Abstract

This document contains information related to SQL Server 2005 and SQL Server 2008 and is a compilation of research from various sources on the Internet and from first-hand Data Base Administrator (DBA) experience describing various aspects of configuring SQL Server maintenance plans. Portions of this document describing specific SQL Server settings are subject to change without notice.
The information contained in this document represents the current view of BID2WIN Software, Inc. on the issues discussed as of the date of publication. Because BID2WIN Software must respond to changing market conditions, it should not be interpreted to be a commitment on the part of BID2WIN Software, and BID2WIN Software cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. BID2WIN SOFTWARE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of BID2WIN Software Incorporated.

BID2WIN Software may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from BID2WIN Software, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.

© 2010 BID2WIN Software, Incorporated. All rights reserved.

BID2WIN Software, Inc. and BUILD2WIN are either registered trademarks or trademarks of BID2WIN Software Incorporated in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Version 1.0
# Contents

What is a Maintenance Plan? ........................................................................................................... 1
How are Maintenance Plans Executed? ............................................................................................. 1
How are Maintenance Plans Created? ............................................................................................. 1
What can I do with a Maintenance Plan? .......................................................................................... 1
  Check Database Integrity ................................................................................................................... 1
  Update Statistics .............................................................................................................................. 2
  Rebuild Index .................................................................................................................................. 2
  Reorganize Index ............................................................................................................................. 3
  Shrink Database .............................................................................................................................. 5
  Backup Database ............................................................................................................................ 6
Conclusion ........................................................................................................................................... 9
References ........................................................................................................................................ 9
**What is a Maintenance Plan?**
A maintenance plan is a set of steps performed in a specific order on a database to keep it healthy. The tasks carried out by the maintenance plan are necessary to maintain good database performance and may also be used to perform backup operations. Every SQL Server database should have at least one maintenance plan associated with it, but a single plan can be created to perform the same tasks on a number of databases. A general rule of thumb is ‘the larger the database, the more critical it is to have a maintenance plan.’

**How are Maintenance Plans Executed?**
Maintenance plans can be executed on-demand, but usually they are scheduled to run automatically on a given interval or at a specific time of day. You might expect that these plans would be scheduled using the familiar Windows Task Scheduler, which is used to run commands, batch routines and other general Windows jobs. Instead, SQL Server has its own task handling service called the SQL Server Agent.

The SQL Server Agent is a special Windows service for SQL Server which can do more than just run maintenance plans. For example, BUILD2WIN Field Tracking takes advantage of the SQL Server Agent to execute SQL Scripts to make the process of synchronizing mobile workstations much faster.

**How are Maintenance Plans Created?**
Maintenance plans are usually created using the Maintenance Plan Wizard, accessible from the Management node of a SQL Server instance within SQL Server Management Studio. SQL Server also allows you to create maintenance plans using more advanced interactive tools from the Management node. Instructions for using the Maintenance Plan Wizard are beyond the scope of this document, but there are numerous resources available on the Internet describing how to use this feature of SQL Server. This document will concentrate on what the maintenance plan ought to do and why.

**Important Notes:**
- You will need to connect to your SQL Server instance with the SQL SA account
- The Management node and Maintenance Plan Wizard is not provided with SQL Server Express

**What can I do with a Maintenance Plan?**
The following sections describe the tasks that can be included in a maintenance plan along with the reason for including each task.

**Check Database Integrity**
The Check Database Integrity task performs a variety of tests, such as verifying allocation of all database pages, confirming structural integrity of tables and indexed views, and checking consistency between system metadata tables. Any issues discovered through these actions would need to be subsequently addressed through proper administrative actions.

**Warning:** Although it is possible to verify integrity of multiple databases within the same task, it is not recommended to do so with any version of SQL Server 2005 SP2. For more information, please see Microsoft Knowledge Base articles 934458 and 934459.

**Ignore databases where the state is not online**
This setting determines if the Check Integrity maintenance task will be executed if the target database is not currently in a normal online state. Examples of possible database states include; Online, Restoring, Recovering, Recovery_Pending.
Suspect, Emergency and Offline. Generally, if the database is not in the normal online state, a maintenance task should not be executed on it. BID2WIN Support Department recommends always selecting this setting.

Include Indexes

Recommend setting is Yes. If the maintenance plan will rebuild indexes (discussed below), then it is not necessary to check their integrity, and this setting can be unselected to exclude indexes from the integrity checking task. However, if you are setting up separate plans for integrity checking vs. routine re-indexing, you may wish to include indexes. If you are unsure which is appropriate, selecting this setting will not cause harm.

Attempt to repair any minor problems

Recommended setting is No. This option does not exist in SQL Server 2008. It is equivalent to using the SQL DBCC command with the REPAIR_FAST option which, when run manually requires that the database be in single-user mode. If you are using SQL Server 2005, it is recommended to leave this setting unchecked.

