

## SQL Server 200x Index Maintenance Best Practices

Kimberly L. Tripp

SQLskills.com

Email: [Kimberly@SQLskills.com](mailto:Kimberly@SQLskills.com)

Blog: <http://www.SQLskills.com/Blogs/Kimberly>

<http://www.SQLskills.com>



### Speaker – Kimberly L. Tripp

- Independent Consultant/Trainer/Speaker/Writer
- Founder, **YSolutions, Inc.** [www.SQLskills.com](http://www.SQLskills.com)
  - Email: [Kimberly@SQLskills.com](mailto:Kimberly@SQLskills.com)
  - *Become a subscriber on SQLskills.com and learn about new resources which can improve your productivity and server performance!*
- Microsoft Regional Director <http://msdn.microsoft.com/isv/rd/>
- SQL Server MVP <http://mvp.support.microsoft.com/>
- Author for some SQL Server 2005 Whitepapers on MSDN (links from home page on [www.SQLskills.com](http://www.SQLskills.com))
- SQL Server 2005 Launch Content Manager – Data Platform Track Sessions, Demos and Cross-training
- Writer/Editor for SQL Magazine [www.sqlmag.com](http://www.sqlmag.com)

**SQL**  
skills.com

immerse yourself in sql server

@Copyright 2005, Kimberly L. Tripp

Immersion Events – Intense, Focused, Real-world Training!

## Overview

- How Fragmentation Occurs
- What Fragmentation Means
- How do you know?
- Rebuilding or Defragging
- Achieving Online Index Rebuilds

## Index Fragmentation (Leaf Level)

- Data Modifications [can] lead to Fragmentation
- INSERT
  - Yes – Key value is not ever increasing/decreasing
  - NO – Key is ever increasing/decreasing
- UPDATE
  - Yes – Updates... to variable width columns – where the values are getting wider
  - NO – Columns are fixed width, columns have “place holder” values (i.e. DEFAULT constraints) to minimize row expansion on update OR no updates
- DELETE
  - Yes – Deletes are singleton deletes (swiss cheese problem)
  - NO – Deletes are RANGE deletes for archival purposes

## How Fragmentation Occurs

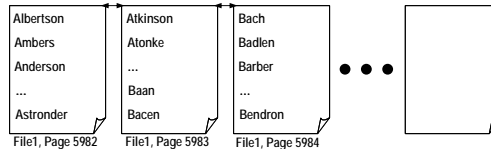
### Modifications Can Lead to Page Splits

- INSERTs to a page which is already full  
*(there is a special case when the INSERT is greater than the last row or less than the first row – on the page)*
- UPDATES to a variable width column on a page which is already full

Data Page

Review and Scenario

Header 96 Bytes
20 Rows/Page
8096 Bytes for data
96 Bytes Free



## How Fragmentation Occurs

### Modifications Can Lead to Page Splits

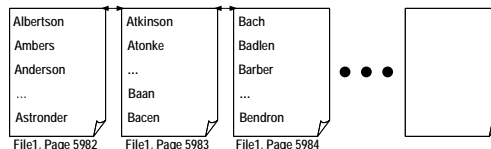
- INSERTs to a page which is already full  
*(there is a special case when the INSERT is greater than the last row or less than the first row – on the page)*
- UPDATES to a variable width column on a page which is already full

Data Page

Review and Scenario

Header 96 Bytes
20 Rows/Page
8096 Bytes for data
96 Bytes Free

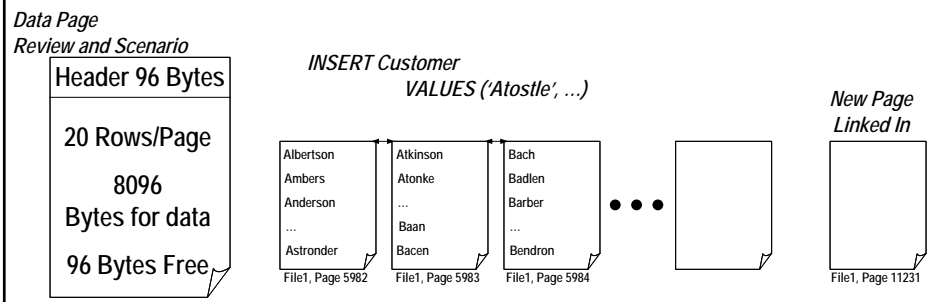
INSERT Customer  
VALUES ('Atostile', ...)



