dbaTDPMon

TROUBLESHOOT DATABASE PERFORMANCE & MONITORING

Version 2023.10

This utility is a bespoke database / system maintenance, health-check and monitoring solution for SQL Server. It runs for versions from SQL Server 2000 until 2022. 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.
Maintenance-plan, health-check and monitoring modules can be used as “agent less” management system

Why dbaTDPMon?

- implement database maintenance best practices (including system databases)
- support for parallel database maintenance (multiple databases at once)
- 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
- ***Columnstore Index Maintenance***
  - decision based on deleted records and deleted segments
- ***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
- Always On Availability Groups “aware”

Daily Health Checks & Monitoring

- online/offline instances and databases health state
- report failed SQL Agent jobs / disk space issues / replication issues / long or blocked transactions, etc.
- report outdated backups and checkdb
- analyze errorlogs and OS Event logs
- collect data from multiple servers in parallel

 danandrei.stefan@gmail.com Document date: 14/10/2023

Table of Contents

 danandrei.stefan@gmail.com

I. Installation / Update

1. Utility installation

The utility can be downloaded from: http://github.com/rentadba/dbaTDPMon
It comes as a zip file containing SQL files (objects and code), full documentation (this document) and a powershell script to be used for install, upgrade or uninstall.
In order to install the utility, run the `.\setupTDP.ps1` script the needed parameters:

```
setupTDP.ps1  -instanceName = "$env:ComputerName"
              -databaseName = "dbaTDPMon"
              -module="all"
              -projectName="PRODUCTION"
              -useParallelExecution="Yes"
              -recreateSQLAgentJobs="No"
              -dataFilePath=""
              -logFilePath=""
              -sqlLoginName=""
              -sqlLoginPassword=""
              -queryTimeout = 1800
              -uninstall="No"
```

where:

instanceName	the SQL instance on where to install the utility
databaseName	utility database name; usually it is dbaTDPMon
module	which of the utility module to install; available options are: ▪ maintenance-plan ▪ health-check (for SQL versions greater than 2k) ▪ all (for SQL versions greater than 2k)
projectName	a "name" of the SQL Server instances group; an alias to be used to monitor a group of SQL Servers; default value is PRODUCTION
useParallelExecution	if internal tasks should be executed in parallel (default), using SQL Agent jobs or in a serial fashion
recreateSQLAgentJobs	when performing upgrade, if the current SQL Agent jobs will be kept (default) or they will be dropped and recreated
dataFilePath	full path location for the utility database data files default value is: *HKLM\Software\Microsoft\MSSQLServer\MSSQLServer\DefaultData*
logFilePath	full path location for the utility database log file(s) default value is: *HKLM\Software\Microsoft\MSSQLServer\MSSQLServer\DefaultLog*
sqlLoginName	a sysadmin login name to be used to deploy the utility
sqlLoginPassword	the login password
uninstall	when Y(es), utility will perform a clean uninstall

Notes:

- if repository database is not found, a clean install will be performed
- if a repository database is found, upgrade is being performed and patch files are being applied
- for SQL Server 2000 systems, SQLCMD (2k5+) is mandatory. It can be downloaded from:
  https://www.microsoft.com/en-us/download/details.aspx?id=24793
  SQLServer2005_SQLCMD.msi & sqlncli.msi
  SQLServer2005_SQLCMD_x64.msi & sqlncli_x64.msi

Sample calls:

- install on local machine/default instance, only the maintenance-plan module:
  `./setupTDP.ps1  -instanceName . -databaseName dbaTDPMon -module maintenance-plan`
- patch an existing repository and deploy all the other modules
- `./setupTDP.ps1  -instanceName . -databaseName dbaTDPMon -module all`

 danandrei.stefan@gmail.com

II. Modules description

a. maintenance-plan

1. Database Backup

- file name template: `ServerName_DBName_yyyymmdd_hhmmss_BackupType.Ext`
  - for AlwaysOn Availability Groups, instead of {ServerName}, {ClusterName } is used
  - BackupType: FULL, DIFF, LOG
  - file extensions: BAK / TRN
  - each database on its own folder
- 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)
  - if not set, can be sent as a stored procedure parameter value
- default retention 7 backups (at least one full included) (set up in `dbo.appConfiguration`)
  - may change to last 7 days
  - 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 and Diff
  - daily at 00:00:00, system database on Saturday only
  - job sends full execution log over email (+2k5)
    - backupset information included in email
- dbaTDPMon - Database Backup – Log
  - daily, every hour

2. User Database Maintenance

- Daily: Kill Orphan Connections (+2k5)
  - *Not available in the parallel version of the maintenance-plan*
- 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*
  - *by default, disabled in the parallel version of the maintenance-plan*

 danandrei.stefan@gmail.com

- 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: 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: 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*)
    - create statistics for all columns which are part of an index (sp_createstats 'indexonly')
- Weekly: Shrink Database (TRUNCATEONLY)
    - DBCC SHRINKDATABASE WITH TRUNCATEONLY
    - *on Sunday;*
    - *by default, disabled in the parallel version of the maintenance-plan*
- Weekly: Shrink Log File
    - DBCC SHRINKFILE WITH TRUNCATEONLY
    - *first Saturday of the month;*
    - *by default, disabled in the parallel version of the maintenance-plan*

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

- 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*
- distribution - 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`

 danandrei.stefan@gmail.com

- 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*

Note: "\maintenance-plan\job-scripts\msdb-create-custom-indexes.sql" script is executed at install time in order to create additional / needed custom indexes in order to improve msdb database cleanup speed and maintenance execution times.

