALLOC)
	       o (default)
	   8 - perform consistency check of catalogs (DBCC CHECKCATALOG)
	       o (default)
	  16 - perform consistency check of table constraints (DBCC CHECKCONSTRAINTS)
	  32 - perform consistency check of table identity value (DBCC CHECKIDENT)
	       o additional filter is applied: only tables with columns having identity set will be analysed
	       o If database status is not ONLINE and READ_WRITE it will be skipped
	  64 - perform correction to space usage (DBCC UPDATEUSAGE)
	 128 - cleaning wasted space in Database (variable-length column) (DBCC CLEANTABLE)
	
	Where applicable, ALL_ERRORMSGS and/or NO_INFOMSGS options are always used.

@flgOptions	type = int
	default value = 3 (2 | 1)
	
	Various options to control the execution of the selected database consistency check actions.
	For multiple actions, parameter value is the sum of selected actions (each value represents a bit):
	    1 - run DBCC CHECKDB/DBCC CHECKTABLE using PHYSICAL_ONLY
	       o (default)
	    2 - use NOINDEX when running DBCC CHECKTABLE
	       o index consistency errors are not critical
	       o (default)
	   32 - Stop execution if an error occurs
	       o default behaviour is to print error messages and continue execution

@executionLevel	type = tinyint default value = 0 Controls the way messages are indented when printed. This represents the number of tab characters from the left side.
@debugMode	type = bit default value = 0 Controls which messages are printed. When enabled (1), dynamic SQL statements that are executed are also printed.

5.2 Database Maintenance

The stored procedure below manages the way indexes, heaps and/or statistics maintenance is performed:

```
[dbo].[usp_mpDatabaseOptimize]    @SQLServerName        [sysname]=@@SERVERNAME,
                                  @DBName               [sysname],
                                  @TableSchema          [sysname]     =    '%',
                                  @TableName            [sysname]     =    '%',
                                  @flgActions           [smallint]    =     27,
                                  @flgOptions           [int]         = 45697,
                                  @DefragIndexThreshold [smallint]    =      5,
                                  @RebuildIndexThreshold [smallint]   =     30,
                                  @PageThreshold        [int]         =   1000,
                                  @RebuildIndexPageCountLimit   [int] = 2147483647,
                                  @StatsSamplePercent   [smallint]    =    100,
                                  @StatsAgeDays         [smallint]    =      7,
                                  @StatsChangePercent   [smallint]    =      1,
                                  @MaxDOP               [smallint]    =      1,
                                  @executionLevel       [tinyint]     =      0,
                                  @DebugMode            [bit]         =      0
```

Parameter Name	Description
@SQLServerName	type = sysname default value = @@SERVERNAME The instance name on which the database optimization maintenance will be performed. Can be either a local instance or a linked server.
@DBName	type = sysname The database name for which optimization maintenance will be performed. It must be a valid database name. Wildcards or database lists are not supported.
@TableSchema	type = sysname default value = % Object schema to be analyzed. If this parameter is specified, only schemas with name matching it will be analyzed. Wildcard are supported. Lists are not supported.
@TableName	type = sysname default value = % Object name to be analyzed. Only objects (tables and materialized views) with name matching parameter value and with reserved pages will be analyzed. MS shipped objects and disabled indexes are skipped. Wildcards are supported. Lists are not supported.

@flgActions	type = smallint
	default value = 27 (16 | 8 | 2 |1)
	
	Select which database optimization maintenance tasks should be performed.
	For multiple actions, parameter value is the sum of selected actions (each value represents a bit):
	    1 - defragmenting database tables indexes (ALTER INDEX REORGANIZE)
	        o (default)
	    2 - rebuild heavy fragmented indexes (ALTER INDEX REBUILD)
	        o (default)
	    4 - rebuild all indexes (ALTER INDEX REBUILD)
	    8 - update statistics for table (UPDATE STATISTICS)
	        o (default)
	   16 - rebuild heap tables (SQL versions +2K5 only)
	        o (default)

@flgOptions	type = int
	default value = 45697 (32768 | 8192 | 4096 | 512 | 128 | 1)
	
