

Running Database Engine Tuning Advisor and Selecting Indexes

28 Jun 2010 3:13 PM | 0

In my earlier 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 were 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 my last post, [Creating a Workload by Using a Trace Log](#), I wrote about how to create a workload. In this post, I will look at the second part of this route, running Database Engine Tuning Advisor to identify performance-improving indexes.

This in-depth process takes some time and has a few potential issues. I will assume that you have obtained one or more trace logs and that you are familiar with Database Engine Tuning Advisor. If you are not, view the following webcasts and links:

- [MSDN Webcast: SQL Server MasterClass: Tuning Your SQL Server 2005 Database \(Level 200\)](#)
- [TechNet Webcast: SQL Server 2005 Database Tuning Advisor \(Level 300\)](#)
- [Database Engine Tuning Advisor Tutorial](#)

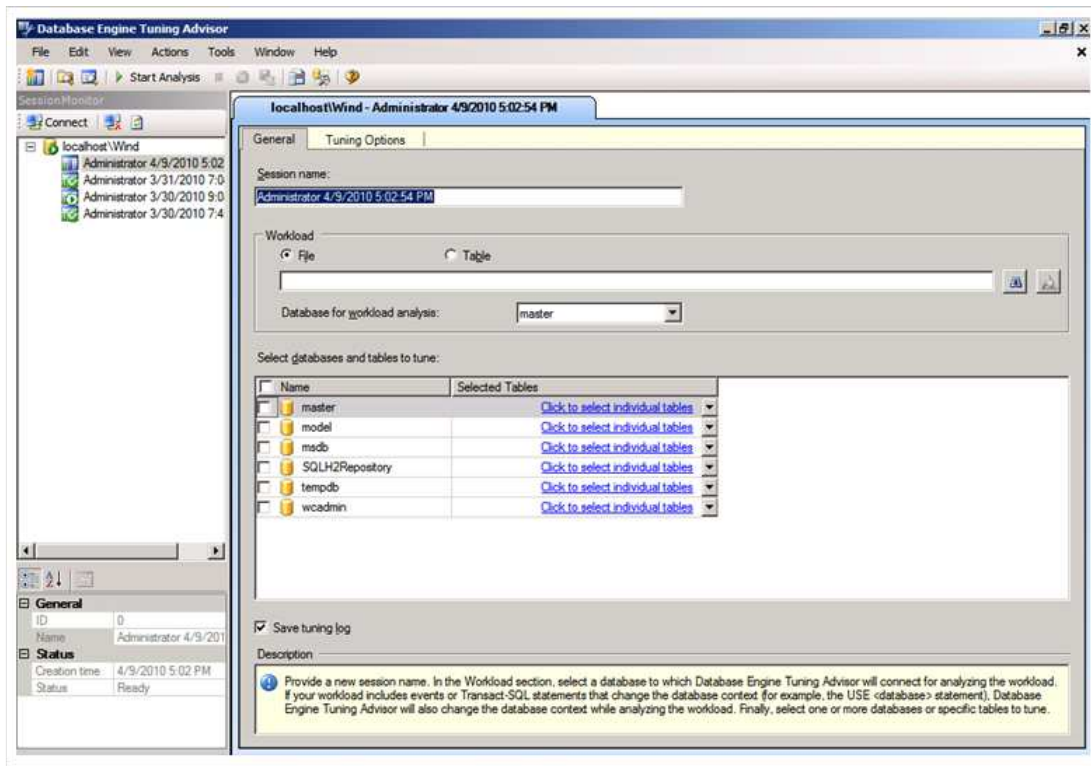
Run a Tuning Session

To start a Database Engine Tuning Advisor tuning session, follow these steps:

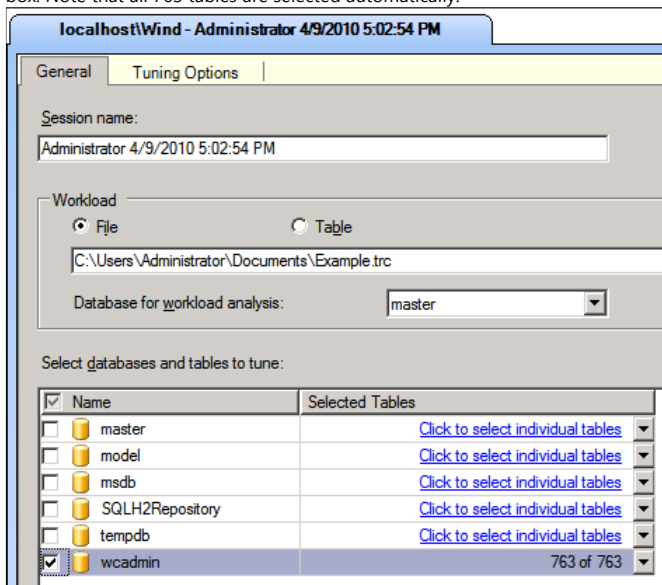
1. Verify that there is no load on the server that is running Windchill. Running Database Engine Tuning Advisor will put a significant load on the server and will result in a loss of performance while it is running. This is strictly an after-hours activity.
2. In the **Connect to Server** dialog box, connect to the server that is running Windchill.



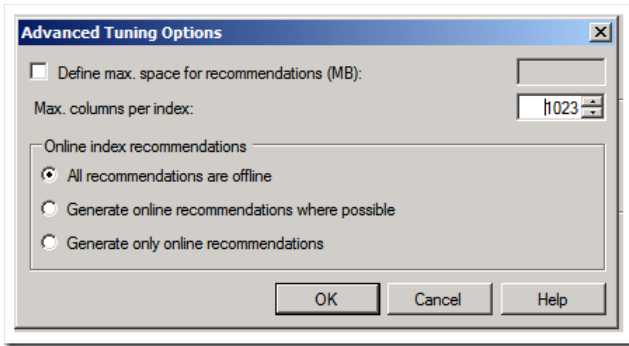
The Database Engine Tuning Advisor starts.



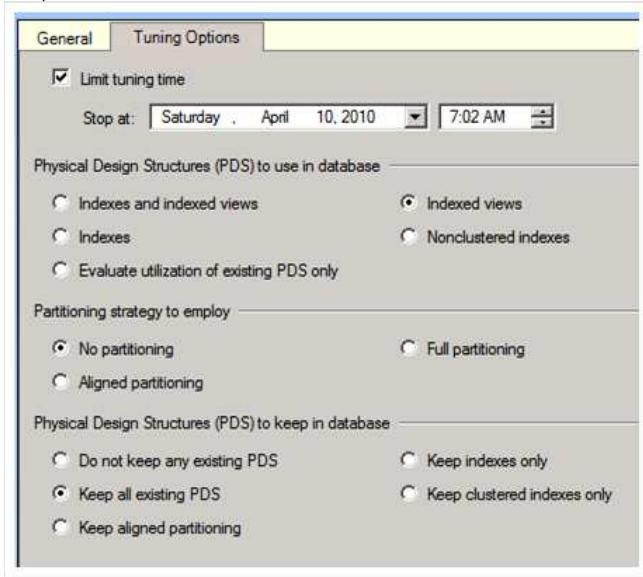
3. In the **Workload** area, select the file that you created by using SQL Server Profiler in the previous post, and then click to select the **wadmin** check box. Note that all 763 tables are selected automatically.



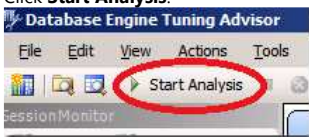
4. For the first run, click the **Tuning Options** tab, and then click the **Indexes, No partitioning**, and **Keep all existing PDS** radio buttons.
5. Click **Advanced Options**, verify that the setting for **Max. columns per index** is **1023**, verify that **All recommendations are offline** is selected, and then click **OK**.



- Estimate how long this server is going to be unused. Typically, this is an end-of-day activity, so set the stopping time to one hour before you come to work or one hour before users are expected to start. Do not allow Database Engine Tuning Advisor to run without a stopping time or to cross into production hours.



- Click **Start Analysis**.



