



SQL Server 2005

New Features in Indexing

Kimberly L. Tripp
President, SYSolutions, Inc.
Founder, SQLskills.com



Speaker – Kimberly L. Tripp

- Independent Consultant/Trainer/Speaker/Writer
- Founder, SYSolutions, Inc. www.SQLskills.com
email: Kimberly@SQLskills.com
Become a subscriber on SQLskills.com and learn about new resources which can improve your productivity and server performance!
- SQL Server MVP <http://mvp.support.microsoft.com/>
- Microsoft Regional Director
<http://www.microsoftregionaldirectors.com/Public/>
- Writer/Editor for SQL Magazine www.sqlmag.com
- Coauthor MPress: *SQL Server 2000 High Availability*
- Presenter/Technical Manager for SQL Server 2000 High Availability Overview DVD (MS Part# 098-96661)



Overview

- Maintenance Improvements
 - Online operations
 - LOB Compaction
- Disabling Indexes
- Fragmentation Analysis Improvements
- DM Objects for Performance
- Statistics Changes
- Include Non-key Columns for better covering
- Index Tuning – Database Tuning Advisor
- XML Indexes
- Full Text Indexes

Microsoft

Index Maintenance

- Rebuilding
 - Rebuilds entire index structure
 - Updates statistics
 - Requires locks which leads to downtime
 - Requires Locks ∴ downtime
 - Rebuild CL requires Exclusive Table Lock
 - Rebuild NC requires Shared Table Lock
- Defragging
 - Not as exhaustive as rebuilding, only cleans up fragmentation in the leaf level
 - Does not move the object for better extent density
 - Does not require long running locks

Microsoft

Index Maintenance – Rebuild

- SQL Server 2000
 - DROP and re-CREATE
 - DBCC DBREINDEX
 - CREATE with DROP_EXISTING
- SQL Server 2005
 - All of the above, except: DBCC DBREINDEX is OFFLINE only, scripts using it should be changed to use ALTER INDEX...REBUILD
 - ALTER INDEX...REBUILD replaces DBCC DBREINDEX
 - Two modes of rebuilding:
 - ONLINE – allows concurrent user access (queries as well as modifications) to the index during rebuild
 - OFFLINE – works using locks (same as 2000)

Microsoft

Index Maintenance

ONLINE Rebuild

Indexes can be rebuild ONLINE if:

- The index is NOT disabled (*more on this coming up*)
- 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
- A disabled nonclustered index that does not have a disabled clustered index CAN be rebuilt online

Microsoft

Index Maintenance

How ONLINE Index Rebuilds Work

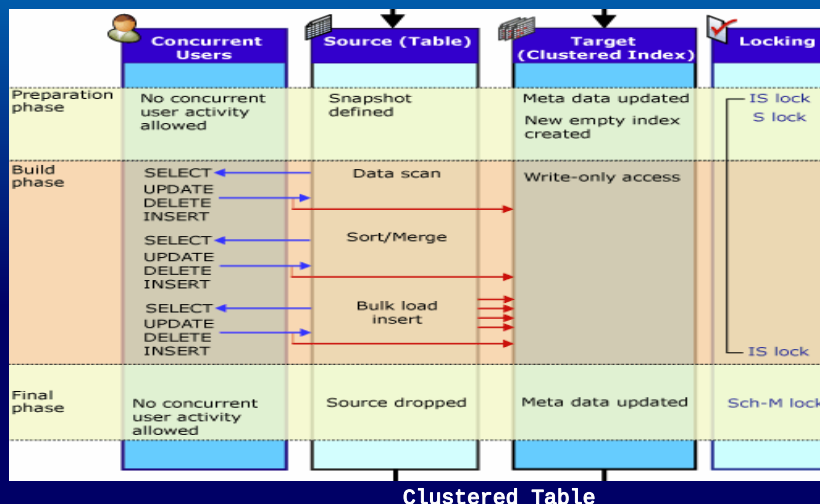
- Initial index "snapshot" generated (does not use or require snapshot isolation)
- The "source" index is maintained and queries are directed at "source" for index usage/optimization
- The "target" index is the index being rebuilt and is only used internally for the rebuild process
- Updates are directed at BOTH the source and the target during the index rebuild
- Once rebuilt there's a very slight pause to redirect access ALL requests to target index
- Source index can be dropped

Microsoft

Index Maintenance

How ONLINE Index Rebuilds Work

HEAP to Clustered Index



Microsoft

Diagram from BOL Topic: How Online Index Operations Work

Online Progress Report

The screenshot shows the SQL Profiler window with a trace running. The trace shows the execution of an online index rebuild. The progress report is as follows:

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

The SQL Profiler window also shows the SQL text: `alter index test on member rebuild with (online = on); go`. The status bar indicates "Trace is running." and "Connections: 1".

