

Index Fragmentation

If you have arrived at this page as a result of running the [Server Health](#) report, congratulations you are on the first step to improving your server health.

What causes Index Fragmentation

Over time as data is inserted, deleted, or updated in a table, the indexes associated with that table become fragmented. Another cause is running [DBCC SHRINKDATABASE](#) which I covered in a [previous post](#).

What does this mean to my SQL Server

A fragmented index is bad for a number of reasons:

- It takes up more memory on your SQL Server.
- It takes up more disk space on the SQL Server.
- The index does not perform as fast as you might expect. There may be more logical reads to get to your data.
- Your backup files will be larger than needed.

What Not To Do

The logical assumption would be to just add a maintenance task. The problem with this is that you will be reorganizing or rebuilding all the indexes in your database. This will have a significant performance impact, and it may fill up your disk space by flooding your transaction logs with all the changes.

How Do I Fix It

The right thing to do depends on whether you are using Standard or Enterprise Edition of SQL Server.

Recommendation from Microsoft

> 5% and < = 30%	ALTER INDEX REORGANIZE
> 30%	ALTER INDEX REBUILD WITH (ONLINE = ON)*

If you are using SQL Server Standard Edition the ALTER INDEX REBUILD WITH (ONLINE = ON) is not available. This is only available in Enterprise Edition. In Standard Edition if you want to rebuild the index, it ends up rebuilding it in offline mode, which could have a serious impact on performance. My recommendation if you are using Standard Edition is to always REORGANIZE, rather than REBUILD.

Remember if you are doing an REORGANIZE of the index that you will need to regenerate statistics after the REORGANIZE. If you rebuild, statistics will be regenerated for you.