	Various options to control the execution of the selected database optimization actions.
	For multiple actions, parameter value is the sum of selected actions (each value represents a bit):
	    1 - compact large objects (LOB)
	        o (default)
	    2 - not used
	    4 - rebuild all dependent indexes when rebuild primary indexes
	        o should be used along with option 8
	    8 - disable non-clustered index before rebuild (saves space)
	   16 - disable foreign key constraints that refer current table before rebuilding with disable clustered/unique indexes
	   32 - stop execution if an error occurs
	        o default behaviour is to print error messages and continue execution
	   64 - when enabling foreign key constraints do no check values
	        o default behaviour is to enable foreign key constraint with check option
	  128 - create statistics on index columns only
	        o (default)
	        o default behaviour is to not create statistics on all eligible columns
	  256 - create statistics using default sample scan
	        o default behaviour is to create statistics using full scan mode
	  512 - update auto-created statistics
	        o (default)
	 1024 - get index statistics using DETAILED analysis
	        o (default is to use LIMITED)
	        o for heap tables, DETAILS will always be used
	 4096 - rebuild/reorganize indexes using ONLINE=ON, if applicable
	        o (default)
	        o using this option will increase the transaction log that is being generated.
	           https://support.microsoft.com/en-us/kb/2407439
	 8192 - when rebuilding heaps, disable/enable table triggers
	        o (default)
	16384 - for versions below 2008 do heap rebuild using temporary clustered index
	32768 - analyse only tables with at least *@PageThreshold* pages reserved (+2k5 only)
	        o (default)
	65536 - clean-up of ghost records (`sp_clean_db_free_space`)
	        o this may be forced by setting the property *'Force cleanup of ghost records*' to true
	
	✓ @flgOptions value of 45697 ensures indexes are rebuild online, with the penalty of additional disk space requirement (default)
	✓ @flgOptions value of 45725 (45697 + 16 + 8 + 4) ensures minimum additional disk space requirements, but some objects are temporary disabled (indexes and foreign-keys)

@DefragIndexThreshold	type = smallint default value = 5 This parameter specifies the index reorganize/defragment threshold lower value. Indexes with fragmentation value lower than this parameter will be skipped.
@RebuildIndexThreshold	type = smallint default value = 30 This parameter specifies the index rebuild threshold lower value.
@PageThreshold	type = int default value = 1000 This parameter specifies the minimum number of pages an object (table or materialized view) should have in order to be analyzed. Objects with allocated pages less than current threshold will be skipped.
@RebuildIndexPageCountLimit	type = int default value = 2147483647 (approx. 16TB) This parameter specifies the maximum number of pages an index may have in order to be rebuild. If index size in pages exceeds this value, index will always be REORGANIZED.

@StatsSamplePercent	type = smallint default value = 100 This parameter specifies the sample percent to be used when updating statistics. Default 100 means full scan.
@StatsAgeDays	type = smallint default value = 7 This parameter specifies the age of statistics (in days) to be analyzed.
@StatsChangePercent	type = smallint default value = 1 This parameter specifies the minimum percent value of changes from which statistics will be updated.
@MaxDOP	type = smallint default value = 1 MAXDOP value to be used for index rebuild operations. If option 4096 is on and index can be rebuild online, SQL Server version is checked and @MaxDOP value is set to 1, if KB2969896 applies. https://support.microsoft.com/en-us/kb/2969896
@executionLevel	type = tinyint default value = 0 Controls the way messages are indented when printed. This represents the number of tab characters from the left side.
@debugMode	type = bit default value = 0 Controls which messages are printed. When enabled (1), dynamic SQL statements that are executed are also printed.

Index Reorganize / Rebuild algorithm
- o "drive" table for indexes to be analysed is built based on parameter values used for name filtering (@DBName, @TableSchema, @TableName) and on @PageThreshold value (if option 32768 is enabled)
- o logical fragmentation is determined as :
    - ▪ `avg_fragmentation_in_percent` from `sys.dm_db_index_physical_stats` (SQL 2005 and beyond) / `LogicalFrag` from DBCC SHOWCONTIG (SQL 2000)
- o additionally
    - ▪ `page_density_deviation = (floor (8060 / avg_record_size_in_bytes) * avg_record_size_in_bytes) / 8060 * fill_factor – avg_page_space_used_in_percent`
- o An index will be reorganized if:
    - ▪ allocated pages >= @PageThreshold
        - • *and*
            - o logical fragmentation >= @DefragIndexThreshold and fragmentation < @RebuildIndexThreshold
            - o *or* page_density_deviation >= @DefragIndexThreshold and page_density_deviation < @RebuildIndexThreshold and option 1024 is enabled
    - ▪ or allocated pages >= @RebuildIndexPageCountLimit
        - • *and*
            - o fragmentation >= @RebuildIndexThreshold
            - o *or* page_density_deviation >= @RebuildIndexThreshold and option 1024 is enabled
- o An index will be rebuilt if:
    - ▪ allocated pages >= @PageThreshold
    - ▪ *and* allocated pages < @RebuildIndexPageCountLimit
    - ▪ *and*
        - • fragmentation >= @RebuildIndexThreshold
        - • *or* page_density_deviation >= @RebuildIndexThreshold and option 1024 is enabled
- o only clustered, non-clustered, XML and spatial indexes are analysed
- o performing an index rebuild
    - ▪ an index may be disabled if
        - • option 8 is enabled (disable non-clustered indexes)