## How Fragmentation Splits Pages

If the page is full – on INSERT or UPDATE

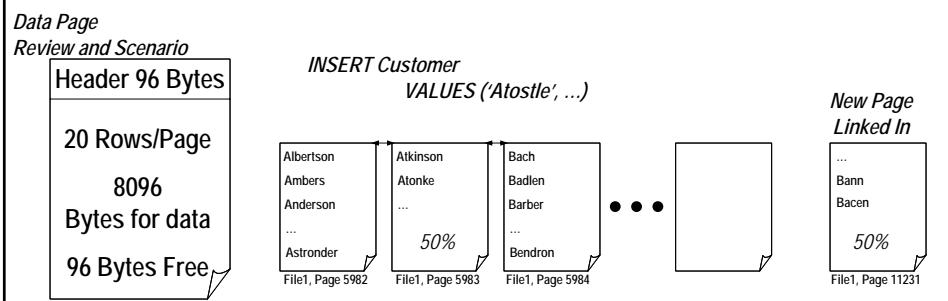
- 1) A New Page is “linked” in



## How Fragmentation Splits Pages

If the page is full – on INSERT or UPDATE

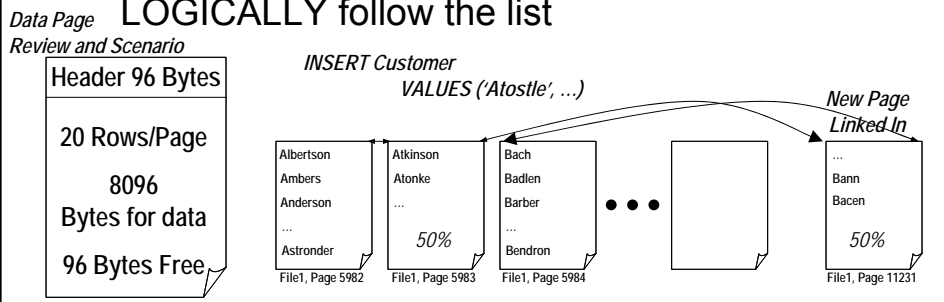
- 1) A New Page is “linked” in
- 2) 50% of the page being split is moved to the newly linked in page



## How Fragmentation Splits Pages

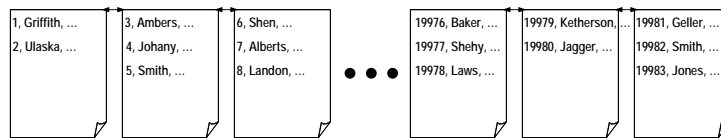
If the page is full – on INSERT or UPDATE

- 1) A New Page is “linked” in
- 2) 50% of the page being split is moved to the newly linked in page
- 3) Doubly-linked list/Pointers are updated to LOGICALLY follow the list



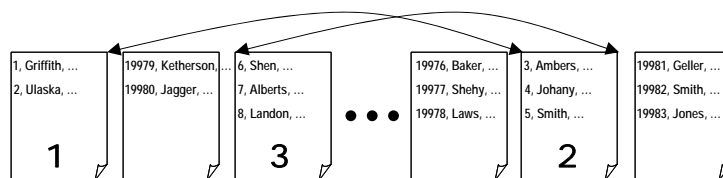
## What Fragmentation Means

- Internal Fragmentation
  - Free Space on the Page
    - Desirable for OLTP
      - Free Space for New Records
      - Free Space for Updates
    - Undesirable OLAP
      - Table requires more pages
      - Large Analysis queries require more I/O



