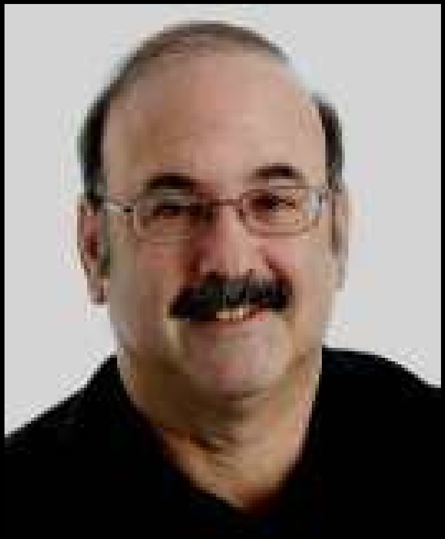
Faster T-SQL sprocs and functions: Lessons Learned

Andy Novick

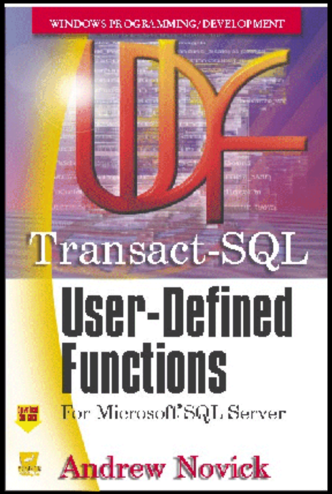
www.NovickSoftware.com

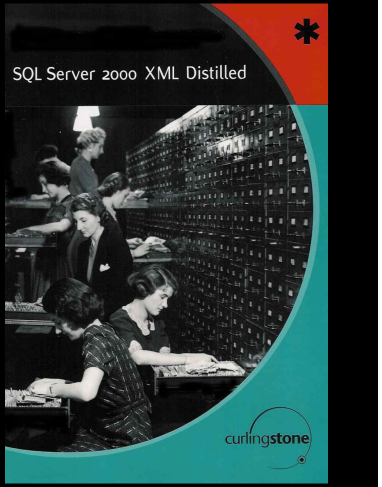
MVP Microsoft® Most Valuable Professional

Introduction



- Andy Novick – Novick Software, Inc.
- Business Application Consulting
  - SQL Server
  - .Net
- www.NovickSoftware.com
- Books:
  - Transact-SQL User-Defined Functions
  - SQL 2000 XML Distilled