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
Flood control: maximum alerts in 5 minutes	50	Spam filter
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
Ignore alerts for: Error 15281 - SQL Server blocked access to procedure	TRUE	Do not generate alerts when timeout errors are encountered
Ignore alerts for: Error 1927 - There are already statistics on table	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,
                                            @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 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 analyzed 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 behavior is to print error messages and continue execution
@maxDOP	type = smallint default value = 0 Controls the parallelism degree. Applies to SQL Server 2014 SP2 onwards.
@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,
                                    @skipObjectsList       [nvarchar](1024) = NULL,
                                    @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); for columnstore indexes will compress all row groups (COMPRESS_ALL_ROW_GROUPS)(default) 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 behavior is to print error messages and continue execution 64 - when enabling foreign key constraints do no check values o default behavior is to enable foreign key constraint with check option 128 - create statistics on index columns only o (default) o default behavior is to not create statistics on all eligible columns 256 - create statistics using default sample scan o default behavior 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, DETAILED 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 - analyze 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.
@rebuildIndexPageCount Limit	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 = 0 This parameter specifies the sample percent to be used when updating statistics. 0 means using default sample size, 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 / heap 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 Starting with SQL Server 2016 SP2 and SQL Server 2017 RTM CU3, @maxDOP value can be used for updating statistics.
@skipObjectsList	type = nvarchar(1024) default value = null Comma separate list of objects (table names, index names, statistic names) to be excluded from the maintenance
@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

- "drive" table for indexes to be analyzed is built based on parameter values used for name filtering (@DBName, @TableSchema, @TableName) and on @PageThreshold value (if option 32768 is enabled)
- 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)
- additionally, internal fragmentation is determined as:
  - `page_density_deviation = (floor (8060 / avg_record_size_in_bytes) * avg_record_size_in_bytes) / 8060 * fill_factor - avg_page_space_used_in_percent`
- columnstore index fragmentation is determined as:
  - `SUM(deleted_rows)/ SUM(total_rows) = avg_fragmentation_in_percent`
  - `SUM(segment(deleted_rows=total_rows)) / count(segments) = deleted_segment_percentage`

 danandrei.stefan@gmail.com

- An index will be reorganized if:
  - allocated pages >= @PageThreshold
    - *and*
      - logical fragmentation >= @DefragIndexThreshold and fragmentation < @RebuildIndexThreshold
      - *or* page_density_deviation >= @DefragIndexThreshold and page_density_deviation < @RebuildIndexThreshold and option 1024 is enabled
      - *or* logical *deleted_segment_percentage* >= @DefragIndexThreshold and fragmentation < @RebuildIndexThreshold
  - or allocated pages >= @RebuildIndexPageCountLimit
    - *and*
      - fragmentation >= @RebuildIndexThreshold
      - *or* page_density_deviation >= @RebuildIndexThreshold and option 1024 is enabled
- 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
    - *deleted_segment_percentage* >= @RebuildIndexThreshold
- 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 analyzed for fragmentation (option valid starting with SQL 2005)
- each operation is logged for undo/information purposes

 danandrei.stefan@gmail.com

Heap Rebuild algorithm

- "drive" table for heaps to be analyzed 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
      - if option 8 is enabled all non-clustered, XML and spatial indexes are disabled
      - if option 16 is enabled all foreign keys that refer current table and all foreign keys defined on current table are being disabled
      - a clustered index based on a bigint column (temporary added to the table) is created and then dropped (temporary added column is also dropped)
      - 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
  - 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
      - 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)
      - 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
- each operation is logged for information purposes

 danandrei.stefan@gmail.com

Starting with SQL Server 2014 onwards, for online index or heap rebuilds, the below construct will be used:

WAIT_AT_LOW_PRIORITY(MAX_DURATION = [..] MINUTES, ABORT_AFTER_WAIT=SELF)

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

Name	value
WAIT_AT_LOW_PRIORITY max duration (min)	1

Statistics update algorithm

- "drive" table for statistics to be analyzed is built based on parameter values used for name filtering (@DBName, @TableSchema, @TableName) and on @StatsAgeDays and @StatsChangePercent
- Statistics will be updated if
  Age >= @StatsAgeDays *and* Changes Made > 0
  or
  Age < @StatsAgeDays *and* Changes Made >= @StatsChangePercent
- *Age* is computed as the number of days between `GetDate()` and `last_updated`
- *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`
- 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)
- if option 512 is enabled auto-created statistics are updated
- each operation is logged for information purposes

