

Tuning your SQL Server 2005 database engine

Business and IT teams alike are demanding high-speed SQL Server 2005 installations right out of the gate. Getting the DBMS up and running is not usually the greatest challenge; it's ensuring that the platform performs as expected. This is no small task when you already have little time available to plan for a system already supporting production users or the platform was inherited from another department and is now your responsibility.

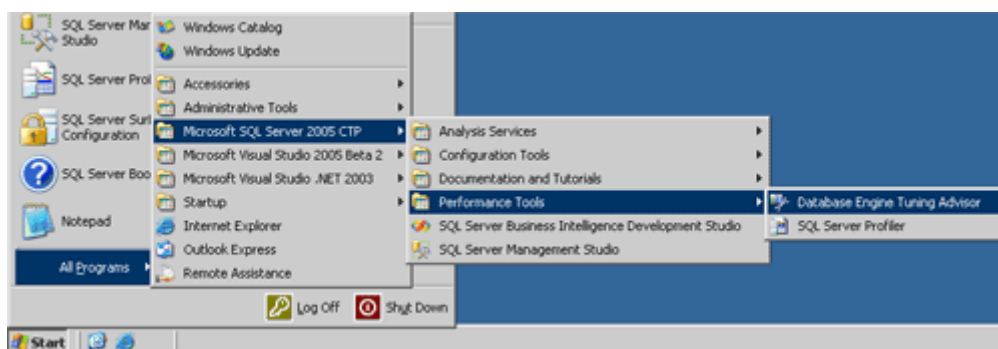
The SQL Server 2005 Database Engine Tuning Advisor is a good place to start addressing issues when you have little to no historical information about a system. This is one of the core tools replacing the SQL Server 2000 Index Tuning Wizard, offering new features and options for the new platform version. It is designed to analyze an existing SQL Server workload to determine beneficial indexes, indexed views and partitions. That includes options for dropping unnecessary indexes to release storage back to the DBMS and operating system.

Here I'll explain how to set up this tool and analyze a [SQL Profiler session](#). Setting up the Database Engine Tuning Advisor is a straightforward process, but many new features are available, which we will cover in this section.

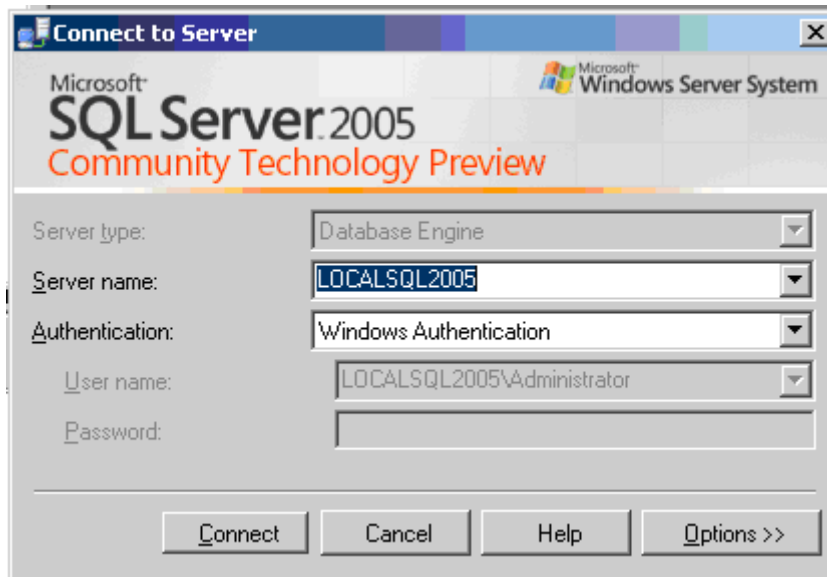
Before you begin, the prerequisite for properly using the Database Engine Tuning Advisor is a valuable Profiler session. This would include data from a time period when a performance issue was communicated and verified. For more information about SQL Server 2005 Profiler, re-read my earlier tip on [features, function and set up in SQL Server 2005](#).

Setting up the Database Engine Tuning Advisor

Step 1: Start SQL Server Database Engine Tuning Advisor via the GUI interface by navigating to Start | All Programs | Microsoft SQL Server 2005 | Performance Tools | Database Engine Tuning Advisor.