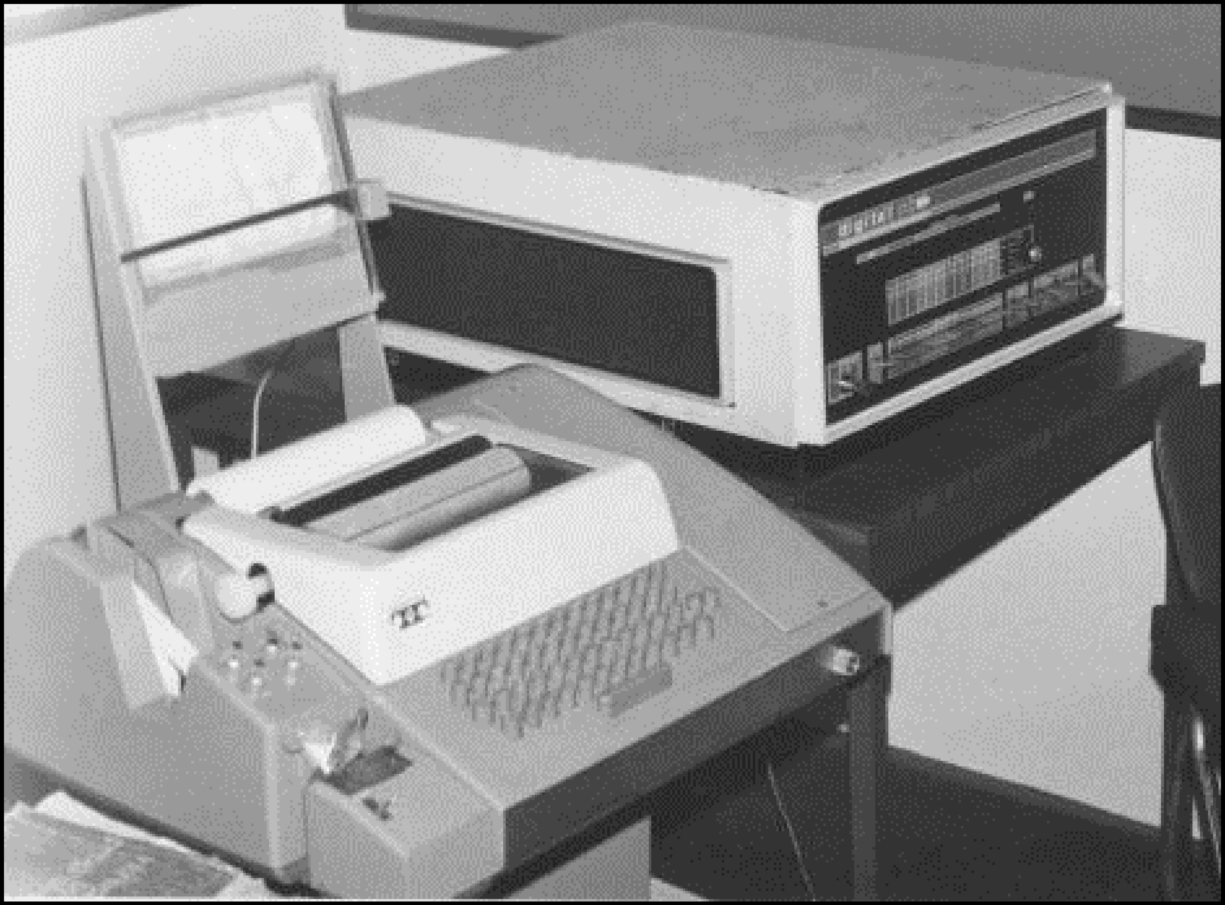


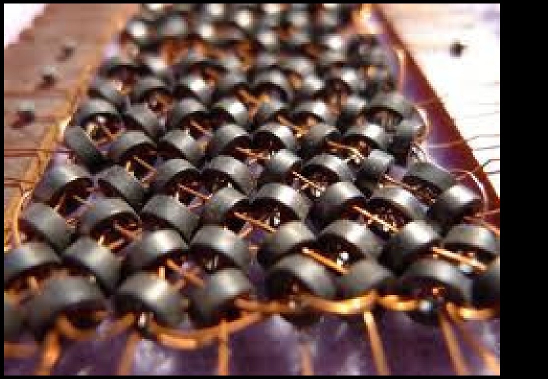


40 years of computing

*September 1971

*PDP-8e and
*KSR33 Teletypewriter

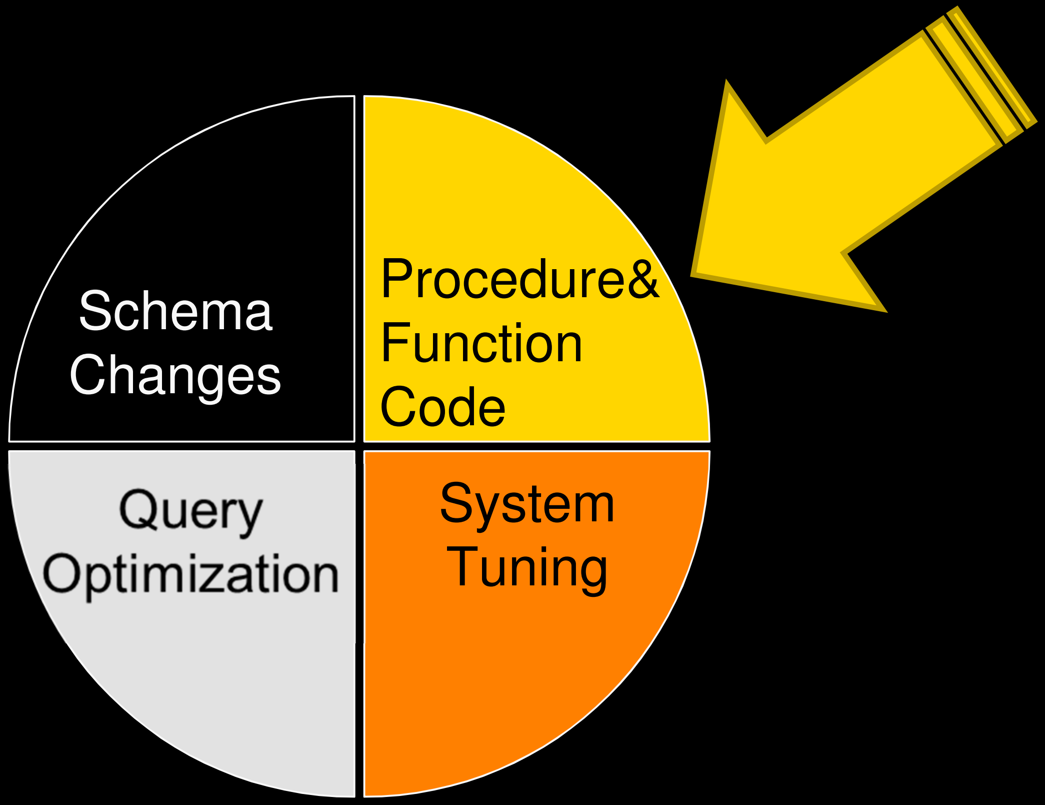




Agenda

- Introduction and Agenda
- Finding out where your procedure spends it's' time
- Which functions should you change
- Rewriting functions for performance
- Working with tempdb
- Other code tricks
- Better Logic – The old fashion way

What we’re talking about today!



#1

Know where the time going?

Ways to figure out where the time is going

- Profiler/Server traces
- Extended Events
- DMV's
  - sys.dm_exec_procedure_stats
  - Sys.dm_exec_query_stats
- Write your own logging code.
- At the detailed level: SSMS – Include Actual Query Plan

Profile/Server Trace

- Relatively high overhead
- Server traces are much lower overhead
- There can be “Too much” detail
- The defaults aren’t always the best
- Best for development systems

Better Tracing – 3 level approach

- High Level - What sprocs are being executed
  - Template - Minimal RPC and Batch
  - Suitable for production
  - Use as a server side trace

- Medium Level – sproc call tree
  - Ns_trace_call_tree
  - Development tool

- Detailed
  - Template – Detaiiled Partial Columns Ordered with Plans
  - Development ONLY!