The screen will start filling up with the capture.

| EventClass | TextData | Duration | SPID | DatabaseID | DatabaseName | Object Type | LoginName |
|--------------------|---|----------|------|------------|--------------|-------------|-----------|
| RPC:Completed | exec sp_execute 631,10191,N'wt.proj... | 0 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | SELECT 'wt.fc.dynamicenum.FilteredD... | 0 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | exec sp_execute 631,7630,N'wt.proje... | 0 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | SELECT 'wt.fc.dynamicenum.FilteredD... | 0 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | exec sp_execute 631,4,N'wt.project... | 0 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | SELECT 'wt.team.RolePrincipalMap',A... | 0 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | exec sp_execute 632,N'roleGroups',1... | 0 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | SELECT 'wt.team.RolePrincipalMap',A... | 0 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | exec sp_execute 632,N'orgs',10190 | 0 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | (SELECT 0 indexAlias ,CONVERT(image... | 0 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | declare @p1 int set @p1=633 exec ... | 7 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | SELECT 'wt.access.PolicyAc1',A0.blo... | 0 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | exec sp_execute 607,7,N'wt.rule.Typ... | 0 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | SELECT 'wt.rule.TypeBasedRule',A0.c... | 2 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | declare @p1 int set @p1=634 exec ... | 6 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | SELECT 0 indexAlias ,A0.blob\$entryS... | 0 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | declare @p1 int set @p1=635 exec ... | 2 | 54 | 5 | wcadmin | | wcadmin |
| SP:StmtCompleted | SELECT 'wt.preference.PreferenceIns... | 0 | 54 | 5 | wcadmin | 20816 - PQ | wcadmin |
| RPC:Completed | declare @p1 int set @p1=636 exec ... | 0 | 54 | 5 | wcadmin | | wcadmin |
| SQL:BatchCompleted | IF @@TRANCOUNT > 0 COMMIT TRAN | 0 | 54 | 5 | wcadmin | | wcadmin |
| RPC:Completed | exec sp_unprepare 636 | 0 | 54 | 5 | wcadmin | | wcadmin |

Then, the **Progress** tab is displayed.

localhost\Wind - Administrator 4/9/2010 5:02:54 PM

General | Tuning Options | **Progress**

Tuning Progress

5 Remaining

Details:

| Action |
|--------------------------------------|
| Submitting Configuration Information |
| Consuming Workload |
| Performing Analysis |
| Generating Reports |
| Generating Recommendations |

Tuning Log

The process will take several hours to complete if lots of transactions were recorded during the workload. When the tuning calculations are complete, the **Recommendations** tab and the **Reports** tab are displayed.

Apply the Recommendations

Click the **Recommendations** tab to see the index recommendations and details about the recommendations. You might think that the list of recommendations that is generated is overwhelming.

| Index Recommendations | | | | |
|-----------------------|------|------|------|--|
| Database Name | O... | T... | P... | Definition |
| wcadmin | | | | {[InheritedDomain] asc} |
| wcadmin | | | | {[distributable] asc} |
| wcadmin | | | | {[idA3A5] asc, [idA2A2] asc, [idA3B5] asc} include ([classnamekeyroleAObjectRef], [classnamekeyroleBObjectRef], [cr... |
| wcadmin | | | | {[idA3A5] asc, [name] asc, [markForDeleteA2] asc, [idA2A2] asc, [idA3B5] asc} include ([classnamekeyroleAObjectRef] |
| wcadmin | | | | {[finalPhase] asc} |
| wcadmin | | | | {[updateCountA2] asc} |
| wcadmin | | | | {[updateCountA2] asc} |
| wcadmin | | | | {[clientOverridable] asc} |
| wcadmin | | | | {[markForDeleteA2] asc} include ([clientData], [clientDelegate], [clientOverridable], [defaultValue], [displayName], [fullD... |
| wcadmin | | | | {[enabledFlag] asc, [markForDeleteA2] asc, [idA3containerReference] asc, [classnameA2A2] asc} include ([classname] |
| wcadmin | | | | {[ruleType] asc, [enabledFlag] asc, [idA3containerReference] asc, [lobiType] asc, [idA2A2] asc, [markForDeleteA2] asc} |
| wcadmin | | | | {[updateCountA2] asc} |
| wcadmin | | | | {[idA3A3] asc} include ([displayStatus], [uiComponent]) |
| wcadmin | | | | {[uiComponent] asc, [idA3A3] asc} include ([displayStatus]) |
| wcadmin | | | | {[disabled] asc} |
| wcadmin | | | | {[idA3B3] asc, [disabled] asc, [idA2A2] asc} include ([classnamekeyB3], [negative], [owner], [permissionMask], [classna... |
| wcadmin | | | | {[latestIterationInfo] asc, [idA2A2] asc, [branchIdIterationInfo] asc} |