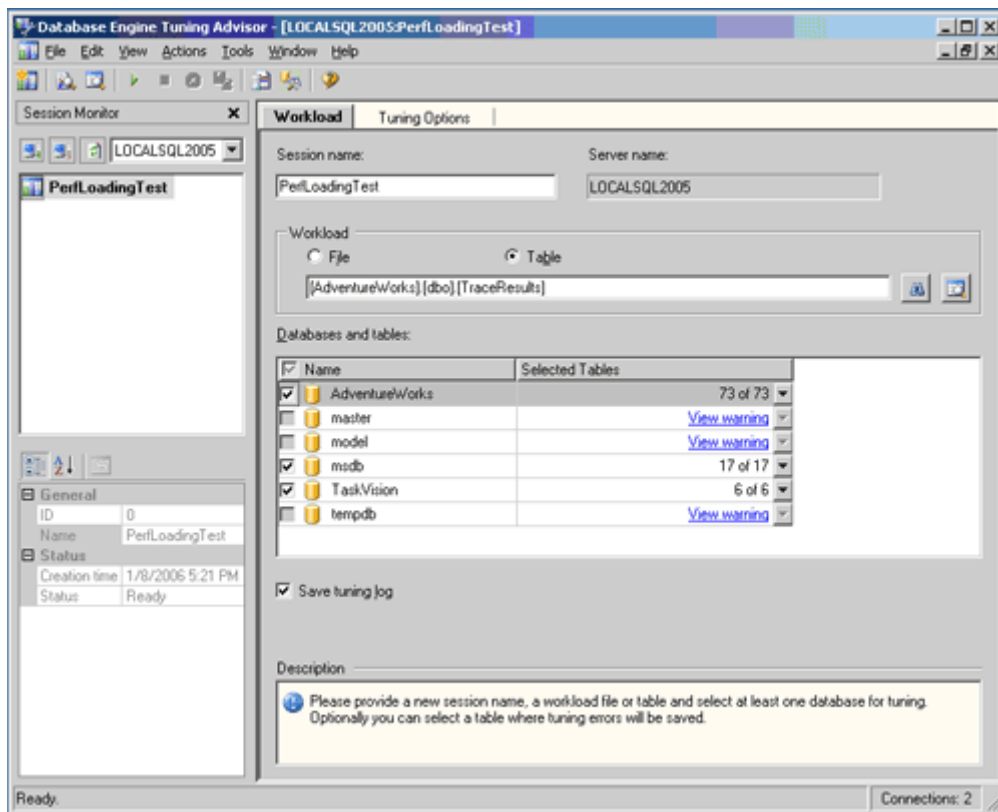
Step 2: Authenticate to SQL Server via the Database Engine Tuning Advisor via Windows Authentication or SQL Server Authentication. For this example, we will use Windows Authentication.



Step 3: Configure the Database Engine Tuning Advisor -- Workload Tab:

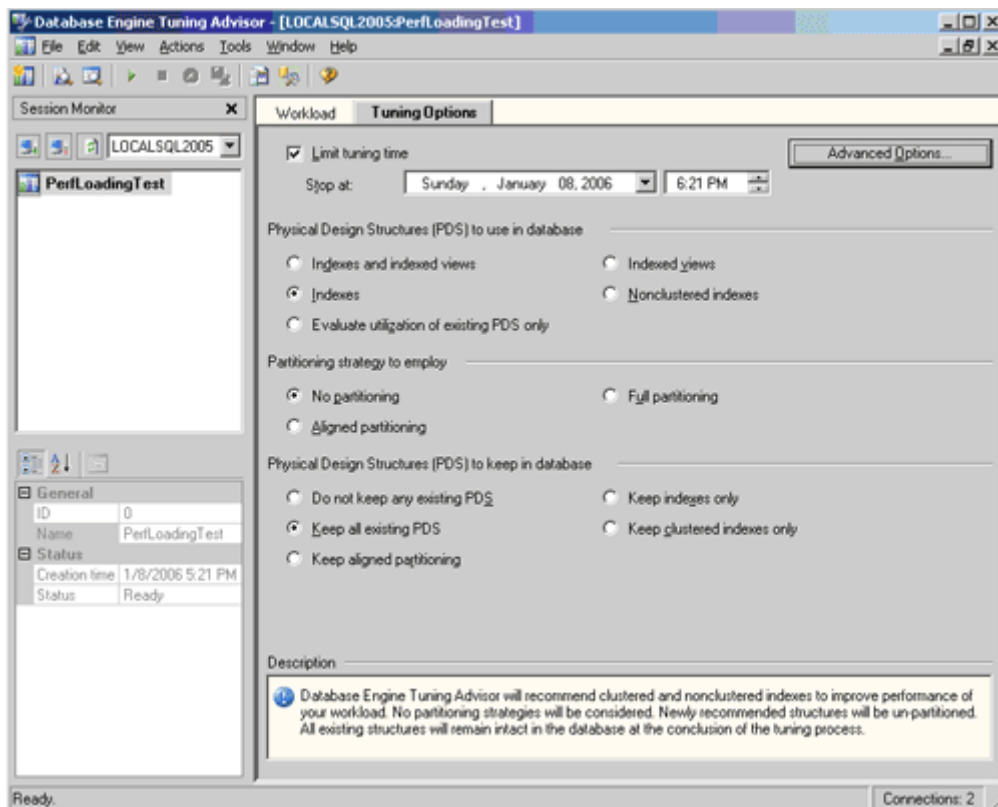
1. Name the session.
2. Specify a Workload file or table with Profiler data to serve as the basis for the analysis.
3. Select the databases and tables for the selected analysis.
4. Make sure the 'Save tuning log' check box is checked to review the final output.

For this session, we named the session PerfLoadingTest, referenced a table with Profiler data stored in the AdventureWorks database and selected all databases and tables.

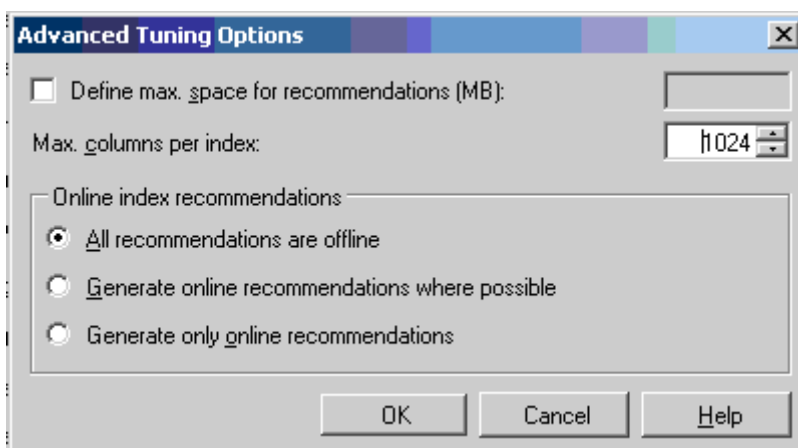


Step 4: Configure the Database Engine Tuning Advisor – Tuning Options Tab:

1. The first option helps limit the tuning time to a specific date and time. For this tip, we limited the time frame to the current date/time.
2. The second set of configurations pertains to the physical design structures, which are primarily indexes. For this tip, we selected the 'Indexes' radio button.
3. The third set of configurations affect partitioning strategy. For this example, we selected 'No partitioning.'
4. The fourth set of configurations address keeping or replacing physical design structures (i.e., indexes and partitions). For this example, we selected 'Keep all existing PDS.'