- *and* base table it not included in replication
- *and*
  - option 4096 is disabled (use online rebuild, if applicable)
  - *or* option 4096 is enabled but index rebuild cannot be performed online
  - https://msdn.microsoft.com/en-us/library/ms190981(v=sql.110).aspx
  - if option 16 is enabled and current index is used to enforce a foreign-key relationship and index is subject to being disabled
    - all foreign keys that refer current table will be disabled
    - https://technet.microsoft.com/en-us/library/ms177456(v=sql.110).aspx
  - when rebuilding clustered indexes
    - if option 4 is enabled all dependent indexes (non-clustered, XML and spatial) are also rebuilt
    - if index is subject to being disabled
      - all dependent indexes (non-clustered, XML and spatial) are disabled
  - when rebuilding XML primary indexes
    - if option 4 is enabled all secondary XML indexes are also rebuilt
    - if index is subject to being disabled
      - all secondary XML indexes are also disabled
  - SORT_IN_TEMPDB is always ON
    - https://msdn.microsoft.com/en-us/library/ms188281(v=sql.110).aspx
  - if option 4096 is enabled and index can be rebuilt online, it will be rebuilt online
    - default is to rebuild index with ONLINE=OFF
  - all disabled objects are enabled back
  - performing an index reorganize
    - can be performed only if current index allows page locks
    - LOB_COMPACTION option can be controlled using option 1 (enabled by default)
  - if option 32768 is enabled, only tables which have at least *@PageThreshold* allocated pages will be analysed for fragmentation (option valid starting with SQL 2005)
  - each operation is logged for undo/information purposes

Heap Rebuild algorithm
- "drive" table for heaps to be analysed is built based on parameter values used for name filtering (@DBName, @TableSchema, @TableName) and on @PageThreshold value (if option 32768 is enabled)
- extent fragmentation is determined as :
  - `avg_fragmentation_in_percent` from `sys.dm_db_index_physical_stats` (SQL 2005 and beyond)
- additionally
  - `forwarded_record_percentage = forwarded_record_count / record_count` from `sys.dm_db_index_physical_stats`
  - `page_density_deviation = (floor (8060 / avg_record_size_in_bytes) * avg_record_size_in_bytes) / 8060 * fill_factor – avg_page_space_used_in_percent`
- An heap will be rebuild if:
  - allocated pages >= @PageThreshold
  - *and*
    - extent fragmentation >= @RebuildIndexThreshold
    - *or* forwarded_record_percentage >= @DefragIndexThreshold
    - *or* page_density_deviation >= @RebuildIndexThreshold
  - http://sqlblog.com/blogs/tibor_karaszi/archive/2014/03/06/how-often-do-you-rebuild-your-heaps.aspx
- heap tables with disabled unique indexes will be skipped
  - heap rebuild involves index rebuild and unique indexes may enable unwanted constraints
- rebuild is performed as
  - algorithm 1 (default)
    - for versions greater or equal to 2008R2, ALTER TABLE REBUILD is executed
    - for versions below 2008R2 *and* when option 16384 is enabled