Demo – Better tracing

Better Tracing – User Templates

- Standard Template
  - Difficul to find where the code is coming from

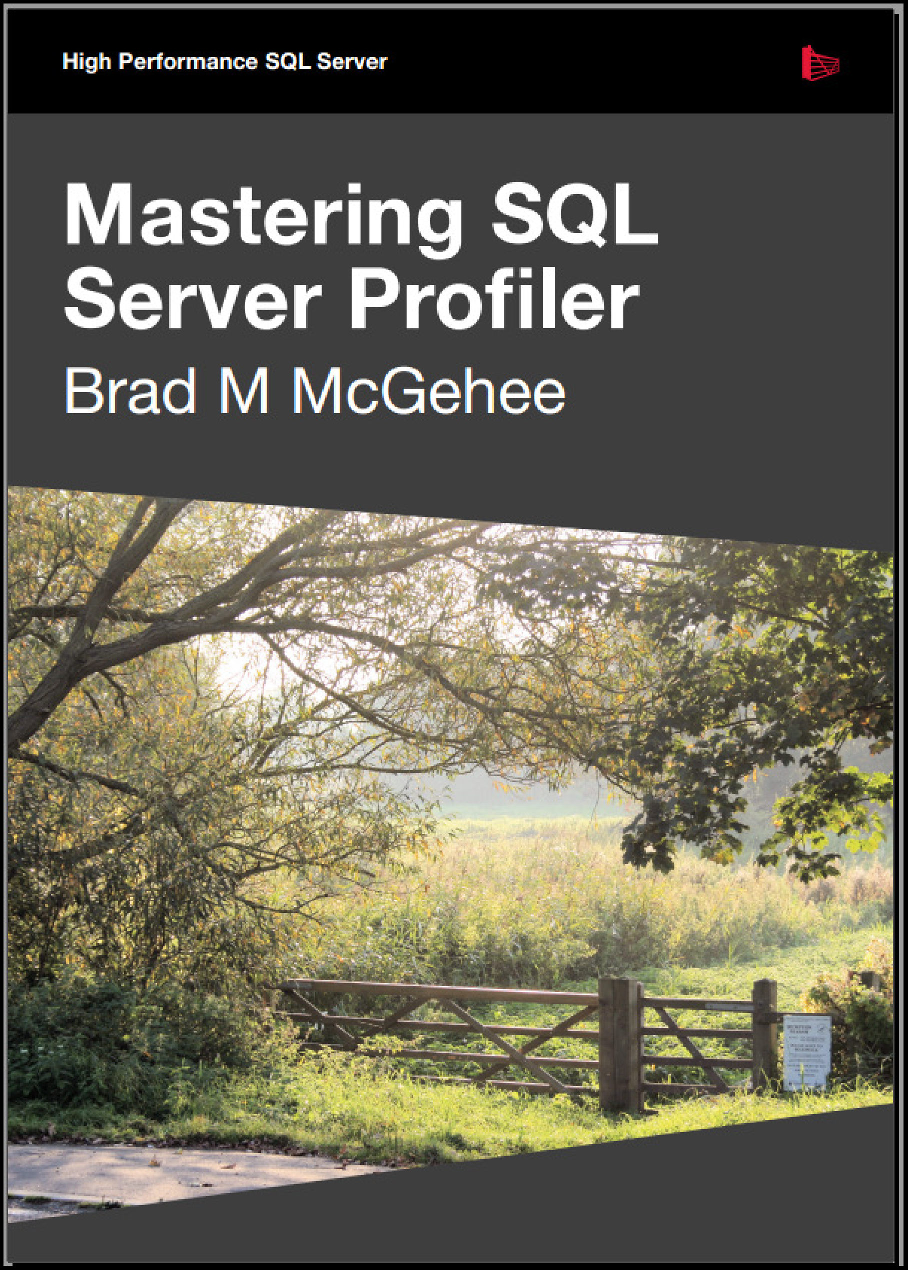
- User templates are in
  - C:\Users\<yourusername>\AppData\Roming\Microsoft\SQL Profiler\10.0\Templates\Microsoft SQL Server\1050\’

Better Tracing – Loading a Trace file

- Select * into mytracetable
  FROM fn_trace_gettable(‘c:\temp\mytrc.trc’), 999)

- Join to get event names from

  sys.trace_events


High Performance SQL Server
Mastering SQL Server Profiler
Brad M McGehee

Extended Events

- The new tool
- Configuration and use can be somewhat difficult.
- Very low overhead
- Can be used “With Care” on production systems
- Denali/2012 includes all the events that SQL Trace has.

DMV’s

- More than 400 DMV’s available
- Procedure stats and query stats
- Query plans (before execution) available
- Are cumulative from Instance startup

Performace DMV’s

- Sys.os_dm_exec_procedure_stats

- Sys.os_dm_exec_query_stats

Write logging code

- That’s a lot of extra work!
- The logging can slow the procedures
- Getting around “Transactions” is tricky
- If done well, can be left in the code for production.

