

Creating a Workload by Using a Trace Log

23 Jun 2010 7:54 PM | [Q](#)

In my previous post, [Improving Windchill Performance by Adding Missing Indexes](#), I illustrated the easy way to identify some missing indexes by using SQL Server 2005 Performance Dashboard. These are the low-hanging fruit. To reach the higher fruit, you need to use a longer and more-traditional route: create a workload and run it by using the Database Engine Tuning Advisor. In this post, I'll look at the first part of this route. This in-depth process takes some time and has a few potential issues.

Step 1: Record a Workload

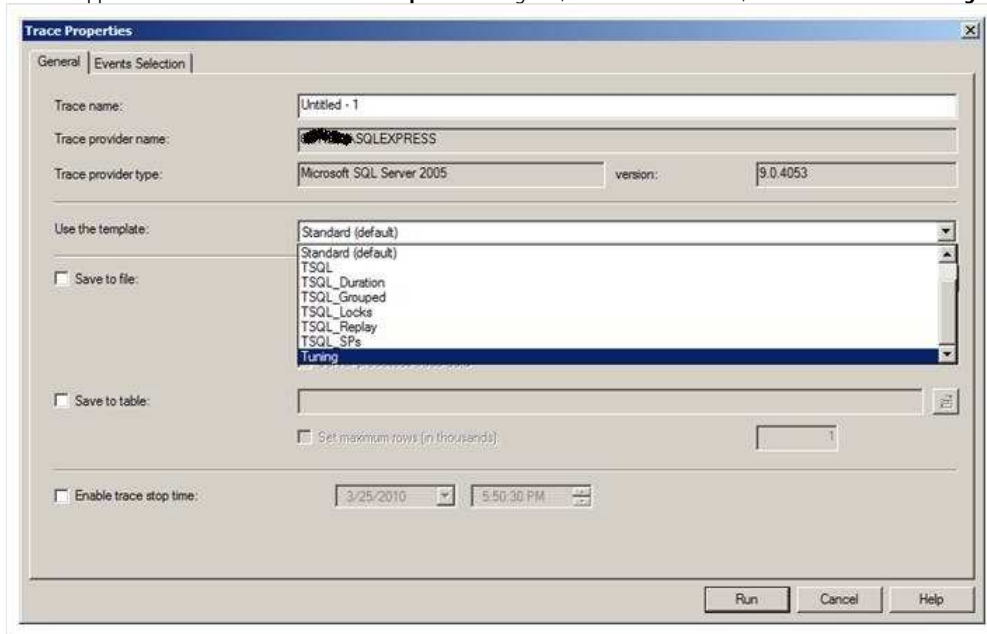
The first step is to record a workload by using SQL Server Profiler. If you are not familiar with SQL Server Profile, you can get a quick introduction at [How to Use SQL Server Profiler](#).

If you start SQL Server Profiler and start an arbitrary trace on an active Windchill database, you might find yourself frustrated. A single operation with Windchill with full tracing and real-world data can result in a 2-GB trace file. If you start a timed trace without a restriction on the size of the capture, you might run out of disk space on your computer (if you capture to a local file) or on the server that is running Windchill by the database file filling the hard drive (if you capture to a table). If you capture to a table and run out of space, the server that is running Windchill will stop, to the dismay of your users.