- o if option 8 is enabled all non-clustered, XML and spatial indexes are disabled
- o if option 16 is enabled all foreign keys that refer current table and all foreign keys defined on current table are being disabled
- o a clustered index based on a bigint column (temporary added to the table) is created and then dropped (temporary added column is also dropped)
- o if option 8 is enabled all non-clustered, XML and spatial indexes are rebuilt
- o if option 16 is enabled all foreign keys that refer current table and all foreign keys defined on current table are being enabled
- ▪ algorithm 2
  - • note: columns of type text, ntext, image and timestamp are excluded
  - • a copy of the table is taken
  - • if option 8 is enabled all non-clustered, XML and spatial indexes are disabled
    - o primary key is left enabled
  - • if option 16 is enabled all foreign keys that refer current table and all foreign keys defined on current table are being disabled
  - • if option 8192 is enabled all triggers defined on current table are disabled
  - • source table (the one being rebuilt) is being truncated (all records deleted) and DBCC UPDATEUSAGE is executed
  - • records are inserted back from the copy table (if applicable, IDENTITY_INSERT is set to ON)
    - o DBCC CHECKIDENT and DBCC UPDATEUSAGE are executed
  - • if option 8 is enabled all non-clustered, XML and spatial indexes are rebuilt
  - • if option 16 is enabled, all foreign keys that refer current table and all foreign keys defined on current table are being enabled
  - • if option 8192 is enabled all triggers defined on current table are enabled
- o each operation is logged for information purposes

Statistics update algorithm
- o "drive" table for statistics to be analysed is built based on parameter values used for name filtering (@DBName, @TableSchema, @TableName) and on @StatsAgeDays and @StatsChangePercent
- o Statistics will be updated if
  - Age >= @StatsAgeDays *and* Changes Made > 0
  - *or*
  - Age < @StatsAgeDays *and* Changes Made >= @StatsChangePercent
- o *Age* is computed as the number of days between `GetDate()` and `last_updated`
- o *Changes Made* is computed as :
  - ▪ for versions greater than or equal to 2008R2 SP2 (10.50.4000)
    - • `modification_counter / rows` from `sys.dm_db_stats_properties`
  - ▪ for versions lower than 2008R2 SP2
    - • `rowmodctr / rowcnt` from `sysindexes`
- o statistics for indexes which were rebuilt (including all dependent indexes when rebuilding a clustered index or a primary XML index if option 4 was enabled) will not be updated (as they are already up to date)
- o if option 512 is enabled auto-created statistics are updated
- o each operation is logged for information purposes

Ghost Records clean-up algorithm
- o if option 65536 is enabled or "*Force cleanup of ghost records*" option in `dbo.appConfigurations` table is set to true
  - ▪ index fragmentation analysis is performed using DETAILED mode (default is LIMITED)
  - ▪ if the sum of `ghost_records_count` column for indexes which were not rebuilt or reorganized is higher than a threshold value
    - • `sp_clean_db_free_space` is executed for the current database
    - • https://msdn.microsoft.com/en-us/library/dd408732(v=sql.110).aspx
  - ▪ default threshold is 131072 (128k), value of "Ghost records cleanup threshold" option in `dbo.appConfigurations` table

danandrei.stefan@gmail.com Document date: 20/08/2015

5.3 Database Backup

The stored procedure below manages the way database backups and backup retention policy are performed:

```
[dbo].[usp_mpDatabaseBackup]        @sqlServerName      [sysname]= @@SERVERNAME,
                                    @dbName             [sysname],
                                    @backupLocation     [nvarchar](1024)=NULL,
                                    @flgActions         [smallint]  = 1,
                                    @flgOptions         [int]       = 2011,
                                    @retentionDays      [smallint]  = NULL,
                                    @executionLevel     [tinyint]   = 0,
                                    @debugMode          [bit]       = 0
```

Parameter Name	Description
@sqlServerName	type = sysname default value = @@SERVERNAME The instance name on which the database backup will be performed. Can be either a local instance or a linked server.
@dbName	type = sysname The database name for which a backup will be performed. It must be a valid database name. Wildcards or database lists are not supported.

@backupLocation	type = nvarchar(1024)
	
	Location on which backup files will be placed (disk or UNC path)
	If value is null, the value of "Default backup location" option from dbo.appConfiguration table will be used.
	
	Server name and database name (if option 64 is enabled) are appended to this parameter value:
	• The backup files destination will be:
	    @backupLocation\@sqlServerName\@dbName
	• If destination folder/subfolder does not exists, they will be created
	    o error is raised is folder cannot be created
	
	File name format is:
	• @sqlServerName_@dbName_yyyymmdd_hhmmss_{type}.{ext}
	    o {type} could be: full, diff or log
	    o {ext} could be: BAK, TRN

@flgActions	type = smallint
	default value = 1
	
	Select which database backup type should be performed.
	Multiple actions are not allowed for this parameter.
	    1 - perform full database backup
	        o for versions greater than or equal to 2005 and when option 1 is not enabled DATA_PURITY option is always used
	    2 - perform differential database backup
	        o for versions greater than or equal to 2005 DATA_PURITY option is always used
	        o if option 8 is enabled will check for a full database backup
	        o not allowed for master database
	    4 - perform transaction log backup
	        o will not be performed if database recovery model is SIMPLE
	        o if option 8 is enabled will check for a full or differential database backup
	If option 8 is enabled and a full database backup is missing a full database backup will be taken before the current backup (diff or log).