Partitioning

- if a table/heap or index is partitioned, the fragmentation analysis will be done for each partition and only the fragmented ones will be reorganize / rebuild using the above algorithms
- starting with SQL Server 2014 onwards, for partitioned tables and indexes, incremental statistics update is supported

Ghost Records clean-up algorithm

- 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

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 ○ error is raised is folder cannot be created File name format is: • @sqlServerName_@dbName_yyyymmdd_hhmmss_{type}.{ext} ○ {type} could be: full, diff or log ○ {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 ○ 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 ○ for versions greater than or equal to 2005 DATA_PURITY option is always used ○ if option 8 is enabled will check for a full database backup ○ not allowed for master database 4 - perform transaction log backup ○ will not be performed if database recovery model is SIMPLE ○ 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 ○ (default) ○ for versions greater than or equal to 2005 2 - use COMPRESSION, if available ○ (default) ○ for version 2008, only for Enterprise or Developer editions ○ for versions greater than or equal to 2008 R2 (Enterprise, Developer and Standard editions) 4 - use COPY_ONLY ○ only for versions greater than or equal to 2005 8 - force change backup type ○ (default) ○ if backup type *log* is selected and no database backup is found (full or diff) a full database backup will be triggered first ○ 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 ○ (default) ○ 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 ○ default behavior 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 4096 - use xp_dirtree to identify orphan backup files to be deleted, when using option 128 (default) 8192 - take tail log backup - NORECOVERY
@retentionDays	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 A value of 0 means the backup cleanup process will not be executed.
@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.

Database backup clean-up algorithm

- 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)
- 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"
- 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"
- 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"
- 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. AlwaysOn Availability Groups limitations

The below limitations are imposed while running maintenance on PRIMARY replicas:
database backup

- If Backup Preference is set to 1: Secondary Only, log backups are not allowed on primary
- if Backup Preference is set to 2: Preferred Secondary, performing backups on the primary replica is acceptable if no secondary replica is available for backup operations

The below limitations are imposed while running maintenance on SECONDARY replicas:
database backup

- copy-only full backups are allowed
- differential backups are not supported on secondary replicas.
- BACKUP LOG supports only regular log backups (the COPY_ONLY option is not supported for log backups on secondary replicas)
- to back up a secondary database, a secondary replica must be able to communicate with the primary replica and must be SYNCHRONIZED or SYNCHRONIZING.
- If Backup Preference is set to 0: Primary, no backup type is allowed on secondary replica
- backups are allowed only on preferred secondary replica

database maintenance

- shrink is not allowed on a secondary replica
- DBCC UPDATEUSAGE / DBCC CLEANTABLE cannot be run on a secondary replica
- reorganize / rebuild index, update statistics, rebuilding heap are not allowed on a secondary replica

If replica state is not in SYNCRONIZING or SYNCRONIZED, backup operation will be skipped and an alert message will be sent over email (if configured).

7. XML schemas & additional details in dbo.vw_logEventMessages

7.1 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 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>
```

7.2 Alert event details

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

```
<?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>
```

7.3 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 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>
```

7.4 Index fragmentation details

This schema is used to store information on index fragmentation before it was reorganized/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="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="partition_number" type="xs:int"></xs:element>
                     <xs:element name="is_partitioned" type="xs: int "></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>
```

7.5 Statistics health details

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

```
<?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="is_incremental" type="xs: int "></xs:element>
                <xs:element name="partition_number" 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>
```

7.6 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="partition_number" type="xs:int"></xs:element>
                <xs:element name="is_partitioned" type="xs: int "></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>
```

7.7 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>
```

7.8 HTML report generation details

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