Microsoft

Index Maintenance

Special Considerations for ONLINE Rebuild

- Concurrent activity is impacted for very brief period at start and completion of rebuild
- Performance will be impacted as changes are directed to two indexes
- Changing from a HEAP to a Clustered Table or from a Clustered Table to a HEAP causes an internal "mapping index" to be created to allow "conversion" for row lookup between the "source" and "target" indexes
- Only one online index operation per table

Error 1912: Could not proceed with index DDL operation on x because it conflicts with an online operation that is already in progress on the object.

Microsoft

Index Maintenance OFFLINE Rebuild

- Indexes are rebuilt OFFLINE when:
 - DBCC DBREINDEX is used
 - The index contains any LOB Data types
 - The index is partitioned
 - The non-clustered index is disabled or the non-clustered index is based on a table whose clustered index is disabled
- **TIP:** For HA, you might consider logical/vertical partitioning for some LOB data if base table access (non-LOB columns) should always be ONLINE and LOB data is not always needed.
- **TIP:** To reduce the amount of disk space needed for OFFLINE non-clustered index rebuilds, disable the index before rebuilding. To rebuild an index ONLINE, do not disable it.

Microsoft

Index Maintenance OFFLINE LOB_COMPACTION

- SQL Server 2000
 - Large objects can become fragmented
 - DBCC SHOWCONTIG could show it – with ExtentScanFragmentation on IndID 255
 - No commands compact freed space within BLOBs without exporting/importing
 - For more info: Blog entry, Wednesday, August 25
- SQL Server 2005
 - When a clustered index is rebuilt – all LOB columns within the table are compacted
 - When a non-clustered index that includes LOB data types is rebuilt – all LOB columns within the leaf level of the index are compacted

Microsoft

Disabling an Index

- Prevents access to the index
- All data for non-clustered index is deleted
- All data for clustered index is inaccessible, non-clustered indexes are automatically disabled
- A disabled index can be rebuilt or dropped. To enable the index, rebuild it
- DBCC CHECKDB and DBCC INDEXDEFRAG do not act on disabled indexes
- Use ALTER INDEX...REBUILD to rebuild a disabled index, DBCC DBREINDEX can rebuild a disabled index

Microsoft

Disabling an Index

Secondary Impact and Rebuilding

- Disabling the index also disables the constraint
 - Foreign Key constraints which reference the disabled constraint will automatically be disabled with warning
 - Foreign Key constraints must be **manually** enabled using ALTER TABLE CHECK CONSTRAINT
- Disabling a clustered index also disables the non-clustered indexes on the table
 - Rebuilding the clustered index does NOT rebuild the non-clustered unless ALTER INDEX ALL REBUILD specified
 - Use ALTER INDEX REBUILD to rebuild the non-clustered indexes

Microsoft

Rebuild or Defrag

- Rebuild when:
 - AvgFragmentation > 30%
 - You have a dedicated maintenance window – why not?! ☺
- Defrag when:
 - AvgFragmentation <= 30%
 - You cannot rebuild online...
- How do you know?
 - SQL Server 2000
 - DBCC SHOWCONTIG(tablename)
 - SQL Server 2005
 - sys.dm_db_index_physical_stats

Microsoft

Analyzing Fragmentation

sys.dm_db_index_physical_stats

- SQL Server 2000 – DBCC SHOWCONTIG
 - Non-obvious command
 - Hard to programmatically analyze default output
 - Need to pre-create temp table to store/analyze
- SQL Server 2005 – DM object
 - Multi-statement Table-valued function
 - Can use SELECT INTO to create temp table for programmatic analysis (**great feature of DM Objects!**)
 - Can run in multiple modes, target specific tables and/or indexes

```
SELECT * FROM
sys.dm_db_index_physical_stats
(TableNames, IndexName, PartitionNumber, Mode)
```

Microsoft

Analyzing Fragmentation

sys.dm_db_index_physical_stats

- **TableName** = [DEFAULT | NULL | 'TableName']
 - DEFAULT/NULL: return ALL base data: CL, Heap, LOB
 - TableName: sysname type. Can be 1/2/3-part qualified
- **IndexName** = [DEFAULT | NULL | '*' | 'IndexName']
- **PartitionNumber** = [DEFAULT | NULL | #]
 - DEFAULT/NULL/0: return ALL partitions
 - #: returns only the details about specific partition. For VLDB case, only analysis active read/write partition
- **Mode** = [DEFAULT | NULL | 'SpecificMode']
 - DEFAULT/NULL/LIMITED: return FAST scan and only block eXclusive TABLE locks and schema changes.

Microsoft

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.
- **DETAILED: S Lock.** Entire table analyzed.