@flgOptions	type = int
	default value = 2011 (1024 | 512 | 256 | 128 | 64 | 16 | 8 | 2 | 1)
	
	Various options to control the execution of the selected database consistency checks actions.
	For multiple actions parameter value is the sum of selected actions (each value represents a bit):
	    1 - use CHECKSUM
	        o (default)
	        o for versions greater than or equal to 2005
	    2 - use COMPRESSION, if available
	        o (default)

<table>
<tr><td></td><td>

       o   for version 2008, only for Enterprise or Developer editions

       o   for versions greater than or equal to 2008 R2 (Enterprise, Developer and Standard editions)

4   - use COPY_ONLY

       o   only for versions greater than or equal to 2005

8   - force change backup type

       o   (default)

       o   if backup type *log* is selected and no database backup is found (full or diff) a full database backup will be triggered first

       o   if backup type *diff* is selected and no full database backup is found a full database backup will be triggered first

16   - verify backup file

       o   (default)

       o   for versions greater than or equal to 2005, and when option 1 was enabled, VERIFYONLY is executed using the CHECKSUM option

32   - Stop execution if an error occurs

       o   default behaviour is to print error messages and continue execution

64   - create folders for each database

       o   (default)

128 - when performing cleanup and deleting full database backups also delete orphan diff and log backups

       o   (default)

256 - for +2k5 versions, use `xp_delete_file` option

512 - skip databases involved in log shipping

       o   (default)

       o   primary or secondary for transaction log backup

       o   secondary for full or diff backup

1024 - on AlwaysOn Availability Groups, for secondary replicas, force copy-only backups

       o   (default)

       o   for versions greater than or equal to 2012

2048 - change retention policy from RetentionDays to RetentionBackupsCount

       o   (number of full database backups to be kept)

       o   this may be forced by setting the property *'Change retention policy from RetentionDays to RetentionBackupsCount'* to true

</td></tr>
<tr><td>@retentionDays</td><td>

type = smallint
default value = NULL

If value is null, the value of '*Default backup retention (days)*' option from `dbo.appConfiguration` table will be used.

Number of days (or backup count, if option 2048 is enabled) to keep database backup files. Older files will be deleted from disk.

    o   If @retentionDays is set to Days, this number represents the number of days for which database can be restored depending on the backup strategy, a full backup will always be included

    o   If @retentionDays is set to BackupCount, this number represents the number of full and differential backups to be kept; an older full backup may exists to ensure that a newer differential backup can be restored

</td></tr>
<tr><td>@executionLevel</td><td>

type = tinyint
default value = 0

Controls the way messages are indented when printed. This represents the number of tab characters from the left side.

</td></tr>
<tr><td>@debugMode</td><td>

type = bit
default value = 0

Controls which messages are printed. When enabled (1), dynamic SQL statements that are executed are also printed.

</td></tr>
</table>

Database backup clean-up algorithm
- o If option 2048 is enabled or *'Change retention policy from RetentionDays to RetentionBackupsCount'* option from `dbo.appConfiguration` is set to true
    - ▪ Instead of using days for backup age, will use number or backups (full or diff)
- o a full backup will be identified to be kept in order to ensure that newer backups can be restored (may be older than @RetentionDays parameter value)
    - ▪ the full backup start date represents the date from which older files will be deleted
    - ▪ "kept full database backup"

- o a diff backup will be identified in order to ensure the database can be restored within the @retentionDays period
  - differential backup (if exists) will be used along with the full backup already identified
  - "last kept differential backup"
- o for versions greater than or equal to 2005 and when option 256 is enabled
  - `xp_delete_file` is executed for files older than the "kept full database backup"
- o for version 2000 or when option 256 is disabled or when option 128 is enabled
  - for versions greater than or equal 2005, if `xp_cmdshell` configuration option is not enabled, temporary enable it
  - delete files older than the date of the "kept full database backup"
  - delete differential and transaction log backups older than the date of the "last kept differential backup"