## What Fragmentation Means

- External Fragmentation
  - Logical Left to Right Order of Leaf Level is NOT Physically Left to Right
    - Almost Always A Concern
    - Typically means splits have also resulted in internal fragmentation
    - Can cause disk thrashing for low cache and low selectivity queries



## How to See Fragmentation

- Poor Query Performance over time
- More disk activity
- Poor Cache Utilization
- Verify Query I/O
  - SET STATISTICS IO ON
- Verify Scan Density
  - sys.dm\_db\_index\_physical\_stats
    - Manually
    - Programmatically
    - Automatically
- Periodically re-verify Query I/O

## Detecting Fragmentation

- SQL Server 2000: DBCC SHOWCONTIG
  - Not tabular by default
  - Difficult to programmatically analyze
    - Pre-create temp table – have to know definition
    - Must use dynamic string execution to execute
- SQL Server 2005:
  - sys.dm\_db\_index\_physical\_stats(params)
    - Name length looks ugly...but MUCH better, dm objects make a large difference in overall server analysis
    - Table-valued function returns tabular set
    - EASY to programmatically analyze
      - Can use SELECT INTO to catch result set
      - No dynamic string execution needed!

## Analyzing Fragmentation

### sys.dm\_db\_index\_physical\_stats

- SQL Server 2005
  - Query the new DMV: dynamic management view
  - Multi-statement Table-valued function
  - Can use SELECT INTO to create temp table for programmatic analysis (great feature of ALL DMVs!)
  - Can run in multiple modes, target specific tables and/or indexes

```
SELECT *
INTO IndexFragmentation
FROM sys.dm_db_index_physical_stats
      (DatabaseID, TableID, IndexID,
       PartitionNumber, Mode)
```

## Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

Mode (Default = Limited)

- **LIMITED: IS Lock.** Same as SQL 2000 WITH FAST, only page counts and EXTERNAL fragmentation displayed. Does not detail INTERNAL fragmentation and page density.
- ☑ **SAMPLED: IS Lock.** For tables less than 10,000 pages (~80MB), all details are produced. For tables of more than 80MB, two samples are done (1% and 2%) at every nth page. The samples are compared and if close, 2% sampling output returned. If not close, then up to 10% will be sampled.

## Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

- **DETAILED: S Lock.** Entire table analyzed for both internal and external fragmentation. Returns one row for each level of the index from the leaf level (level 0) all the way up to the root level. This can help you determine fragmentation in the non-leaf levels but at the expense of holding a shared table level lock.

**NOTE:** If fragmentation in the leaf level is minimized through proper index maintenance and fillfactor – then fragmentation in the non-leaf levels should be low.

## Analyzing Fragmentation

- External Fragmentation
  - avg\_fragmentation\_in\_percent (*should be low*)
    - Uses a percentage to show extent switches
  - fragment\_count (*should be low*)
    - How many “chunks” exist within db
  - Avg\_fragment\_size (*should be high*)
    - Number of pages (avg) per fragment
- Internal Fragmentation
  - Average Bytes Free Per Page
    - Amount of free space (if OLTP +, if OLAP -)
  - Average Page Density (fullness)
    - Shown as a percentage – how FULL are the pages

## Rebuilding an Index

- Generally better results w/rebuild v. defrag
- Completely removes all levels of fragmentation – from both the leaf level as well as the b-tree (i.e. completely rebalances the tree)
- Completely updates statistics – with the equivalent of a “full scan” (i.e. accurate statistics)
- If offline, requires Locks ⇔ downtime
- How?
  - ALTER INDEX... REBUILD (offline OR online – *preferred*)
  - DBCC DBREINDEX (offline only – *backward compatibility only*)
  - CREATE w/DROP\_EXISTING = ON  
(can change the CL index definition)