```
<?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>
```

7.9 Skip action information detail

This schema is used to store information on actions that were skipped due to applicable restrictions:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
      <!-- XML Schema Generated from XML Document on Wed Nov 04 2015 10:26:49 GMT+0200 (GTB Standard Time) -->
      <!-- with XmlGrid.net Free Online Service http://xmlgrid.net -->
    <xs:element name="skipaction">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="detail">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="name" type="xs:string"></xs:element>
                            <xs:element name="type" type="xs:string"></xs:element>
                            <xs:element name="affected_object" type="xs:string"></xs:element>
                            <xs:element name="date" type="xs:string"></xs:element>
                            <xs:element name="reason" 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”
  - `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
  - by default connections will be made using current login’s security context
  - 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:
  - VIEW SERVER STATE
  - VIEW DATABASE STATE on each of the user/system databases
  - 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 SQL Errorlog Messages

- get information from active errorlog file(s), only for online/active instances
  - number of errorlog files to be analyzed can be configured in `dbo.appConfigurations`
- 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 OS Event Messages

- get information from Application, System and Setup event messages
- by default, only Critical and Error messages are collected
  - Warning and Information can be collected optionally
- a powershell script is used to get the event messages from each instance host OS

1.7 Collect Internal 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`

2. Generate Reports

A comprehensive HTML report is generated for all analyzed 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 analyzed
- 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 and mounted points 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.

2.1.6. OS Event Messages

Provide OS Event messages printed within the last X hours (default 24). By default, only Critical and Error level messages are collected and printed. Application, Setup and System logs are analyzed.

2.2 Potential Issues

2.2.1. Offline Databases

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

✓ *Configurable though option "Database online admitted state".*

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.

✓ *Configurable though option "SQL Agent Job - Failures in last hours".*

2.2.3. Long Running SQL Agent Jobs

SQL Agent jobs that are currently running for more than 3 (default) hours.

✓ *Configurable though option "SQL Agent Job - Maximum Running Time (hours)".*

2.2.4. Low Free Disk Space

Machines and drive / mount points for which free disk space is below 10% or 3000MB.

✓ *Configurable though options "Free Disk Space Min Percent (percent)" and "Free Disk Space Min Space (mb)".*

 danandrei.stefan@gmail.com

2.2.5. Outdated Backup for Databases

System databases for which last backup date is greater than 7 days.

- ✓ *Configurable though option "System Database BACKUP Age (days)".*

User databases for which last backup date is greater than 2 days.

- ✓ *Configurable though option "User Database BACKUP Age (days)"*

2.2.6. Outdated DBCC CHECKDB Databases

System databases for which last DBCC CHECKDB date is greater than 14 days.

- ✓ *Configurable though option "System Database DBCC CHECKDB Age (days)"*

User databases for which last DBCC CHECKDB date is greater than 14 days.

- ✓ *Configurable though option "User Database DBCC CHECKDB Age (days)"*

2.2.7. 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)

- ✓ *Configurable though options "Minimum Index Maintenance Frequency (days)", "Analyze Index Maintenance Operation", "Analyze Only Messages from the last hours".*

2.2.8. Errorlog Messages

Maximum of 500 messages per instance, from errorlog, that contains errors. Hardcoded filters are applied to exclude unwanted messages. Table `report.hardcodedFilters` contains the hardcoded filters, which can be turned on/off.

- ✓ *Configurable though option "Errorlog Messages in last hours", "Errorlog Messages Limit to Max".*

2.2.9. 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.

- ✓ *Configurable though option "Database max size (mb) - master", "Database max size (mb) - msdb".*

2.2.10. Big Size for Database Log Files

Databases for which log file size is greater than 32 GB.

Configurable though option "Database Max Log Size (mb)".

2.2.11. Databases with Auto Close / Shrink

Databases for which auto close or auto shrink options are set to true.

2.2.12. 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).

- ✓ *Configurable though option "Database Min Data Usage (percent)".*

2.2.13. 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).

- ✓ *Configurable though option "Database Max Log Usage (percent)".*

2.2.14. 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).

- ✓ *Configurable though option "Database Log vs. Data Size (percent)".*

2.2.15. 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).

2.2.16. 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.

SQL Server Agent Jobs:

- dbaTDPMon - Discovery & Health Check
  - *daily at 05:00:00*

Note: table `report.htmlSkipRules` can be used to filter out instances or machine names from being included in any of the potential issues rules checks.

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
Collect SQL Agent jobs step details	false	When collecting information on SQL Agent jobs, get also steps execution details (start time, running time, status, etc.)
Collect SQL Errorlog last files	1	Number of errorlog files to be analyzed (default - active one)
Collect Information OS Events	false	Enable or disable collection of information level event messages
Collect Warning OS Events	false	Enable or disable collection of warning level event messages
Collect OS Events timeout (seconds)	600	Timeout for powershell script execution
Collect OS Events from last hours	24	Report OS events from last X hours; default 24
Parallel Execution Jobs	16	Number of data collector parallel jobs; default 4 x CPU count; To disable internal parallelism, set this to 1
Maximum number of retries at failed job	3	When defining a job step, the number of retries to be used in case of step execution failure
Fail master job if any queued job fails	false	Fail the master job if any of the inner/data collecting jobs are failing
History data retention	367	Retention period for long historical saved statistics

3.2 Configuration Thresholds and Options

Table `report.htmlOptions` 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
OS Event Messages Limit to Max	500	Limit OS Event 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 Agent Job - Maximum Running Time (hours)	3	maximum accepted job running time; default 3
OS Event Messages in last hours	24	report OS messages in the last hours
Online Instance Get Databases Size per Project	false	get only project databases size for an instance; default get all dbs

4. Upper Level Stored Procedures

4.1 Add a new SQL Server instance to the inventory

The stored procedure below must be used to manually add a new server to the inventory:

```
[dbo].[usp_refreshMachineCatalogs]          @projectCode            [varchar](32),
                                            @sqlServerName          [sysname],
                                            @debugMode              [bit] = 0
```

Parameter Name	Description
@projectCode	type = varchar The code of the project (a group of servers) on where to add the new SQL Server instance. The code should match an entry in the `dbo.catalogProjects` table
@sqlServerName	type = sysname The instance name to be added to the inventory. If a linked server do not exists, it will be created All new linked servers are created using for the security option: ***Be made using the login's current security context*** If SQL authentication should be used, then manually change the linked server definition using SSMS or SQL code.
@debugMode	type = bit default value = 0 Controls which messages are printed. When enabled (1), dynamic SQL statements that are executed are also printed.

4.2 Remove a SQL Server instance from the inventory

The stored procedure below must be used to manually remove an existing instance from the inventory:

