

# Re: DBReindex

---

*Source:* <http://www.derkeiler.com/Newsgroups/microsoft.public.sqlserver.security/2008-02/msg00095.html>

---

- *From:* Ola Hallengren <[OlaHallengren@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:OlaHallengren@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 21 Feb 2008 13:12:01 -0800
- 

Hello!

An action you also can consider is to add some intelligence to your job, so that you only rebuild indexes on tables that have more than, say, 30% in fragmentation.

You can also consider ALTER INDEX REORGANIZE.

I have made a stored procedure that have some of this logic that you're welcome to use.

<http://blog.ola.hallengren.com/blog/archives/2008/1/1/3440068.html>  
<http://blog.ola.hallengren.com/attachments/3440068/IndexOptimize.sql>  
<http://blog.ola.hallengren.com/attachments/3440068/Documentation.html>

Best regards

Ola Hallengren  
<http://ola.hallengren.com>

"Erland Sommarskog" wrote:

thejamie (thejamie@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx) writes:

I'm using a standard SQL Server edition and when I went to update my old dbcc dbreindex routine, I tried to update the dbcc dbreindex command with

```
ALTER INDEX '+@index+'  
ON '+@TableName+'  
REBUILD WITH ( PAD_INDEX = OFF, STATISTICS_NORECOMPUTE  
= OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
```

Re: DBReindex

`SORT_IN_TEMPDB = ON, ONLINE = ON )'`

When I replace the older statement with this one, I see  
"Online index operations can only be performed in Enterprise edition of  
SQL Server."

I assume there is a way to rebuild indexes in SQL 2005 but for me it isn't  
what is above.

Change `ONLINE = ON` to `ONLINE = OFF`, and you essentially have `DBCC  
DBREINDEX`.

When the jobs began to fail that come with the wizard maintenance, I  
replaced it with the `dbcc dbreindex` command. As I see this is causing  
my log problem, I am not sure where to go next. For example, not sure  
if maybe I should switch to something like "REBUILD PARTITION".

First of all, I hope you are backing up your transaction log regularly?

What you can do to reduce the strain on the transaction of rebuilding  
indexes is to set the recovery model to `BULK_LOGGED` while your maintenance  
job is running.

`BULK_LOGGED` is to a large extent the same as full recovery, but some  
actions are minimally logged in bulk-logged recovery, for instance the  
creation of indexes. Bulk-logged recovery still gives the possibility  
to restore the database up to the state of the latest transaction log.  
However, you cannot restore to a point in time if there has been any  
minimally logged operation since the last log backup.

An action you also can consider is to add some intelligence to your  
job, so that you only rebuild indexes on tables that have more than,  
say, 30% in fragmentation.

The fact that an index rebuild eats log is not strange: the entire table  
is essentially moved to a new location.

You can also consider `ALTER INDEX REORGANIZE` this command can be  
kinder to the log – but in extreme situations, the tool may be higher.  
And bulk-logged recovery does not help in this case.

--

Erland Sommarskog, SQL Server MVP, [esquel@xxxxxxxxxxxxx](mailto:esquel@xxxxxxxxxxxxx)

Books Online for SQL Server 2005 at

<http://www.microsoft.com/technet/prodtechnol/sql/2005/downloads/books.msp>

Books Online for SQL Server 2000 at

<http://www.microsoft.com/sql/prodinfo/previousversions/books.msp>

Re: DBReindex