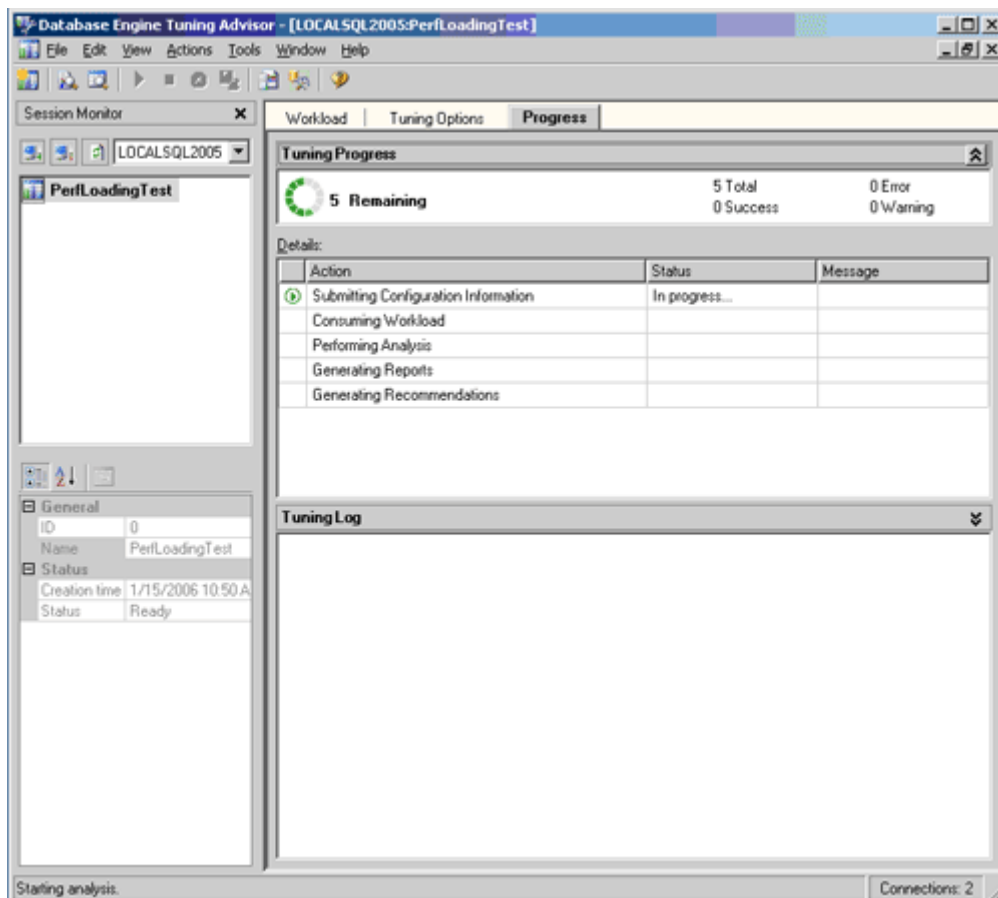


Step 5: The final configuration on the Tuning Option tab is the 'Advanced Options' button located on the top right of the tab. This interface lets you limit the amount of storage for the recommendations, which could be valuable with a limited amount of storage to support SQL Server. In addition, you can specify the type of index recommendation: online or offline.

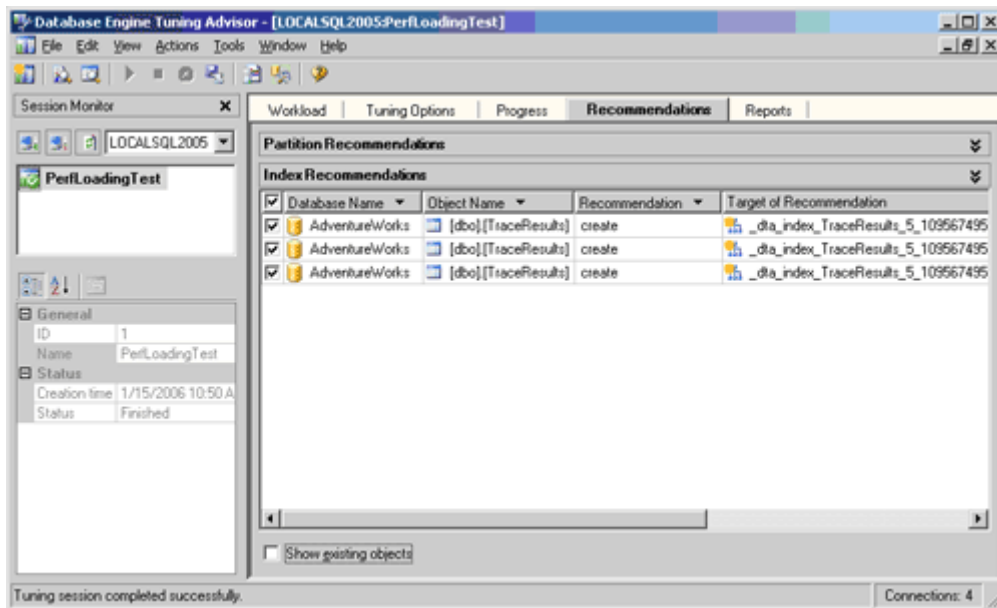


Step 6: Once all configurations are finalized, review them one last time and start the analysis: Press 'F5' or navigate to the 'Actions' menu and the 'Start Analysis' option. As the Database Engine Tuning Advisor is running, a 'Progress' tab will be added to show the overall status of the analysis. Keep in mind that the application may require a