The number of recommendations is often considerably more than the number that SQL Server 2005 Performance Dashboard produces. SQL Server 2005 Performance Dashboard detects the low-hanging fruit, but Database Engine Tuning Advisor covers the entire tree.

When you scroll to the right, you see the **Definition** column.

Now, you need to choose which indexes to implement.

The classic dilemma with adding indexes is to be sure to not add one-too-many indexes. Excessive indexes may result in a loss of performance. My usual cutoff is to stop adding indexes when adding all of the recommendations would result in less than a 5 percent improvement.

You should add the indexes that have the least number of columns specified in the **Definition** column, and then repeat the Database Engine Tuning Advisor run. This approach gives the most impact at the least cost. Here's why:

- The fewer the columns, the smaller the indexes are physically and the fewer the resources required to maintain them.
- Most well-designed databases have well-defined relationships between tables that involve only a few columns.
- When a where clause is encountered in a select clause, performance can be greatly improved by building an index on just one of the where clause columns.

This index might change a table scan into a subset scan. There are greatly diminishing returns on adding additional columns seen in the where clause, because columns often have a high correlation coefficient.

The initial recommendations shown below illustrate the process.

| Index Recommendations | | | | |
|-----------------------|--------------------------------------|-------------------------------------|-------------------|--|
| Database Name | Object Name | Target of Recommendation | Details | |
| wcadmin | [wcadmin].[AbsCollectionCriteriaKey] | PK_AbsCollectionCriteriaKey | clustered, unique | |
| wcadmin | [wcadmin].[AbsCollectionCriteriaKey] | PK_AbsCollectionCriteriaKey | | |
| wcadmin | [wcadmin].[AbsCollectionCriteriaKey] | AbsCollectionCriteriaKey\$UNIQUE | | |
| wcadmin | [wcadmin].[AbsCollectionCriteriaKey] | AbsCollectionCriteriaKey\$COMPOSITE | | |
| wcadmin | [wcadmin].[AbsCollectionCriteriaKey] | AbsCollectionCriteriaKey\$COMPOSITE | | |
| wcadmin | [wcadmin].[AbsCollectionCriteriaKey] | AbsCollectionCriteriaKey\$UNIQUE | unique | |
| wcadmin | [wcadmin].[AcceptedStrategy] | PK_AcceptedStrategy | clustered, unique | |
| wcadmin | [wcadmin].[AcceptedStrategy] | PK_AcceptedStrategy | | |
| wcadmin | [wcadmin].[AcceptedStrategy] | AcceptedStrategy\$COMPOSITE1 | | |
| wcadmin | [wcadmin].[AcceptedStrategy] | AcceptedStrategy\$COMPOSITE | | |
| wcadmin | [wcadmin].[AcceptedStrategy] | AcceptedStrategy\$COMPOSITE | | |
| wcadmin | [wcadmin].[AcceptedStrategy] | AcceptedStrategy\$COMPOSITE1 | | |
| wcadmin | [wcadmin].[AccessControlSurrogate] | PK_AccessControlSurrogate | clustered, unique | |
| wcadmin | [wcadmin].[AccessControlSurrogate] | AccessControlSurrogate\$COMPOSITE2 | | |

It is important to note that the **Details** column may contain the words **unique** and **clustered** as qualifiers for the index definition. Typically, these are your best first-choice items:

- Your first choices should be items that are labeled **clustered, unique** in the **Details** column (regardless of the number of columns) because they often affect the structure of data on the disk and thus add extra performance beyond that of indexing. Unfortunately, you cannot click the **Details** column to sort it; instead, you must look at all the rows.
- If there are no recommendations that are labeled **clustered, unique**, move on to the recommendations that have just one column.