6. XML schemas & additional details in dbo.vw_logEventMessages

6.1 Actions / SQL execution details

This schema is used to store information on all actions made (SQL statement executed, OS command ran, etc.), their duration and exit code.
Duration format is always ##h ##m ##s.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="action">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="detail">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="database_name" type="xs:string" minOccurs="0"></xs:element>
              <xs:element name="event_name" type="xs:string"></xs:element>
              <xs:element name="object_name" type="xs:string" minOccurs="0"></xs:element>
              <xs:element name="child_object_name" type="xs:string" minOccurs="0"></xs:element>
              <xs:element name="query_executed" type="xs:string"></xs:element>
              <xs:element name="duration" type="xs:string"></xs:element>
              <xs:element name="error_code" type="xs:int"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.2 Alert event details

This schema is used to store information on an alert event.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="alert">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="detail">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="error_code" type="xs:int"></xs:element>
              <xs:element name="error_string" type="xs:string"></xs:element>
              <xs:element name="query_executed" type="xs:string"></xs:element>
              <xs:element name="duration_seconds" type="xs:int"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.3 Backup file details

This schema is used to store information on a backup file.
Duration format is always ##h ##m ##s.
Size format is always ####.## mb

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:element name="backupset">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="detail">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="database_name" type="xs:string"></xs:element>
               <xs:element name="type" type="xs:string"></xs:element>
               <xs:element name="start_date" type="xs:string"></xs:element>
               <xs:element name="duration" type="xs:string"></xs:element>
               <xs:element name="size" type="xs:string"></xs:element>
               <xs:element name="size_bytes" type="xs:int"></xs:element>
               <xs:element name="verified" type="xs:string"></xs:element>
               <xs:element name="file_name" type="xs:string"></xs:element>
               <xs:element name="error_code" type="xs:int"></xs:element>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
</xs:schema>
```

6.4 Index fragmentation details

This schema is used to store information on index fragmentation before it was reorganized/rebuilt.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:element name="index-fragmentation">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="detail">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="database_name" type="xs:string"></xs:element>
               <xs:element name="object_name" type="xs:string"></xs:element>
               <xs:element name="index_name" type="xs:string"></xs:element>
               <xs:element name="index_type" type="xs:string"></xs:element>
               <xs:element name="fragmentation" type="xs:double"></xs:element>
               <xs:element name="page_count" type="xs:int"></xs:element>
               <xs:element name="fill_factor" type="xs:int"></xs:element>
               <xs:element name="page_density_deviation" type="xs:int"></xs:element>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
</xs:schema>
```

6.5 Statistics health details

This schema is used to store information on statistic health before it was updated.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:element name="statistics-health">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="detail">
           <xs:complexType>
```

```
        <xs:sequence>
          <xs:element name="database_name" type="xs:string"></xs:element>
          <xs:element name="object_name" type="xs:string"></xs:element>
          <xs:element name="stats_name" type="xs:string"></xs:element>
          <xs:element name="auto_created" type="xs:int"></xs:element>
          <xs:element name="rows" type="xs:int"></xs:element>
          <xs:element name="modification_counter" type="xs:int"></xs:element>
          <xs:element name="percent_changes" type="xs:int"></xs:element>
          <xs:element name="last_updated" type="xs:string"></xs:element>
          <xs:element name="age_days" type="xs:int"></xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
</xs:schema>
```

6.6 Heap table fragmentation details

This schema is used to store information on heap table fragmentation details before it was rebuilt.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="heap-fragmentation">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="detail">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="database_name" type="xs:string"></xs:element>
              <xs:element name="object_name" type="xs:string"></xs:element>
              <xs:element name="fragmentation" type="xs:int"></xs:element>
              <xs:element name="page_count" type="xs:int"></xs:element>
              <xs:element name="page_density_deviation" type="xs:double"></xs:element>
              <xs:element name="forwarded_records_percentage" type="xs:double"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.7 Job execution details

This schema is used to store information on job execution details.
Duration format is always ##h ##m ##s.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="job-history">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="job-step">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="step_id" type="xs:int"></xs:element>
              <xs:element name="step_name" type="xs:string"></xs:element>
              <xs:element name="run_status" type="xs:string"></xs:element>
              <xs:element name="run_date" type="xs:date"></xs:element>
              <xs:element name="run_time" type="xs:string"></xs:element>
              <xs:element name="duration" type="xs:string"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.8 HTML report generation details

This schema is used to store information on HTML report generation details.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:element name="report-html">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="detail">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="message" type="xs:string"></xs:element>
               <xs:element name="file_name" type="xs:string"></xs:element>
               <xs:element name="relative_path" type="xs:string"></xs:element>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
</xs:schema>
```

b. health-check

This module performs SQL Server instance and database analysis and applies issue detection rules to a set of monitored instances.