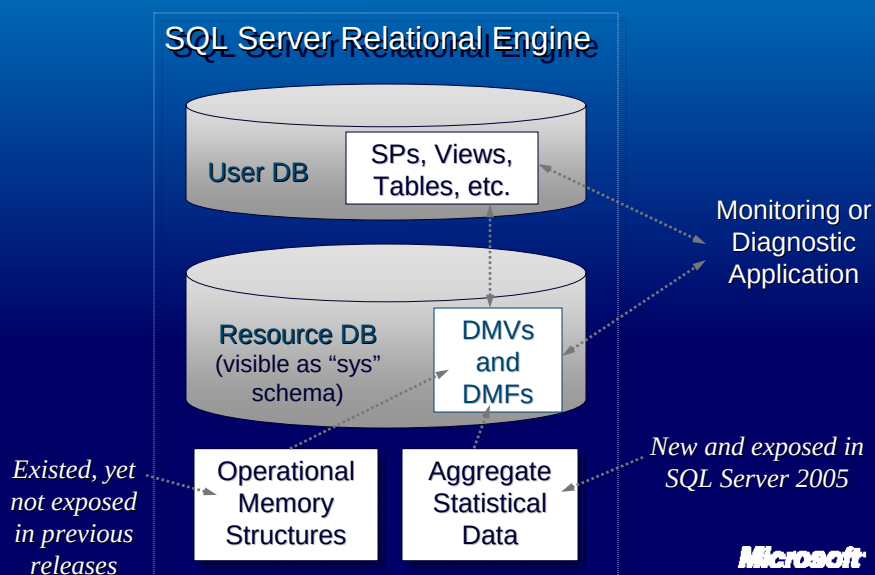
Microsoft

DM Objects – Catalog Views

- Replacement for variety of system tables
 - Catalog Views replace system table access (not just index fragmentation but mostly about system health/perf)
 - Live in resource database
- Compatibility Views
 - Used to simulate common system table access
 - Should be re-written to appropriate Catalog View
- INFORMATION_SCHEMA Views
 - Exist for ANSI SQL-99 Access
- Stored Procedures (sp_), functions (ex. OBJECTPROPERTY, DATABASEPROPERTYEX, SERVERPROPERTY) and views should still work!

Microsoft

DM_ Object Interface



Microsoft

General Server DM Objects

- **dm_db_***
 - Databases and database objects
- **dm_exec_***
 - Execution of user code and associated connections
- **dm_os_***
 - Memory, locking & scheduling
- **dm_tran_***
 - Transactions & isolation
- **dm_io_***
 - Input/Output on network and disks

Microsoft

Finding DM Objects

Three object types:

- Inline Table-valued Function
- Multi-statement Table-valued Function
- View

```
SELECT so.*  
FROM sys.system_objects AS so  
WHERE so.name LIKE N'dm_%'  
ORDER BY so.type
```

Microsoft

Using DM Objects

- What type of object is it?

```
SELECT dbo.DMObjectType(N'dm_obj_name')
```

- If it's a function – what are the input parameters, their order and type?

```
SELECT *
FROM dbo.DMFunctionParams(N'dm_obj_name')
ORDER BY ParameterPosition
```

- What's the result set?

```
SELECT *
FROM dbo.DMObjectColumns(N'dm_obj_name')
ORDER BY ColumnPosition
```

Microsoft

To create these procedures use: "DMObjects.sql"

Using DM Objects

- Find out all of this with one simple sproc:

```
EXEC dbo.sp_GetDMObjectInfo N'dm_obj_name'
```

Dynamic Management Object Type										
1	Dynamic Management View									
DMObjectName	ColumnPosition	ColumnHeader	DataType	MaxLength	Precision	Scale				
dm_exec_query_s...	1	sql_handle	varbinary	64	0	0				
dm_exec_query_s...	2	statement_start_o...	int	4	10	0				
dm_exec_query_s...	3	statement_end_of...	int	4	10	0				
dm_exec_query_s...	4	plan_generation_...	bigint	8	19	0				
dm_exec_query_s...	5	plan_handle	varbinary	64	0	0				
sql_handle	statemen...	statemen...	pla...	plan_handle	creation_time	last_execution_time	ex...	total wor...	last wor...	n
0x030004001500...	1324	1628	2	0x05000400...	2005-01-16 19...	2005-01-16 19:53...	1	322	322	3
0x030004001500...	1630	3800	3	0x05000400...	2005-01-16 19...	2005-01-16 19:53...	1	13541	13541	1
0x0300010013E6...	832	988	1	0x05000100...	2005-01-16 16...	2005-01-16 16:59...	1	1117	1117	1
0x0300010013E6...	1706	-1	1	0x05000100...	2005-01-16 16...	2005-01-16 17:32...	4	2166	413	3
0x030004003029...	20314	24848	1	0x05000400...	2005-01-16 19...	2005-01-16 19:53...	1	455	455	4
0x02000000F260...	0	-1	1	0x06000100...	2005-01-16 19...	2005-01-16 19:53...	1	275	275	2
0x020000001E3A...	98	-1	1	0x06000100...	2005-01-16 17...	2005-01-16 17:00...	1	16863	16863	1

