**Steve Stedman**
Freelance SQL Server Consultant
http://stevestedman.com

# Temp Table vs Table Variable vs CTE and the use of TEMPDB.

For more information on Common Table Expessions and performance, take a look at my book at Amazon.com: Common Table Expressions Joes 2 Pros®: A CTE Tutorial on Performance, Stored Procedures, Recursion, Nesting and the use of Multiple CTEs

There are many reasons that a Temp Table, Table Variable or Common Table Expression can be a good solution, but there is also a lot of confusion around these.  Much of that confusion is around the use of TempDB.

For the purpose of this article I am going to use my standard Departments table from my CTE presentation. It is just a simple database to work with. Also, this is running on my local SQL Server with no other users connected.

Then to compare the difference on the TempDB usage, I am going to use an under-documented function called fn_dblog() which shows you a list of the items that are written to the transaction log in different scenarios.  I will also combine this with the CHECKPOINT command to clear the contents of the log for TempDB.

To try out out, run the following on a TEST or DEVELOPMENT database.  Don't run this on your production system.

Then take a look at the output which will look something like this.

The first time you select from fn_dblog() you may be many rows.  In my example fn_dblog() returned 171 rows, you may have many more.  But after running CHECKPOINT you only get 3 rows, and those are the 3 rows that you will normally get after running CHECKPOINT.

Now lets take a look at how Temp Tables, Table Variables and Common Table Expressions compare on their use of TempDB when doing similar work.

## Temp Tables

Here is a sample of a temp table.

Now if we want to check the log before and after the temp table usage, here is how I would do it.

And you can see from the results this this temp table example we end up with 110 log entries for about 12KB of data written.

## Table Variables

Now here is how we would do the same thing with a table variable.

Then we wrap it with the log checks to see how much TempDB is being used.

Producing the results below for log entries which show 80 log entries for about 8KB of data written to log.  Better than the temp table usage.

Now lets see how the CTE's perform.

With the log checking here is what we get.

which produces these results, which surprised me.

Which shows NO TempDB usage when using the CTE's. Keep in mind that TempDB usage is not a bad thing, the TempDB is there for a reason, and it should be used, but if you are running into contention or load issues on TempDB, you can get around these by using a CTE rather than a Temp Table, or Table Variable.

## Recursive Common Table Expressions

Right after posting this article I received several questions on twitter about how the Recursive CTE is different than the non-Recursive CTE in the usage of TempDB. Based on the comment made by Mr Ogden to the post, I wanted to be sure to get this one answered.

Here is the basic Recursive CTE from my Unleashing CTE's presentation.

and here it is with the TempDB checking:

Which shows the following output.

Showing that there is no log written in TempDB with the Recursive CTE, and therefore TempDB is not used with this recursive CTE. No in the them of Mythbusters, lets take it further. I pulled in a few of my favorite recursive CTE examples from my Unleashing CTE's presentation, which include replacing a number table with a CTE, Calculating the Fibonacci Sequence with a CTE, and Calculating Factorial with a CTE. Lets see what that does to TempDB.

When run produced the following output.

Which shows that nothing was written to TempDB with any of these 3 recursive CTEs. Continuing on the Mythbusters way of breaking myths, lets go further.

## CTE in a Function

Using samples from the same Unleashing CTE's presentation, I decided to take a look at some of the CTEs that were wrapped in a function to see how they perform. I am going to use the paging example from that presentation.

This is simply doing data paging through the sys.columns table.

When this was run it didn't impact TempDB as shown here.

But then we take the same CTE and wrap it in a function like this....

Then call it from inside our usual test case:

Still no TempDB usage.

## Summary

What I have proven here is that if you are concerned with the usage of TempDB, for instance you are already experiencing TempDB problems, then CTE's are a better solution than Temp Tables and Table Variables because they do not use TempDB, for either a normal CTE, a recursive CTE, or a CTE wrapped in a function.

With that being said, there are many other reasons that you may want to choose a temp table, or a table variable over a CTE, so be sure you understand the details when deciding on a Temp Table, Table Variable or CTE.

For more information on Common Table Expessions and performance, take a look at my book at Amazon.com: Common Table Expressions Joes 2 Pros®: A CTE Tutorial on Performance, Stored Procedures, Recursion, Nesting and the use of Multiple CTEs