1. Discovery and Refreshing information

1.1 Catalog Upsert: Discovery & Update

- discovery made using "sqlcmd –L"
  - o `xp_cmdshell` may be enabled/disabled
- in order for an instance to be auto-discovered, SQL Server Browser service must be enabled and started
- creates a linked server using instance name
  - o by default connections will be made using current login's security context
  - o when the job will run it will use the SQL Server Agent service account credentials
- linked server definition can be overwritten, but at least the below minimal permissions are needed on the remote server:
  - o VIEW SERVER STATE
  - o VIEW DATABASE STATE on each of the user/system databases
  - o for some features sysadmin permissions may be needed
- enables instance status: online/active or offline

1.2 Collect Database Status & Details

- get information about databases only for online/active instances
- for SQL Server 2005 and beyond, it runs DBCC DBINFO to retrieve the last date DBCC CHECK was executed

1.3 Collect SQL Server Agent job status

- get information about last execution status for SQL Server agent jobs, only for online/active instances

1.4 Collect Disk Space Usage information

- get information on free disk space, only for online/active instances
- may run WMI statements, for which `xp_cmdshell` will be enabled/disabled
- for SQL Server 2008 R2 and beyond, also get information on mounted volumes

1.5 Collect Errorlog Messages

- get information from active errorlog file, only for online/active instances
- for SQL Server 2000 instances, errorlog file will be parsed and split to match the format used in SQL Server 2005 and beyond

1.6 Collect Event Messages

- get all internal logged events, from online/active instances
- events are mainly logged during maintenance-plan module as varchar/xml
- this is an internal feature only, but information may be accessed from `dbo.vw_logEventMessages`

SQL Server Agent Jobs:
- dbaTDPMon - Discovery & Health Check
  - o *job is automatically started by another SQL Server Agent job*

2. Generate Reports

A comprehensive HTML report is generated for all analysed SQL Server instances (instances can be organized into projects / report is generated per project).

The report is split into 2 major areas:
- Modules
  - covers day-to-day DBA checks
  - provides information on the instances / environment analysed
- Potential issues
  - Hard-coded rules to detect potential database related issues

2.1 Modules

2.1.1. Instance Availability
Provide information on online instances (version / edition / databases size / etc).
Provide information on offline instances (error messages raised while trying to connect).
Left menu (internal HTML links) to information on other modules (for online instances only).

For online instances only:

2.1.2. Databases Status
Provide information on instance databases (name / data size / log size / backup date / dbcc date).

2.1.3. SQL Server Agent Jobs Status
Provide information on instance SQL Agent jobs and their status (execution date/time, duration, messages).

2.1.4. Disk Space Information
Provide information on machine disk space (disk space / free space); only drives used by database files plus C: drive will be shown.

2.1.5. Errorlog Messages
Provide errorlog messages printed within the last X hours (default 24). This option is disabled by default due to report size constraints.

2.2 Potential Issues

2.2.1. Offline Databases
Databases for which status is not in (ONLINE, READ ONLY).

2.2.2. SQL Server Agent Job Failures
SQL Agent jobs for which status is not in (SUCCEDED, IN PROGRESS) and last execution happened in the last 24 hours.

2.2.3. Low Free Disk Space
Machines and drive / mount points for which free disk space is below 10% or 3000MB.

2.2.4. Outdated Backup for Databases
System databases for which last backup date is greater than 7 days.
User databases for which last backup date is greater than 2 days.

2.2.5. Outdated DBCC CHECKDB Databases
System databases for which last DBCC CHECKDB date is greater than 14 days.
User databases for which last DBCC CHECKDB date is greater than 14 days.

2.2.6. Errorlog Messages
Maximum of 500 messages per instance, from errorlog, that contains errors. Hardcoded filters are applied to exclude unwanted messages.