Update Statistics

The Update Statistics Task recalculates information that represents data distribution in tables and indexes, which is then used by the query optimizer to determine the most efficient approach when processing queries. The recalculation process can be based on two types, full scan or a sample-based scan. Full scan may take a considerable amount of time with larger objects, but it provides the best accuracy. A sample-based saves processing time necessary to generate statistical data, but has the risk of potential inaccuracies. Starting with SQL Server 2005 SP2, an updated task editor interface gives you the ability to specify scan type. This provides a choice between a full scan and a sample-based scan, with its size determined by either a percentage of the entire data set or a number of rows.

Note: All tables in the BUILD2WIN database are configured with the STATISTICS_NORECOMPUTE option set to ‘OFF’, which tells SQL Server to attempt to maintain the statistics for the table as changes are made. However, BID2WIN has determined that using the Update Statistics task can significantly improve performance of a BID2WIN or BUILD2WIN database.

Update: All existing statistics, Column statistics only, Index statistics only

The Update setting offers three choices. Although you can choose to update one or the other, a large database should have this set to “All existing statistics”.

Scan Type: Full scan or Sample by

The Scan Type section allows you to update statistics based on a full scan or by entering a Sample By value. The Sample By value can be either a percentage or a specific number of rows. The default is to use the full scan which is naturally more thorough and may improve the resiliency of a particularly large database in terms of the rate of performance degradation. BID2WIN Support recommends using the Full Scan option.

Rebuild Index

The Rebuild Index Task is used to recreate indexes to improve performance of index scans and seeks. This task also applies a specific amount of free space on the leaf-level pages according to the value of its Fill Factor parameter, compacting and reordering them in the process. This task also decreases index fragmentation and lowers the amount of
page splits during subsequent data inserts/modifications. By optimizing the distribution of data and free space on the index pages, the database is also prepared for faster future growth. It is recommended to use this option for the most significant gains in performance.

**Note:** If you have to run the Rebuild Index Task, do not run the Update Statistics task afterward. It is a redundant step and can degrade database performance.

**Free Space Options**

The Free Space Options section provides additional setting to configure the Rebuild Index task.

**Reorganize Pages with the Default Amount of Free Space**

This setting will drop the indexes on the database tables and re-create them with the fill factor that was specified when the indexes were created. This is the default setting and BID2WIN Support Department recommends leaving the free space options setting toggled to reorganize pages with the default amount of free space.

**Change Free Space per Page Percentage To**

This setting will drop the indexes on the database tables and re-create them with a new, automatically calculated fill factor, thereby reserving the specified amount of free space on the index pages. The higher the percentage, the more free space is reserved on the index pages, and the larger the index grows. Valid values are from 0 through 100. BID2WIN Support Department does not recommend toggling the free space options setting to this choice.

**Sort Results in temp db**

This setting determines where the intermediate sort results – generated during index creation – are temporarily stored. If a sort operation is not required, or if the sort can be performed in memory, this setting is ignored. Note that this is used only during the creation of the index and has no effect once the index has been created. BID2WIN Support Department does not recommend selecting this setting.

**Keep Index Online While Re-Indexing**

This setting allows users to access the underlying table, clustered index data, and the associated indexes during the index rebuild operation. BID2WIN Support Department recommends selecting this setting.

**Pad index**

The Pad index setting was removed with SQL Server 2005 Service Pack 2. It controls the amount of free space in intermediate-level pages of the index that is being rebuilt based on the value of the fill factor parameter. BID2WIN Support Department does not recommend using this option if it exists in your version of SQL Server.

**Ignore duplicate keys**

The Ignore duplicate keys setting was removed with SQL Server 2005 Service Pack 2. It dictates behavior triggered by an attempt to add a duplicate into a table with a unique index, as part of a multi-row transaction. With this option enabled, such a scenario will generate a warning but permit non-violating rows to be inserted. Keeping it disabled (which is the default) will result in an error and will rollback of the entire INSERT transaction. BID2WIN Support Department does not recommend using this option if it exists in your version of SQL Server.

**Reorganize Index**

The Reorganize Index Task provides a less intrusive, but also less powerful alternative to rebuilding an index. It allows you to perform online reordering and compacting of index leaf-level pages without reapplying their fill factor parameter.
This ensures defragmentation which results in improved performance. In general, Microsoft recommends the use of this task in situations when the index fragmentation level is below 30%. In most cases, BID2WIN Support Department recommends using the Rebuild Indexes task (see above) rather than Reorganize Index. However, in the case of larger databases, it may be necessary to opt for multiple plans where reorganizing indexes is performed in the more frequent plan and rebuilding the indexes is performed in a plan scheduled to run less frequently.

**Compact Large Objects**

This option specifies that all pages containing large object (LOB) data are compacted. Compacting large objects can improve disk space use. This option is selected by default. If the Reorganize task is used, BID2WIN Support Department recommends leaving this option selected.

