OPTIMIZING INDEXES (ONLINE MODE) ON SQL SERVER 2005

On large database systems, with large numbers of insert and update commands, the problem of index fragmentation is one of the main causes of performance degradation and a proper index optimization strategy is a must.

I see every day that many DBAs plans the index optimization via a custom **T-SQL** script or via the standard **SQL Maintenance Plan**, but they don't know that actually SQL Server 2005 permits you to "tune" this process.
SQL Server 2005 provides an option (**ONLINE = ON** or **OFF**) to help *manage the performance and concurrency requirements* while creating or rebuilding an index. With the new Online Indexing feature (ONLINE=ON) you can continue to make queries and operations on the underlying table durig the index rebuild, while the Offline indexing (ONLINE=OFF) blocks the table.

There's an important thing to remember: **Online** index create or rebuild (ONLINE=ON) provides maximum concurrency, but uses more resources and takes longer to complete!!

To help control how **SQL Server** manages the temporary space during an index operation, SQL Server provides another option: **SORT_IN_TEMPDB**. SQL Server uses temporary storage for sorting and other intermediate tasks while creating or rebuilding an index. This temporary storage can be used from the user database, or it can be used from the TEMPDB database.

When the **SORT_IN_TEMP** option of the **CREATE INDEX** or **ALTER INDEX** statement is set to **OFF** (*the default*), the temporary storage is used from the user database. When **SORT_IN_TEMP** is set to ON, the temporary storage is used from the **TEMPDB** database. These are the recommendations for a better Index create/rebuild strategy (directly from Microsoft). I recommend to print it: Make sure that the **TEMPDB** is on a disk subsystem that provides sufficient I/O throughput, and that the **TEMPDB** is big enough to accommodate the temporary space that is required for the index create or rebuild operation.
By default, the **TEMPDB** is created in the **Data** directory under the SQL Server installation folder (for example, C:\SQL2005\MSSQL.1\MSSQL\Data). In this configuration, there might not be enough space for the **TEMPDB**, and the storage might not have adequate **I/O** throughput.

Therefore, it is a best practice to move the TEMPDB to a storage area with sufficient space and performance after you install SQL Server.

Also, note that the TEMPDB database is a common resource for the entire instance of SQL Server. You should consider the activities in all the user databases that might be using TEMPDB while you plan for the TEMPDB.

- To achieve the least time to create or rebuild an index, use the offline option (ONLINE=OFF). However, this prevents all user *access to the underlying table* for the duration of the index create or rebuild.

- To achieve the least effect on other users accessing the table, use the online option (ONLINE=ON). However, an online operation takes more time and uses more **TEMPDB** space as compared to an offline operation.

- To use the least amount of space in TEMPDB while you rebuild a clustered index, use the offline (ONLINE=OFF) option. However, this affects concurrency because access to the table is prevented for the duration of the index rebuild.

- To use the least amount of space in TEMPDB while you rebuild a non-clustered index, use the online (ONLINE=ON) option. The online rebuild also provides the best concurrency, but takes longer to complete.

- If there are transactions on the table that are concurrent with the online index create or rebuild, you need to plan for additional space in **TEMPDB** for the version store.