2.2.7. Big Size for System Databases
Instances on which *master* database size is greater than 32 MB or *msdb* database size is greater than 1024 MB.

2.2.8. Big Size for Database Log Files
Databases for which log file size is greater than 32 GB.

2.2.9. Databases with Auto Close / Shrink
Databases for which auto close or auto shrink options are set to true.

2.2.10. Low Usage of Data Space
Databases with size greater than 512 MB which have less than 50% of space used (wasted space, reclaimable via DBCC SHRINKDATABASE).

2.2.11. High Usage of Log Space
Databases with size greater than 512 MB which have more than 50% of log space used (a transaction log backup may be required).

2.2.12. Log vs. Data – Allocated Size
Databases with size greater than 512 MB and for which log size vs. data size is greater than 90% (may spot transaction intensive databases).

2.2.13. Databases with Fixed File(s) Size
Databases for which any of the data or log file has a fixed file size (require internal space monitoring to avoid file getting full).

2.2.14. Databases with Improper Page Verify Option
Databases for which running SQL Server version is greater than 2005, but Page Verify is NOT set to CHECKSUM, or which run on SQL Server version 2000 and Page Verify is set to NONE.

2.2.15. Frequently Fragmented Indexes
Indexes which got fragmented within a period of 2 days (default) and were either rebuild (default) or reorganized in the last 24 hours (default).

For these indexes, consider lowering the fill-factor in order to reduce the page-splits and fragmentation.
`dbo.usp_hcChangeFillFactorForIndexesFrequentlyFragmented` stored procedure can be used to change the fill-factor to all detected frequently fragmented indexes (lower current fill-factor by a specified percent)

3. Report Configuration options
3.1 Reports path and email distribution list

Table `dbo.appConfigurations` contains parameters that can be used for report configuration:

name	value	description
Default project code	NULL	Project code to be used, when not specified as parameter
Database Mail profile name to use for sending emails	NULL	Database mail profile to be used for email notifications
Default recipients list - Reports (semicolon separated)	NULL	List of email addresses to which the generated report will be sent as attachment
Local storage path for HTML reports	NULL	Full path to disk where the HTML files will be saved
HTTP address for report files	NULL	If an IIS server is available, the URL to be used for browsing report version

3.2 Configuration Thresholds and Options

Table dbo.reportHTMLOptions contains configuration thresholds and options for each of the above:

name	value	description
Database online admitted state	ONLINE, READ ONLY	Comma separated, default ONLINE, READ ONLY
Database max size (mb) - master	32	Maximum allowed size for master database; default 32
Database max size (mb) - msdb	1024	Maximum allowed size for msdb database; default 1024
SQL Agent Job - Failures in last hours	24	Report job failures in the last hours; default 24
Database Min Size for Analysis (mb)	512	Minimum size of the database to be analyzed; default 512

Database Max Log Size (mb)	32768	Maximum allowed size for log file; default 32768
Database Min Data Usage (percent)	50	Minimum allowed percent for data space usage; default 50
Database Max Log Usage (percent)	50	Maximum allowed percent for log space usage; default 50
Database Log vs. Data Size (percent)	90	Maximum allowed percent between log and data allocated size; default 90
User Database BACKUP Age (days)	2	Maximum allowed age in days for outdated backups; default 2
System Database BACKUP Age (days)	7	Maximum allowed age in days for outdated backups; default 7
User Database DBCC CHECKDB Age (days)	14	Maximum allowed age in days for outdated DBCC CHECKDB; default 30
System Database DBCC CHECKDB Age (days)	14	Maximum allowed age in days for outdated DBCC CHECKDB; default 30
Free Disk Space Min Percent (percent)	10	Minimum allowed percent for free disk space, default 10
Free Disk Space Min Space (mb)	3000	Minimum allowed free disk space in mb, default 3000
Errorlog Messages in last hours	24	Report errorlog messages in the last X hours; default 24
Errorlog Messages Limit to Max	500	Limit errorlog messages to a maximum number; default 1000
Minimum Index Maintenance Frequency (days)	2	Interval between 2 index maintenance operations for the same HoBT; default 2
Analyze Index Maintenance Operation	REBUILD	Which index maintenance operation to analyze (REBUILD and/or REORGANIZE)
Analyze Only Messages from the last hours	24	Analyze only messages raised in the last hours; default 24

SQL Server Agent Jobs:
- dbaTDPMon - Generate Reports
  - *daily at 04:00:00*