

## Updating SQL Server Statistics

Updating SQL Server statistics may not be as obvious as it may sound.

I

Imagine this scenario. I invite you over for dinner and a game of Scrabble. After dinner, I start to do the dishes, while you set up the game. Rather than just doing the dishes that are dirty from our dinner, I decide to take all the dishes from the shelves, and drawers in my kitchen. I create a gigantic pile of all of the dishes, both clean and dirty on the kitchen counter. You get the game setup, and we start playing the game. Between turns, I jump back into the kitchen, and wash a few dishes, then on my turn, I jump back into the game and play my turn, then back to the kitchen again. During the game you are constantly waiting for me to jump back into the game to make my move. You could see how long this would draw out the game with me washing all the dishes, even the ones that were already clean. 4 hours later, I finally finish the dishes, and shortly thereafter we finish the game. It is unlikely, even if you love to play Scrabble that you would ever return to my house for dinner and a game again. Tomorrow night, when I have dinner with my family, I do the same thing, and every night after dinner I wash all the dishes in the kitchen, even the clean ones. You can see how wasteful this sounds, and perhaps some people would even label me as OCD or something worse in this scenario.

Although this is obviously the wrong way to do things, often times we end up doing the equivalent of this in our SQL Server maintenance tasks.

### Jump to Updating SQL Server Statistics

## Steve Stedman

Freelance SQL Server Consultant

<http://stevestedman.com>

~~SQL Server Statistics are made up of a set of information that is used by the SQL Server engine to help assess how indexes are going to be used when a query is run. This set of information is~~  
known as a histogram. The histogram has details on the distribution of data in an index, and depending on what a query is asking for the statistics are used to determine if the index is the best option, and whether an index scan or index seek can be used. If statistics are out of date, SQL Server can make the wrong choices which leads to bad performance.

Once statistics are generated, they will be used by the query optimizer. If you have changes to a significant number of rows in your table, then the statistics are not accurate and can lead to bad decisions by the query optimizer. For tables that are adding or changing a significant number of rows then the statistics should be updated frequently. For tables that have a small percentage of the rows being added or changed regularly then the statistics need to be updated less often.

The problem with updating statistics is that to get rebuilt, they need to scan the table, sometimes this scans the entire table, or it can be set to scan a percentage of the table. The waste here is when you update the statistics for a large table with a very small number of rows that have changed. In this case imagine a table with 10 million rows, and in the last week there have been 10,000 rows inserted. Those 10,000 rows are only 1/10th of 1 percent of the entire table, or an insignificant change. That small of a change is not likely to impact the overall statistics.

Now picture a table with 20,000 rows that has had 10,000 rows inserted in the last week, that's 50% of the entire table. In this case the statistics are likely to be very inaccurate due to the current table being double the size it was when the statistics were generated. In this case statistics should be rebuilt.

The problem with the built in SQL Server Update Statistics maintenance plan task, or other similar methods that update all statistics is that statistics get updated for all indexes on all tables whether they need it or not. This is like the example above of washing all the dishes in my kitchen whether they are clean or dirty.

When the statistics are being rebuilt, the process may require a significant number of data pages to be loaded into memory, which causes other data pages being used by running queries to be pushed out of memory. This will also cause a huge increase in the amount of I/O from the storage system holding the database files. This can have a significant impact on the performance of other queries running on the system, and slow down everything.

## Steve Stedman

Freelance SQL Server Consultant

<http://stevestedman.com>

The solution is to stop rebuilding all statistics. Instead rebuild only some of the statistics on a daily or weekly basis. A job can be set up to rebuild just those statistics that have a significant number of changes to the table, or to rebuild just those statistics that haven't been rebuilt in a certain time window. I like to do a combination of the two to keep things up to date without causing too much of an impact on the system.

This is also good to do during the off hours, or slow times for your server. Some people argue that you should generate statistics during the typical high performance time on your server, but all that does is cause problems. Since the way that statistics are rebuilt is unrelated to how tables are actively being used by queries, they can be rebuilt just fine during off hours.

Have a daily job that runs in off hours that does the following:

For the first N statistics that meet the following criteria rebuild them.

- Statistics for indexes that may have never been built. (level 1)
- Statistics for tables that have changes. (level 2)
- Statistics for tables that haven't been rebuilt for some time. (level 3)

Here is an example that will update statistics for the 10 indexes needing it the most.

I hope this helps give a better understanding of Updating SQL Server Statistics, and how to update without causing your system to slow to a crawl.

## Need Help?

Need help with performance tuning, or updating SQL Server statistics, just contact [Stedman Solutions, LLC](#) we can help.