1 For more information, please refer to the Microsoft TechNet web page at the following address: http://technet.microsoft.com/en-us/library/ms189858.aspx.

2 LOB data types are image, text, ntext, varchar(max), nvarchar(max), varbinary(max), and xml.
Shrink Database

The Shrink Database Task is typically used to reduce the size of files containing a target database and corresponding log files, which is accomplished by compacting its content and lowering the amount of free space they contain. The effectiveness of this task depends on having a sufficient amount of free space in the database as well as other factors such as minimum database size. In a nutshell, this task rearranges allocated pages of a file, moving their content to unallocated pages in the beginning of the file to make them contiguous. BID2WIN Support Department does not recommend using this task.

Warning: When creating maintenance plans, it is a best practice to not select the option to shrink the database for the following reasons:

- When shrinking the database, SQL Server moves pages toward the beginning of the file, allowing the tail end of the files to be shrunk. This process can increase the transaction log size because all moves are logged.
- If the database is heavily used and there are many inserts, the database files will have to grow again. This will typically cause performance degradation. SQL Server 2005 addresses slow auto-growth with instant file initialization; therefore, the growth process is not as slow as it was in the past. However, at times auto-grow does not catch up with the space requirements, causing performance degradation.
- Constantly shrinking and growing of the database leads to excessive database fragmentation. If you need to shrink the database size, it should be done manually when the server is not being heavily utilized, and by specifying that sorting takes place in the tempdb database.

Note: For the same reasons stated above, the Auto-Shrink feature of SQL Server should never be enabled.
Backup Database

This task allows you to specify the source databases, destination files or tapes, and overwrite options for a full backup. Backing up a database can be accomplished by three different ways:

Full Backup

With this strategy, the entire database is backed up. In the event of a failure, all the committed transactions that occurred after the most recent database backup are lost. The primary advantage of using only complete database backups is simplicity. Backing up is a single operation and normally scheduled at regular intervals. Should a restore be necessary, it can be accomplished easily in one step.

Use full backups if:

- The database is small. The amount of time required to backup a small database is reasonable.
- The database has few data modifications
- You are willing to accept the loss of changed data if the database fails between backups and must be restored to its previous state

Differential Backup

This strategy is used to augment either the database backup strategy or a database and transaction log backup strategy. Differential backups consist of only those portions of the database that have changed since the last database backup. The first stage in this strategy is to always take a complete database backup. You can then schedule the transaction log backups as usual. The significant different is that from that point on, instead of scheduling a complete database backup, you would schedule a differential backup after the day's transaction log backups. The differential backup strategy combined with the transaction log backup strategy reduces the number of transaction log backups that need to be restored while rebuilding/recreating a database.

Use differential backup strategy if:

- The amount of time spent in recovering the database by applying all the transaction logs is not acceptable

Transaction Log Backup

With this strategy, the entire database is backed up less frequently while the transaction log is backed up more frequently between full database backups. In case of failure, you will be able to recover all backed-up transactions and possibly even committed (complete) transactions that occurred since the last transaction log backup. Only uncommitted (incomplete) transactions will be lost.

Use database and transaction log backups if:

- The database is considerably large or expected to grow large in the near future
- There are substantial and frequent updates/data modifications taking place on the database
- In case of a disaster, the need is to recover the database to as recent a state as possible and avoid losing any transactions that have taken place on it
- You cannot afford to lose changes since the most recent database backup
**Maintenance Cleanup**

The Maintenance Cleanup task is frequently paired up with the Backup Database task in order to delete outdated backup files and maintenance plan reports.
History Cleanup

The History Cleanup Task serves a role similar to the Maintenance Cleanup task, although instead of dealing with files created by maintenance plans, it removes outdated entries from a number of tables in the msdb database (backupfile, backupfilegroup, backupmediafamily, backupmediaset, backupset, restorefile, restorefilegroup, and restorehistory). Note that these entries might be generated not only by maintenance plans, but can also appear as the result of performing standard backups and restores, or executing SQL Server Agent jobs. The scope of cleanup is determined by the content of "Remove historical data older than" property which, starting with SQL Server 2005 SP2, can have its value expressed in hours, in addition to days, weeks, months, and years.

Execute SQL Server Agent Job

The Execute SQL Server Agent Job task allows you to select SQL Server Agent jobs to run as part of the maintenance plan. Simply pick the ones you are interested in directly from the task editor dialog box. This is a very powerful feature which allows you to extend the functionality of your maintenance plan with custom scripts.
Conclusion

Maintenance plans are a very powerful toolset for ensuring optimal performance and data security for SQL Server databases.

References

SQL Server 2005 Integration Services

Inside SQL Server Maintenance Plans

SQL Server 2005 Maintenance Practices

Why you should not shrink your data files

Creating a SQL Server Maintenance Plan

SQL Server administration best practices
http://vyaskn.tripod.com/sql_server_administration_best_practices.htm