```
[dbo].[usp_removeFromCatalog]               @projectCode            [varchar](32),
                                            @sqlServerName          [sysname],
                                            @debugMode              [bit] = 0
```

Parameter Name	Description
@projectCode	type = varchar The code of the project (a group of servers) on where the SQL Server instance is associated The code should match an entry in the `dbo.catalogProjects` table
@sqlServerName	type = sysname

	The instance name to be removed from the inventory. All existing data for the instance to be removed will be deleted.
@debugMode	type = bit default value = 0 Controls which messages are printed. When enabled (1), dynamic SQL statements that are also printed.

4.3 Manually generate the health-check report

The stored procedure below is used to generate the HTML health-check report:

```
[dbo].[usp_reportHTMLBuildHealthCheck]      @projectCode           [varchar](32)=NULL,
                                            @flgActions            [int]          = 63,
                                            @flgOptions            [int]          = 266338303,
                                            @reportDescription     [nvarchar](256) = NULL,
                                            @reportFileName        [nvarchar](max) = NULL,
                                            @localStoragePath      [nvarchar](260) = NULL,
                                            @dbMailProfileName     [sysname]       = NULL,
                                            @recipientsList        [nvarchar](1024)= NULL,
                                            @sendReportAsAttachment[bit]          = 0
```

Parameter Name	Description
@projectCode	type = varchar The code of the project (a group of servers) on where the SQL Server instance is associated The code should match an entry in the `dbo.catalogProjects` table
@flgActions	type = int Each bit on this parameter value represents an action / report zone to be build: o 1 - Instance Availability o 2 - Databases status o 4 - SQL Server Agent Job status o 8 - Disk Space information o 16 - Errorlog messages o 32 - OS Event messages Default value is to run all (63 = 1 + 2 + 4 + 8 + 16 + 32)
@flgOptions	type = int Each bit on this parameter value represents an option of the report / action: o 1 - Instances - Offline o 2 - Instances - Online o 4 - Databases Status - Issues Detected o 8 - Databases Status - Complete Details o 16 - SQL Server Agent Jobs - Job Failures o 32 - SQL Server Agent Jobs - Permissions errors o 64 - SQL Server Agent Jobs - Complete Details o 128 - Big Size for System Databases - Issues Detected o 256 - Databases Status - Permissions errors o 512 - Databases with Auto Close / Shrink - Issues Detected o 1024 - Big Size for Database Log files - Issues Detected o 2048 - Low Usage of Data Space - Issues Detected o 4096 - Log vs. Data - Allocated Size - Issues Detected o 8192 - Outdated Backup for Databases - Issues Detected o 16384 - Outdated DBCC CHECKDB Databases - Issues Detected o 32768 - High Usage of Log Space - Issues Detected o 65536 - Disk Space Information - Complete Detais o 131072 - Disk Space Information - Permission errors o 262144 - Low Free Disk Space - Issues Detected o 524288 - Errorlog messages - Permission errors o 1048576 - Errorlog messages - Issues Detected o 2097152 - Errorlog messages - Complete Details o 4194304 - Databases with Fixed File(s) Size - Issues Detected o 8388608 - Databases with (Page Verify not CHECKSUM) or (Page Verify is NONE) o 16777216 - Frequently Fragmented Indexes (consider lowering the fill-factor) o 33554432 - SQL Server Agent Jobs - Long Running SQL Agent Jobs o 67108864 - OS Event messages - Permission errors o 134217728 - OS Event messages - Complete Details Default value is all options except for 2097152.
@reportDescription	type = varchar A small description of the report, which will be printed under the report title in the HTML Default value is NULL (no description).

@reportFileName	type = varchar The HTML file name. If not specified, the default file name will be generated as: Daily_HealthCheck_Report_for_{Project Name}_from_{Date and Time}_{Report ID}.html
@localStoragePath	type = varchar A local or UNC path on where to store the HTML report files
@dbMailProfileName	type = sysname A database name profile, to be used for sending emails. Check `msdb.dbo.sysmail_profile` for current server available profile names.
@recipientsList	type = varchar A list of email addresses, semi-colon separated, on which to send the health-check report
@sendReportAsAttachment	type = bit Specify if the HTML file should be sent as attachment or not. Default value is 1 (send file).

c. monitoring

This module performs SQL Server instance monitoring.
For all instances defined as part of a project, there will be monitoring jobs performing specific checks and sending alerts when thresholds or errors are met.