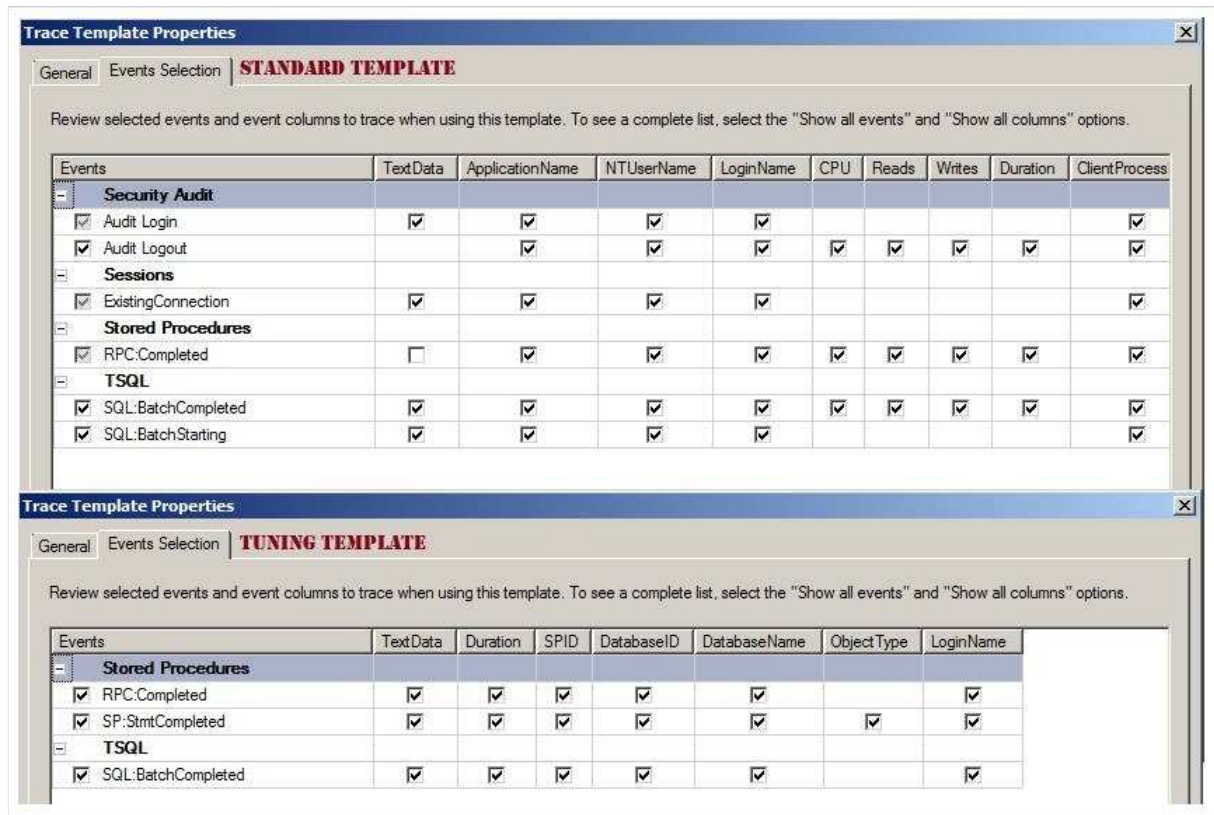
You can run a trace in two recommended ways:

- You can use the built-in performance template.
- You can build your own lightweight template.

The first approach is the easiest. In the **Trace Properties** dialog box, click the **General** tab, and then select the **Tuning** template.



This is a reduced template, as illustrated by the following figure.



Step 2: Write to a File

The next step is to write to a file so that you can limit the amount of data that is written. You do not want to give a blank check on your disk space to SQL Server Profiler. You might be tempted to try to write to a table and limit the table to only 10,000 rows; but, even that can result in a huge table because binary large objects (Text, NText, and Image) will be written to this table. For example, 10,000 rows x 10-MB document = 100 TB. Do you have a spare 100-TB hard drive on your system?