To create these procedures use: "DMObjects.sql"

Tuning Query Performance

- `sys.dm_exec_query_stats`
 - One row per query plan currently in the cache
 - Min, max, avg, last for each: Execution time, Physical reads/writes, Logical reads/writes
 - Execution count
 - First and last execution times
 - Number of times query has been recompiled
 - Uses a pointer to refer to `sql_text` (for efficiency)
- `sys.dm_exec_sql_text()`
 - In memory SQL text
 - Use with `sql_text` pointer
- `sys.dm_exec_query_plan()`
 - In memory Execution and Query Plans
 - Use with `plan_handle` pointer

Microsoft

Statistics Changes

- Improved Wildcard pattern matching
 - Better `%comp%` pattern matching
 - `DBCC SHOW_STATISTICS`, String Index = YES
- Update statistics
 - Column-based (COLMODCTR) rather than row-based (ROWMODCTR); however, counter is not exposed.
 - Significant row modifications (where a specific column is not being modified) will not cause statistics to be invalidated early

Microsoft

INCLUDE Non-key Columns

- Leaf level of index can include non-key columns
- Index key limited to 900 bytes/16 columns – this is to keep tree structure optimal/small
- Allows more covering indexes

Microsoft

Database Tuning Advisor

- Index Tuning Wizard replaced by DTA
- Tuning recommendations for Indexes and Indexed Views
- Adds:
 - Partitioning recommendations
 - Time-bound tuning
 - Indexes with Included columns -> more efficient covering
 - XML Input/Output
 - Drop ONLY mode
 - Parameterized command line execution

Microsoft

XML Indexes

- Primary Index:
 - Requires a clustered primary key on the base table
 - B-Tree of element and attribute names, node values, and node types, retains document order and structure, as well as the path from the root of the XML instance to each node for efficient evaluation of path expressions
- Secondary Indexes off of the primary XML index
 - PATH → columns (path, value)
 - PROPERTY → columns (PK, path, value)
 - VALUE → columns (value, path)

Microsoft

Full Text Indexes

- Performance and Scalability
 - Indexing performance: order of magnitude improvement
 - 20 M Rows on 2000 → 14 days to populate
 - 20 M Rows on 2005 → 10 hours to populate
 - 30-50% perf speed-up on most Full-Text queries
 - Scale: no arbitrary limits, can scale to 2B rows per catalog
- Integration
 - Native Backup, Restore and Recovery
 - Transportability via attach/detach
- Functionality for DEV and DBA
 - Thesaurus support
 - Ignore noise words

Microsoft

Review

- Maintenance Improvements
 - Online operations
 - LOB Compaction
- Disabling Indexes
- Fragmentation Analysis Improvements
- DM Objects for Performance
- Statistics Changes
- Include Non-key Columns for better covering
- Index Tuning – Database Tuning Advisor
- XML Indexes
- Full Text Indexes

Microsoft

Resources

- Check out www.SQLskills.com for information about upcoming **SQL Immersion** events, useful downloads and event scripts. All of the scripts used in this presentation are available.
- Read my blog:
<http://www.SQLskills.com/Blogs/Kimberly/>
- Subscribe to SQLskills:
<http://www.sqlskills.com/login.aspx>
- MPress: *SQL Server 2000 High Availability*
Authors: Allan Hirt with Cathan Cook,
Kimberly L. Tripp and Frank McBath
ISBN: 0-7356-1920-4



Resources

- *Improving Performance w/SQL Server 2000 Indexed Views*
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsql2k/html/indexedviews1.asp>
- *Statistics Used by the Query Optimizer in SQL Server 2000*
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsql2k/html/statquery.asp>
- *SQL Server 2000 Index Defragmentation Best Practices*
<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/ss2kidbp.msp>
- *Using Partitions in a Data Warehouse*
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsql2k/html/partitionsindw.asp>

Microsoft

Resources

- See www.microsoft.com/sql for all sorts of stuff!
- Have the latest BOL? See www.microsoft.com/sql, Technical Resources, Product Documentation
- From Books Online "Home Page" select White Papers to get to full list of SQL Server resources on MSDN or go to <http://msdn.microsoft.com/sql/>
- For TechNet articles use:
<http://www.microsoft.com/technet/prodtechnol/sql/default.asp?frame=true>
- MSDN Webcasts:
<http://support.microsoft.com/default.aspx?PR=pwebcst&FR=0&SD=MSDN&LN=EN-US&CT=SD&SE=NONA>
- Support Resources: <http://www.microsoft.com/sql/support/default.asp>

Microsoft