The module should be installed on the same machine which runs the health-check module, as it relies on health-check module collected information.
All alerts event times are in UTC.

1. Enabled agents

1.1 Free Disk Space

Monitor will look for local disk & mounted volumes where databases files exists plus drive C of the machine on which SQL Server instance is running.
Data collected is stored in table `[health-check].[statsDiskSpaceInfo]`.

Default job name: ***dbaTDPMon - Monitoring - Disk Space***
Default schedule: every 15 minutes during 06:30am and 10:30 pm
every 30 minutes during 10:30pm and 06:30 am

Default thresholds (configured within table `monitoring.alertThresholds`)

Name	Sign	Warning Threshold	Critical Threshold
Logical Disk: Free Disk Space (%)	<	8.000	5.000
Logical Disk: Free Disk Space (MB)	<	3000.000	2048.000

For each item which is crossing the threshold value, an email will be send to "Default recipients list - Alerts (semicolon separated)" list from `dbo.appConfiguration` table.

email subject sample:
[PROD] alert on [TESTSERVER]: [critical] - low disk space - E:\MSSQL\Log

- *server class: [PROD]*
- *server name: TESTSERVER*
- *volume with space issue: E:\MSSQL\Log*

email body sample:
- *severity: critical*
- *machine_name: TESTSERVER.prod.local*
- *counter_name: low disk space*
- *target_name: E:\MSSQL\Log*
- *measure_unit: MB*
- *current_value: 10.000*
- *current_percentage: 0.00*
- *refference_value: 204806.938*
- *threshold_value: 2048.000*
- *threshold_percentage: 5.00*
- *event_date_utc: 2016-06-30 21:00:04*

Notes:
- current_value represent current disk space, in MB, in the example above, 10 MB
- refference_value represent current volume size, in MB. In the example above, *204806.938 MB*
- machine name represent the OS host name / cluster node in which the volume is mounted. In the example above, *TESTSERVER.prod.local*

1.2 Replication

Monitor will look for all publications and their subscriptions from SQL Server instances by querying distribution database, if exists.
Data collected is stored in table `[monitoring].[statsReplicationLatency]`.

Default job name: ***dbaTDPMon - Monitoring - Replication***
Default schedule: every 60 minutes during 00:00am and 23:59 pm

1.2.1 Replication Latency
Alert is fired when the trace token latency cannot be determined or is greater than a specified value.

Default thresholds (configured within table `monitoring.alertThresholds`)

Name	Sign	Warning Threshold	Critical Threshold
Replication Latency	>	15.000	20.000

For each item which is crossing the threshold value, an email will be send to "Default recipients list - Alerts (semicolon separated)" list from `dbo.appConfiguration` table.

email subject sample:

[PROD] alert on [TESTSERVER]: [critical] - replication latency - Publication: TestPublication - Subscriber:SUBSERVER.TestDB

- *server class: [PROD]*
- *server name: TESTSERVER*
- *publication: TestPublication*
- *subscriber: SUBSERVER.TestDB*

email body sample:
- *severity: critical*
- *machine_name: TESTSERVER.prod.local*
- *counter_name: replication latency*
- *target_name: Publication: TestPublication / Subscriber: [SUBSERVER].[TestDB] / Publisher: [TESTSERVER].[SourceDB] / Distributor: [DISTSERVER]*
- *measure_unit: sec*
- *current_value: 600*
- *threshold_value: 600*
- *event_date_utc: 2016-01-30 21:13:12*

1.2.2 Subscription not active
Alert is fired when a subscription was not initialized.

email subject sample:
[PROD] alert on [TESTSERVER]: [SourceDB] - [error] - subscription not active - SUBSERVER

- *server class: [PROD]*
- *server name: TESTSERVER*
- *publisher: SourceDB*
- *subscriber: SUBSERVER*

email body sample:
- *error number: 21488*
- *The subscription is not active. Subscription must have active in order to post a tracer token.*
- *Publication: [TestPublication] / Subscriber [SUBSERVER].[TestDB] / Publisher: [TESTSERVER].[SourceDB] / Distributor: [DISTSERVER] / Articles: 1*
- *duration: 4 seconds*
- *event-date (utc): 2016-06-02 19:00:05*

 danandrei.stefan@gmail.com

1.2.3 Subscription marked inactive

Alert is fired when subscription is marked inactive.

email subject sample:

[PROD] alert on [TESTSERVER]: [SourceDB] - [error] - subscription marked inactive - SUBSERVER

- *server class: [PROD]*
- *server name: TESTSERVER*
- *publisher: SourceDB*
- *subscriber: SUBSERVER*

email body sample:

- *error number: 21074*
- *The subscription(s) have been marked inactive and must be reinitialized.*
- *Publication: [TestPublication] / Subscriber [SUBSERVER].[TestDB] / Publisher: [TESTSERVER].[SourceDB] / Distributor: [DISTSERVER] / Articles: 1*
- *duration: 3 seconds*
- *event-date (utc): 2016-05-16*

1.3 Failed SQL Agent jobs

Monitor will look for instances and what SQL Server Agent jobs had been falling since the last job execution. For these, an alert will be sent, along with job log, if file can be accessed and attached.