| Index Recommendations | | | | | | | | |
|-------------------------------------|---------------|------------|----------|---------------------|---------|------|-----------|---|
| <input checked="" type="checkbox"/> | Database Name | Object ... | Recom... | Target of Recomm... | Details | P... | Size (KB) | Definition |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_Ho... | | | 288 | ([idA3A5] asc, [idA2A2] asc, [idA3B5] |
| <input checked="" type="checkbox"/> | wcadmin | [wc... | create | _dta_index_No... | | | 8 | ([parentTemplateIsNull] asc) |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_Ph... | | | 96 | ([idA3A5] asc, [iname] asc, [markForD |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_Pr... | | | 1032 | ([markForDeleteA2] asc) include ([clie |
| <input checked="" type="checkbox"/> | wcadmin | [wc... | create | _dta_index_Str... | | | 1000 | ([quarantined] asc) |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_Ty... | | | 56 | ([enabledFlag] asc, [markForDeleteA2 |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_Ty... | | | 64 | ([ruleType] asc, [enabledFlag] asc, [id |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_UI... | | | 224 | ([idA3A3] asc) include ([displayStatus] |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_UI... | | | 224 | ([uiComponent] asc, [idA3A3] asc) inc |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_W... | | | 104 | ([idA3B3] asc, [disabled] asc, [idA2A2 |
| <input type="checkbox"/> | wcadmin | [wc... | create | _dta_index_W... | | | 16 | ([latestIterationInfo] asc, [idA2A2] asc, |

Now, click **Apply Recommendations**, and the **Applying Recommendations** dialog box is displayed.

Applying Recommendations

Success 5 Total 0 Error
5 Success 0 Warning

Details:

| Action | Status | Message |
|--|---------|---------|
| Generating Transact-SQL script for recomme... | Success | |
| Switching database context to 'wcadmin'. | Success | |
| Applying 'create' recommendation on '_dta_j... | Success | |
| Applying 'create' recommendation on '_dta_j... | Success | |
| Refreshing status of databases. | Success | |

Rerun the Tuning Session

Having to wait until the evening to rerun the Database Engine Tuning Advisor is often difficult, but slow, systematic analysis is the best way to ensure that you add the indexes that have the maximum per-index value. If you have multiple trace logs available, I suggest that you set the Database Engine Tuning Advisor's workload to rotate the trace logs each evening to get better results. Mathematics and statistics predict better results by rotating trace logs. The next time through, you may have to go up to two column recommendations and then to three. Count any columns that are shown in the include clause of the **Description** column as one column.

If one of the trace logs fails below the threshold for adding indexes, do not stop. Eliminate this trace log from the list, and continue with the next trace log. For example, the next Database Engine Tuning Advisor session resulted in the following recommendations.

So, just adding two of the long list of recommendations resulted in 50 percent of the potential improvement. (The original run had an estimated 4 percent improvement; this run has an estimated 2 percent improvement.)

| Estimated improvement: 2% | | | | | | | |
|-------------------------------------|---------------|-----------------------------|---------|-----------------|---------|------|--|
| Partition Recommendations | | | | | | | |
| Index Recommendations | | | | | | | |
| <input checked="" type="checkbox"/> | Database Name | Object Name | Reco... | Target of Re... | Details | P... | Definition |
| <input checked="" type="checkbox"/> | wcadmin | [wcadmin].[HolderToContent] | create | _dta_ind... | | | ([idA3A5] asc, [idA2A2] asc, [idA3B5] asc) include ([classname |
| <input checked="" type="checkbox"/> | wcadmin | [wcadmin].[UIAccess] | create | _dta_ind... | | | ([uiComponent] asc, [idA3A3] asc) include ([displayStatus]) |

The key points to remember about running Database Engine Tuning Advisor are:

- Never run it during business hours.
- Always tell it when to stop.
- Run it against several loads captured by SQL Server Profiler.
- Add the simplest indexes first, and then rerun the session.

Hitting the Threshold

At some point, you will reach your improvement threshold and stop adding indexes. Then, you have one more step to do: delete unused or expensive indexes. My next post will show you how to do this.

Ken Lassenen 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.