## Defragging an Index

- Does not completely rebuild ALL levels of index – focuses on LEAF level
- Does not update statistics
- Does not require locks for length of transaction – defrag executes as mini-trans
- May take longer if table is extremely fragmented
- May take less time if table is not fragmented
- Does not run as a single large transaction – transaction log backups can execute and the transaction log space can be freed WHILE running
- How?
  - ALTER INDEX... REORG (*preferred*)
  - DBCC INDEXDEFRAG (*backward compatibility only*)

## How to Minimize Fragmentation

- Add Internal Space on Pages to Allow Fluctuation in Row Size (due to updates against variable width columns)
- How?
  - When **rebuilding** set appropriate FILLFACTOR Setting
- Other Data Dependant Options?
  - Use Only Fixed Width Columns
  - No Updates
  - No Deletes

## FILLFACTOR

FILLFACTOR impacts ONLY the LEAF level of an index (PAD\_INDEX affects B-tree)

0 – Default value. Leaf level is filled to 100%

- Excellent for OLAP
- Not ideal for OLTP

100 – Leaf level is filled to 100% (same as default)

- Not ideal for OLTP

1-99

- IDEAL for OLTP – you know your data!
- But how do you OPTIMALLY set this?

## FILLFACTOR

### Estimating appropriate FILLFACTOR

- What is going to cause splits in this structure?
  - INSERTs?
    - How often?
    - Is the data ever-increasing?  
The more volatile ⇒ LOWER FillFactor
  - UPDATES to variable width datatypes?
    - How wide?
    - How volatile?  
The wider and/or more volatile  
⇒ LOWER FillFactor
- How often can you rebuild?
  - The more frequent ⇒ HIGHER FillFactor

## FILLFACTOR

### Testing your FILLFACTOR Estimate

- SELECT FROM  
sys.dm\_db\_index\_physical\_stats between  
rebuids to see how far and how fast the scan  
density drops
  - The faster the table becomes fragmented  
(i.e. the lower the scan density goes)  
⇒ LOWER FillFactor  
or DECREASE the time between  
defrag/rebuids
- Programmatically analyze the output from DMV  
to determine if a rebuild or defrag should occur

## Rebuilding an Index

```
ALTER INDEX member_ident  
ON member REBUILD  
WITH (ONLINE = ON, FILLFACTOR = 90)
```

Msg 2725, Level 16, State 2, Line 1

Online index operation cannot be performed for index  
'member\_ident' because it contains column  
'photograph' of data type text, ntext, image,  
varchar(max), nvarchar(max), varbinary(max) or  
xml. The operation must be performed offline.

```
ALTER INDEX ChargePK  
ON Charge REBUILD  
WITH (ONLINE = ON, FILLFACTOR = 90)
```

## Index Maintenance

### ONLINE Rebuild

Indexes can be rebuild ONLINE if:

- The clustered index is NOT disabled
- The index does not include any large data types:

LOB DATA	image	text	ntext
	varbinary(max)	varchar(max)	nvarchar(max)
	xml		

- The index is not a partition
- The index is not on a local temp table

## Index Maintenance

### How Online Index Rebuilds Work

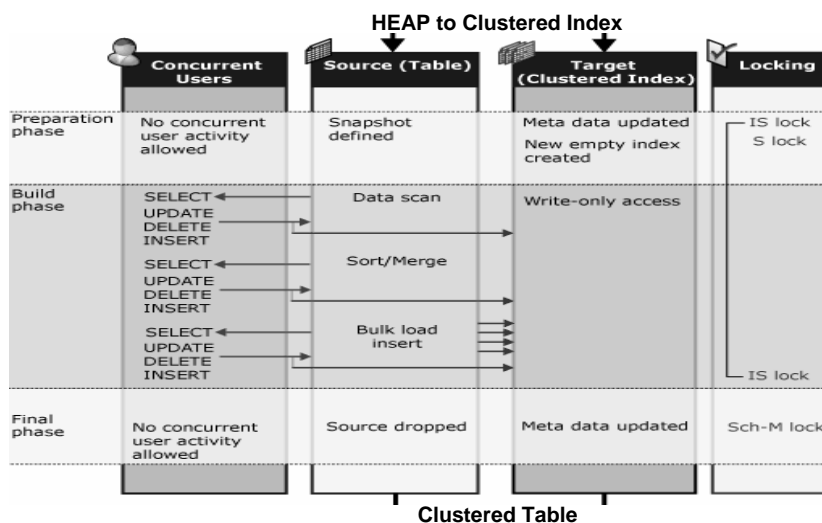


Diagram from BOL Topic: How Online Index Operations Work

## Online Progress Report

The screenshot shows the SQL Profiler interface with a trace capturing an online index rebuild. The main table displays the following data:

EventClass	TextData	EventSubClass	ObjectID	ObjectName	BigIntData1	BigIntData2	Applicatio
ExistingConnection	-- network protocol: LPC set quoted_id...						SQL Ser...
SQL:BatchStarting	alter index test on member rebuild with (...						SQL Ser...
Progress Report: Online Inde...		1 - Start	213575799	test			SQL Ser...
Progress Report: Online Inde...		2 - Stage 1 execution begin					SQL Ser...
Progress Report: Online Inde...		6 - Inserted row count	213575799		7221	0	SQL Ser...
Progress Report: Online Inde...		6 - Inserted row count	213575799		10000	0	SQL Ser...
Progress Report: Online Inde...		3 - Stage 1 execution end					SQL Ser...
Progress Report: Online Inde...		7 - Done	213575799	test			SQL Ser...
SQL:BatchCompleted	alter index test on member rebuild with (...						SQL Ser...
Trace Pause							
Trace Start							

Below the table, the SQL text being executed is visible:

```
alter index test on member rebuild with (online = on)
go
```

The status bar at the bottom indicates: "Trace is running. Ln 11, Col 3 Rows: 29 Connections: 1"

## Review

- How Fragmentation Occurs
- What Fragmentation Means
- How do you know?
- Rebuilding or Defragging
- Achieving Online Index Rebuilds

## MSDN Webcast Series

<http://www.microsoft.com/events/series/msdnsqlserver2005.msp>

- Session 1: Interaction between data and log
- Session 2: Recovery Models
- Session 3: Table optimization strategies
- Session 4: Optimization through indexes
- Session 5: Optimization through maintenance
- Session 6: Isolation, locking and blocking
- Session 7: Optimizing procedural code
- Session 8: Partitioning...
- Session 9: Profiling for the unknown problems
- Session 10: Common Roadblocks, *A Series Wrapup*

## Resources

Whitepapers written by Kimberly L. Tripp

- “SQL Server 2005 Snapshot Isolation”  
On MSDN and [www.SQLskills.com](http://www.SQLskills.com)
- “SQL Server 2005 Partitioned Tables”  
On MSDN and [www.SQLskills.com](http://www.SQLskills.com)
- Blogged this new one with a preview on SQLskills: The Database Administrator's Guide to the SQL Server Database Engine .NET Common Language Runtime Environment  
[www.sqlskills.com/blogs/kimberly/PermaLink.aspx?guid=385c0e05-497f-4c4c-b24a-155106a08959](http://www.sqlskills.com/blogs/kimberly/PermaLink.aspx?guid=385c0e05-497f-4c4c-b24a-155106a08959)
- Another one in the works: SQL Server 2005 Management Tools



**Thank you!**

**Please take a moment to fill out your evaluation.**

**Kimberly L. Tripp**

Consultant . Trainer . Writer . Speaker

email: [Kimberly@SQLskills.com](mailto:Kimberly@SQLskills.com)

**Make sure to register for special offers  
and other helpful information and resources!**

**[www.SQLskills.com](http://www.SQLskills.com)**



immerse yourself in sql server

@Copyright 2005, Kimberly L. Tripp

Immersion Events – Intense, Focused, Real-world Training!