Data collected is stored in table `[monitoring].[statsSQLAgentJobs]`.

Default job name: ***dbaTDPMon - Monitoring - SQLAgentFailedJobs***
Default schedule: every 10 minutes during 00:00am and 23:59 pm

email subject sample:

[PROD] job status on [TESTSERVER]: [failed] - sql agent job status - Agent history clean up: distribution

- *server class: [PROD]*
- *server name: TESTSERVER*
- *failing job:* Agent history clean up: distribution

email body sample:

Step ID	Step Name	Run Status	Run Date	Run Time	Run Duration	Message
1	Run agent.	Failed	2016-05-31	17:10:00	00h 00m 47s	Executed as user: .\SQLAgent. Transaction (Process ID 164) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction. [SQLSTATE 40001] (Error 1205). The step failed.

1.4 Transaction Status

Monitor will look for all active SQL Server instances and will raise alerts for long running, uncommitted or blocked transactions or for sessions using a high amount of tempdb space.
Data collected is stored in table `[monitoring].[statsTransactionsStatus]`.

Default job name: ***dbaTDPMon - Monitoring - TransactionStatus***
Default schedule: every 10 minutes during 00:00am and 23:59 pm

1.4.1 Long Running Transactions

Default thresholds (configured within table `monitoring.alertThresholds`)

Name	Sign	Warning Threshold	Critical Threshold
Running Transaction Elapsed Time (sec)	>	1800.000	3600.000

For each item which is crossing the threshold value, an email will be send to "Default recipients list - Alerts (semicolon separated)" list from `dbo.appConfiguration` table.

email subject sample:

[PROD] alert on [TESTSERVER]: [SourceDB, tempdb] - running transaction - critical

- *server class: PROD*
- *server name: TESTSERVER*
- *databases involved in the blocking scenario: SourceDB, tempdb*

email body sample:

- *severity: critical*
- *instance_name: TESTSERVER*
- *counter_name: running transaction*
- *session_id: 121*
- *is_session_blocked: No*
- *sessions_blocked: 1*
- *databases: SourceDB, tempdb*
- *host_name: SOMEONE-PC*
- *program_name: .Net SqlClient Data Provider*
- *login_name: PROD\SomeDeveloper*
- *sql_handle: 0x03001500691E317AC56D030056A5000000000000*
- *transaction_begin_time: 2016-01-31 02:39:09*
- *last_request_elapsed_time: 00:00:00.000*
- *transaction_elapsed_time: 01:00:33.000*
- *threshold_value: 01:00:00.000*
- *measure_unit: sec*
- *event_date_utc: 2016-01-31 08:40:02*

Notes:

- *Session 121 is running for 1 hour (in current example).*
- *Host_name, program_name and login_name will identify who started the session.*
- *Databases will identify the client against the SQL is running.*

1.4.2 Uncommitted Transactions

Default thresholds (configured within table `monitoring.alertThresholds`)

Name	Sign	Warning Threshold	Critical Threshold
Uncommitted Transaction Elapsed Time (sec)	>	900.000	1800.000

For each item which is crossing the threshold value, an email will be send to "Default recipients list - Alerts (semicolon separated)" list from `dbo.appConfiguration` table.

email subject sample:

[PROD] alert on [TESTSERVER]: [SourceDB, tempdb] - uncommitted transaction - warning

- *server class: PROD*
- *server name: TESTSERVER*
- *databases involved in the blocking scenario: SourceDB, tempdb*

email body sample:

- *severity: warning*
- *instance_name: TESTSERVER*
- *counter_name: uncommitted transaction*
- *session_id: 56*
- *is_session_blocked: No*
- *sessions_blocked: 0*
- *databases: SourceDB, tempdb*
- *host_name: SOMEONE-PC*
- *program_name: .Net SqlClient Data Provider*
- *login_name: PROD\SomeDeveloper*
- *sql_handle: 0x01000200CF0D87278053B6C00E00000000000000*
- *transaction_begin_time: 2016-01-29 05:13:34*
- *last_request_elapsed_time: 00:00:00.000*
- *transaction_elapsed_time: 00:16:17.000*
- *threshold_value: 00:15:00.000*
- *measure_unit: sec*
- *event_date_utc: 2016-01-29 10:30:03*

Notes:

- *Session 56 is uncommitted for transaction_elapsed_time - last_request_elapsed_time time (16 minutes in current example)*
- *Host_name, program_name and login_name will identify who started the session.*
- *Databases will identify the client against the SQL is running.*

1.4.3 Blocked Transactions

Default thresholds (configured within table `monitoring.alertThresholds`)

Name	Sign	Warning Threshold	Critical Threshold
Blocking Transaction Elapsed Time (sec)	>	600.000	900.000

For each item which is crossing the threshold value, an email will be send to "Default recipients list - Alerts (semicolon separated)" list from `dbo.appConfiguration` table.

email subject sample:

[PROD] alert on [TESTSERVER]: [SourceDB] - blocked transaction - critical

- *server class: PROD*
- *server name: TESTSERVER*
- *databases involved in the blocking scenario: SourceDB*

email body sample:

- *severity: critical*
- *instance_name: TESTSERVER*
- *counter_name: blocked transaction*
- *session_id: 124*
- *is_session_blocked: Yes*
- *sessions_blocked: 1*
- *databases: SourceDB*
- *host_name: SOMEONE-PC*