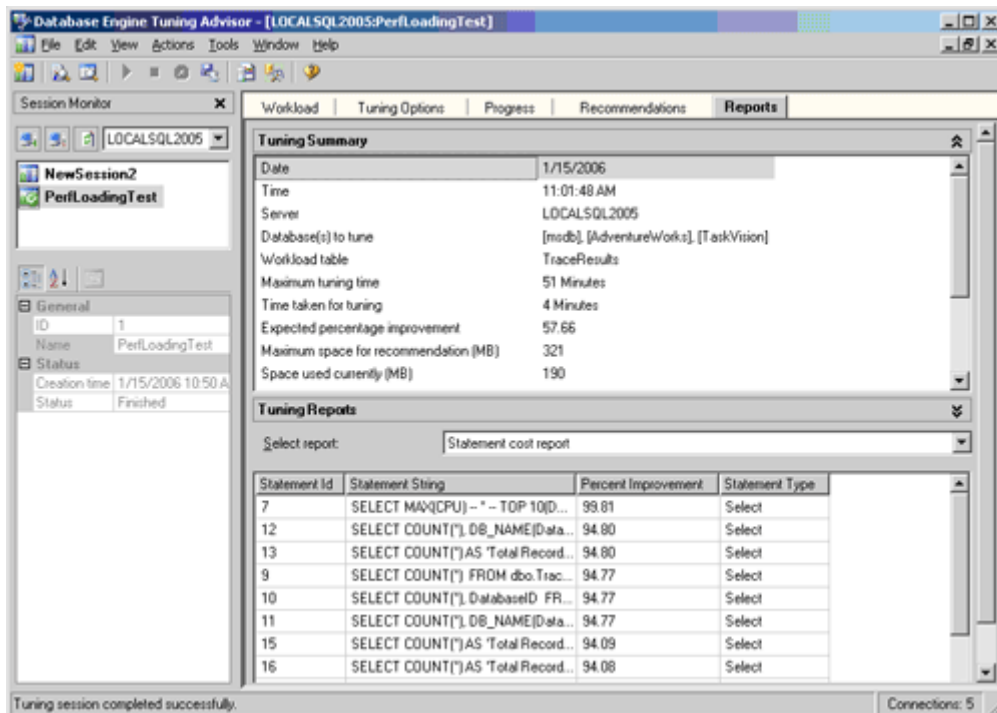
significant amount of time, so be patient.



Step 7: Once the Database Engine Tuning Advisor is complete, two new tabs will appear. These are the 'Recommendations' and 'Reports' tabs. First, we will evaluate the 'Recommendations' tab, which outlines beneficial partitions and indexes. With this 'Actions' menu, you can select the 'Save Recommendations' option to store the needed T-SQL in a text file to apply to your production system during a maintenance window.



Step 8: On the 'Reports' tab, it is important to review the summary and have a good understanding of expected improvement, current total database space and future database space, in addition to recommendations. It is also possible to select a report on the bottom pane. In this example, the 'Statement Cost Report' has been selected. In this report, the key T-SQL statements are outlined based on the percentage improvement from the recommendations. In the bottom pane, T-SQL statements with a 90% improvement are outlined. This is a significant improvement by just three indexes from the recommendation.



Additional tuning recommendations

1. Thoroughly test recommendations to ensure they meet your needs.
 2. Validate that changes to the system will improve the overall performance and not create a situation where performance is marginally improved in one area, but there are major degrades in another part of the application.
 3. Follow a [change management process](#) to deploy the code.
 4. Continue to monitor the SQL Server 2005 platform to ensure it is performing as expected, and continue tuning the system as changes are made to the application.
-

Conclusion

The combination of SQL Server Profiler, the Database Engine Tuning Advisor for SQL Server 2005 and your system knowledge will allow you to capture, address and correct performance tuning issues. Becoming proficient with these tools and having a firm understanding of your system is critical to long-term high performance. Good luck!