

Keeping Your SQL Server Databases Defragmented with Diskeeper

*by Howard Butler Sr., Systems Engineer
and Michael Materie, Systems Engineer MCSE, CCNA, A+, I-Net+*

All SQL Server databases, over time, experience "internal" fragmentation of its data. This occurs when records are removed from database pages, but the space it occupied is still there after deletion. Eventually this space is reused, but as it is reused, the data pages become physically fragmented, which can lead to unnecessary I/O, especially in case of table scans where many data pages are read, one after another.

In SQL Server, there are several ways to defrag internal fragmentation. One of these methods is to use the DBCC REINDEX command to rebuild clustered and non-clustered indexes. Once indexes are rebuilt, data pages are now logically contiguous, and disk I/O is minimized.

Unfortunately, internal fragmentation is only part of the fragmentation problem. When DBCC REINDEX is run, it does nothing about "external" fragmentation. External fragmentation refers to the fragmentation of physical files on your server's disks, which can cause as much, if not more, unnecessary I/O activity as internal fragmentation. Unnecessary I/O activity, as you would expect, hurts SQL Server's overall performance.

SQL Server databases are made up of large database and log files that are pre-allocated in size at the point of their creation. If there is enough contiguous empty space on disk when the original files are created, then they will not be fragmented. But if the empty space available is not continuous, then these original database and log files will be fragmented.

Even if the original database and log files are not fragmented when they are first created, they will almost certainly become fragmented as the database grows over time. For example, if you set the original database size to 100MB and the log to 10MB, and you have them set to grow automatically, and if eventually the database grows to 5GB in size and the log grows to 100MB in size, external fragmentation could become great. Every time the database or log files grow automatically, there is the potential for external fragmentation.

To defrag external fragmentation takes an operating system utility, not a SQL Server utility. One of the most popular tools for defragmenting SQL Server database files is a tool from Executive Software called Diskeeper. Diskeeper has been around for many years, and many of you may already be familiar with it, at least as how it is used for Windows file and print servers. What many DBAs aren't familiar with is that it is probably the best tool for defragmenting external fragmentation on their SQL Servers.

When an external fragmentation tool like Diskkeeper runs, it does not restructure the internal contents of the file, unlike DBCC REINDEX. After Diskkeeper defragments a file, the defragmented file will be a bit for bit duplicate of the original. Therefore, any holes within the database are still present and you will still need, from time to time, to rebuild your indexes to combat internal fragmentation.

There are two types of external fragmentation with which a utility like Diskkeeper deals with: file fragmentation and free space fragmentation. File fragmentation concerns computer disk files that are not whole, but rather are broken into scattered parts; while free space fragmentation means that the empty space on a disk is broken into scattered parts rather than being collected all in one big empty space. File fragmentation causes problems with accessing data stored in computer disk files, while free space fragmentation may cause problems creating new data files or extending (adding to) old ones.

So when Diskkeeper runs, it acts to defrag database and logs files so that instead of being made of up of many pieces, the file is one continuous physical file. In addition, Diskkeeper defrags free space so that when database or log files expand, that they can expand with little or no fragmentation. But this does not last forever. Eventually, fragmentation becomes a problem, and the database and log files need to be defragged again. Ideally, defragging should be performed on a regular basis to prevent unnecessary fragmentation from occurring.

Now here is something you probably have not thought of before. What effect does physical file fragmentation have on rebuilding your SQL Server indexes? In other words, if you don't perform a physical defragmentation, but you perform an internal fragmentation, are there any downsides to this?

Yes, there can be. Because the physical files are fragmented, it will take SQL Server much more time, and I/O, to rebuild its indexes on fragmented physical files than it does on contiguous physical files. This means that before you perform an internal defragmentation process that you might want to perform a physical defragmentation process first. This way, you reduce how long it takes to rebuild your indexes, and you also reduce the I/O load on your server during the index rebuilding process.

While SQL Server database and log file physical fragmentation can have a negative affect on SQL Server's performance, don't forget that there are other files that SQL Server accesses, such as the SQL Server executables, and if you are using Full-Text Indexing, the full-text index files. So not only do you want to defrag SQL Server database and log files, but all files located on your SQL Server.

All file movement in a Diskkeeper defragmentation job is handled by the operating system itself. In fact, the code in the operating system, which was originally written by Executive Software, prioritizes safety in determining what can be defragmented and what cannot. SQL Servers databases (e.g. .LDF, .MDF) are perfectly safe to defragment. As Diskkeeper sends requests the operating system (through an API) to move files, if it comes across files that cannot be safely moved, they are simply skipped over without any error or concern.

So how do you find out how if the physical files on your SQL Server are fragmented? Fortunately, this is easy. As part of Diskkeeper's functionality, you can run a

fragmentation analysis to see just how fragmented your SQL Server files are. As with defragmentation, this can be done while SQL is running.

As you can imagine it is hard to recommend a specific schedule, as each database is different and fragmentation occurs at different rates. A dynamic Diskeeper scheduler called Smart Scheduling determines and automates defragmentation jobs as appropriate. And yes, you can still restrict Diskeeper "Smart" runtimes and resource usage.

So, it's plain to see that a defragmentation utility like Diskeeper can take care of the external disk fragmentation, while a SQL Server utility like DBCC REINDEX can address internal SQL Server disk fragmentation. As a team, they can work together to help ensure the optimum performance of your SQL Servers.

If there is any doubt, simply install Diskeeper and use its analysis function to find out exactly how many individual pieces your files are broken into. I'm quite sure that you will be very surprised. I've seen reports from sites where their database files were in more than 287,000 pieces. OUCH!!!!