- *program_name: .Net SqlClient Data Provider*
- *login_name: PROD\SomeDeveloper*
- *sql_handle: 0x03001500EFC10D630F56030056A5000001000000*
- *transaction_begin_time: 2016-01-31 02:39:14*
- *last_request_elapsed_time: 00:00:00.000*
- *transaction_elapsed_time: 01:30:29.000*
- *threshold_value: 00:15:00.000*

- *measure_unit: sec*
- *event_date_utc: 2016-01-31 09:10:02*

Notes:

- *Session 124 is blocking 1 (sessions blocked) other session(s).*
- *Session 124 is currently blocked by other session*
- *Host_name, program_name and login_name will identify who started the session.*
- *Databases will identify the client against the SQL is running.*

1.4.4 Sessions consuming tempdb space

Default thresholds (configured within table `monitoring.alertThresholds`)

Name	Sign	Warning Threshold	Critical Threshold
tempdb: space used by a single session (mb)	>	8192.000	16384.000

For each item which is crossing the threshold value, an email will be send to "Default recipients list - Alerts (semicolon separated)" list from `dbo.appConfiguration` table.

email subject sample:

[PROD] alert on [TESTSERVER]: [session_id=123] - tempdb space - warning

- *server class: PROD*
- *server name: TESTSERVER*
- *SQL session who is keeping tempdb space: 123*

email body sample:

- *severity: warning*
- *instance_name: TESTSERVER*
- *counter_name: tempdb space*
- *session_id: 123*
- *is_session_blocked: No*
- *sessions_blocked: 0*
- *databases: SourceDB, tempdb*
- *host_name: SOMEONE-PC*
- *program_name: .Net SqlClient Data Provider*
- *login_name: PROD\SomeDeveloper*
- *sql_handle: 0x0300120008C2AB6E43ED870093A5000001000000*
- *transaction_begin_time: 2016-01-30 20:34:05*
- *transaction_elapsed_time: 00:15:56.000*
- *tempdb_usage: 12345*
- *measure_unit: mb*
- *event_date_utc: 2016-01-31 02:50:02*

Notes:

- *Host_name, program_name and login_name will identify who started the session.*
- *Databases will identify the client against the SQL is running.*

2. Ignoring alerts / mark as skip

If an alert has to be ignored, as it is something "normal" or cannot be fixed and it is just polluting the inbox, than it can be marked as skip.
Table `[monitoring].[alertSkipRules]` contains all items that should be ignored.

Table contains by default, some entries to be used as an example.

Category	Alert Name	Skip Value	Skip Value (2)	Active
disk-space	Logical Disk: Free Disk Space (%)			0
disk-space	Logical Disk: Free Disk Space (MB)			0
replication	subscription marked inactive	[PublisherServer].[PublishedDB](PublicationName)	[SubscriberServer].[SubscriberDB]	0
replication	subscription not active	[PublisherServer].[PublishedDB](PublicationName)	[SubscriberServer].[SubscriberDB]	0
replication	replication latency	[PublisherServer].[PublishedDB](PublicationName)	[SubscriberServer].[SubscriberDB]	0
performance	Running Transaction Elapsed Time (sec)	InstanceName		0
performance	Uncommitted Transaction Elapsed Time (sec)	InstanceName		0
performance	Blocking Transaction Elapsed Time (sec)	InstanceName		0
performance	tempdb: space used by a single session	InstanceName		0

 danandrei.stefan@gmail.com

III. Extended / Advanced Options

1. Parallel Database Maintenance

Under construction

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

Name	value	Description
Parallel Execution Jobs	16	Number of data collector parallel jobs; default 4 x CPU count; To disable internal parallelism, set this to 1
Maximum number of retries at failed job	3	When defining a job step, the number of retries to be used in case of step execution failure
Fail master job if any queued job fails	false	Fail the master job if any of the inner/data collecting jobs are failing
Default folder for logs	NULL	Default folder where to store SQL Agent job logs, created at run-time
Internal jobs log retention (days)	367	Retention period for internal jobs, in history tables
Maximum SQL Agent jobs started per minute (KB306457)	60	Restrict the number of SQL Agent jobs to be started in a minute
In "serial" mode (parallel=1), execute tasks using SQL Agent jobs	0	disable / enable use of SQL Agent jobs when running single threaded
Maximum SQL Agent jobs running (0=unlimited)	0	Upper cap for the maximum number of internal jobs to be executed across all projects / tasks in the same time
Use Default Scheduler for maintenance tasks if project specific not defined	1	If a project do not have a scheduler associated, use the default one. If a project do not have a scheduler, it will not start executing tasks.
Maximum job queue execution time (hours) (0=unlimited)	1	Maximum time, in hours, since a job was created in the queue and it was selected to be executed
Maximum SQL Agent jobs running on the same physical volume (0=unlimited)	1	Upper cap for the maximum number of internal jobs to be executed across all projects / tasks in the same time, on the same physical mount point (only if health-check module is installed)

 danandrei.stefan@gmail.com