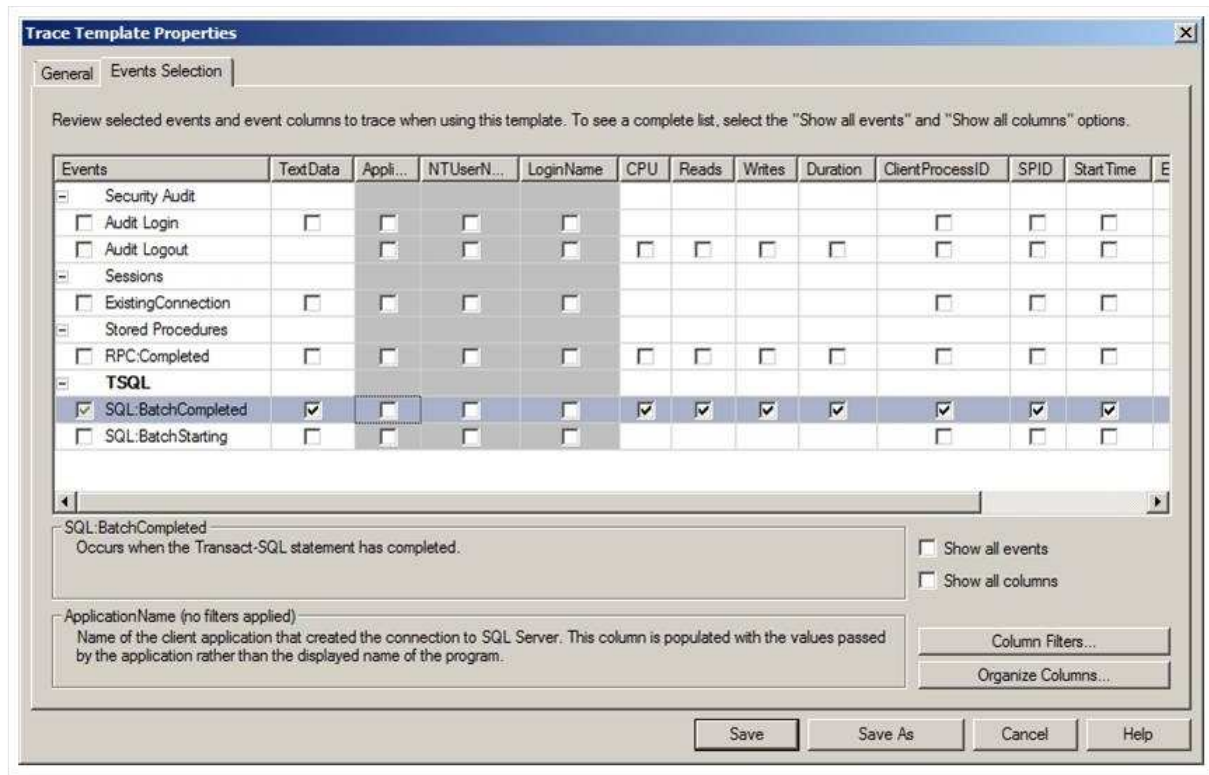
The safest solution is to see how much free space you have on a local desktop drive and divide that by 2. Use the resulting value for the maximum trace file size. The file will stop growing when it reaches this size, or it will be smaller. If you must run it on the server, try doing so on a logical drive that is not used for any database file. You want to minimize the impact of tracing on SQL Server. For more information, you may want to review [Flash Tip: Running Profiler on a Busy SQL Server](#) on TechNet.

The purpose of SQL Server Profiler is to capture a typical workload. So, it should run during the workday and preferably on a quiet workday so that it will not have a noticeable impact on your users. A general rule of thumb is to capture 1 hour of activity or until you run out of the allocated disk space for the log. To do this, click to clear the **Enable file rollover** check box (so that it will stop logging when it runs out of space). If you select the **Enable file rollover** check box, the log will keep recording and will keep only the last activity.

These steps produce a trace log that we will use in the next part of this series.

Create Your Own Trace Template

If you find that the Tuning template still results in huge files, you may want to create a very lightweight template that resembles the following.



For more information, see [How to Create Workloads](#) and [How to Create a Trace](#) on MSDN.

Potential Issues

Watch out for the following issues:

- If you tightened security on SQL Server, specifically if you used the [SHOWPLAN permission](#), the log may not be usable for Database Engine Tuning Advisor.
- In general, it is best not to capture the login name and to let Database Engine Tuning Advisor use the login name that you use to connect to the Windchill database (which must have the SHOWPLAN permission).

Next Time

In the next part of this series, I'll look at using these logs as a workload in Database Engine Tuning Advisor to identify additional missing indexes that SQL Server 2005 Performance Dashboard did not detect. These indexes might improve performance even more.

Ken Lassen is part of the original team that created Dr. GUI of MSDN and specializes in new and resurrected commercial product architecture. He developed architecture for several Microsoft websites, including the original [MSDN](#) site and the current [Microsoft Partner Network](#) site. He's equally at home with SQL Server, XHTML, Section 508 accessibility standards, globalization, Security Content Automation Protocol (SCAP) security, C#, and ASP.NET server controls. When he is not having fun with technology, he enjoys taking lunch-break hikes in the North Cascades.