Functions

- Scalar functions are slow for several reasons
  - Row-by-Row cursor like processing
  - They Inhibit parallelism!
- Table Valued Functions are slow for several reasons
  - Row-by-Row cursor like processing
  - Inhibit Parallelism
  - Use of a Table Variable to return data
- Inline Functions are fast
  - They're a Views with parameters and are in-lined into the query.

Which Function Should be Rewritten?

- Functions are not recorded in dm_exec_procedure_stats

- Traces can record functions SP:Procedure Complete
  - Filter on ObjectType = 20038
  - Overhead can be high!

- Extended Events is the low overhead way to measure function use.

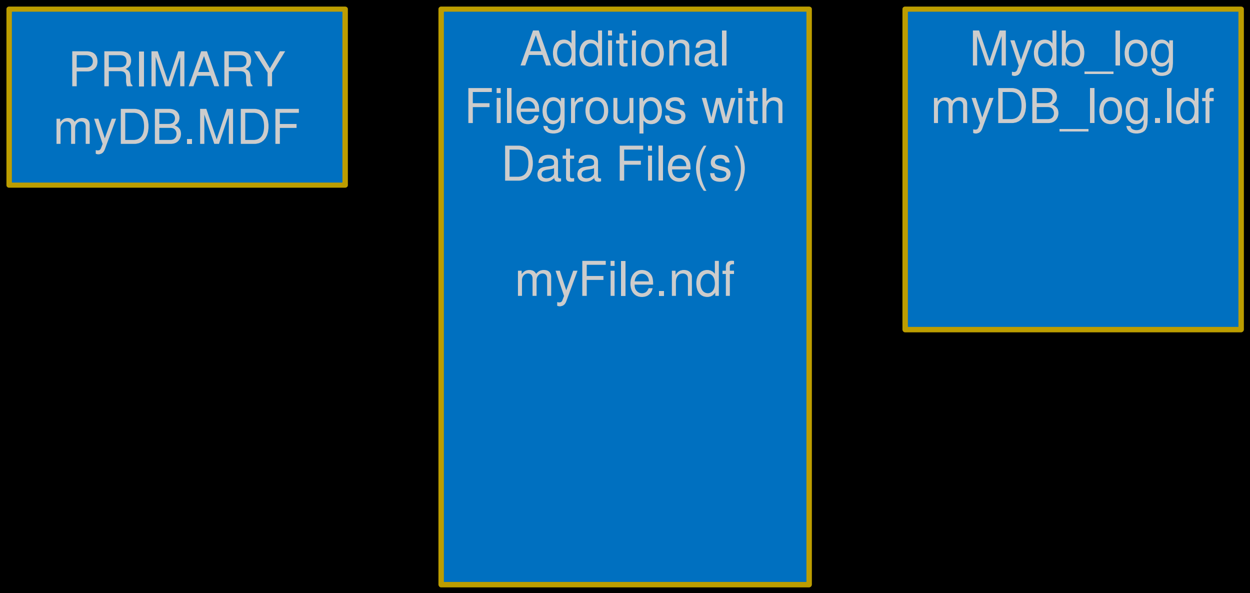
Demos – Functions
Overhead and Parallelism

Demo – Extended Events
Rewriting Functions

ACID Properties

Atomicity	Each transaction is “all or nothing”
Consistency	Database moves from one valid date to another
Isolation	Ensures that concurrent execution results in the same state as serial transactions
Durability	Once a transaction is committed it remains so even if there are crashes, errors, power loss

How is ACID achieved?



PRIMARY
myDB.MDF

Additional Filegroups with Data File(s)

myFile.ndf

Mydb_log
myDB_log.ldf

What does every manager/client want?

The Magic “Go Faster” Switch

-T610

Minimal Logging

- You're already familiar with it:
  - BCP
  - Bulk Insert
  - SELECT …. INTO ……

- 2K8 added more
  - INSERT INTO …. WITH TABLOCK SELECT ….
  - Trace Flag 610

Minimally Logged Requirements

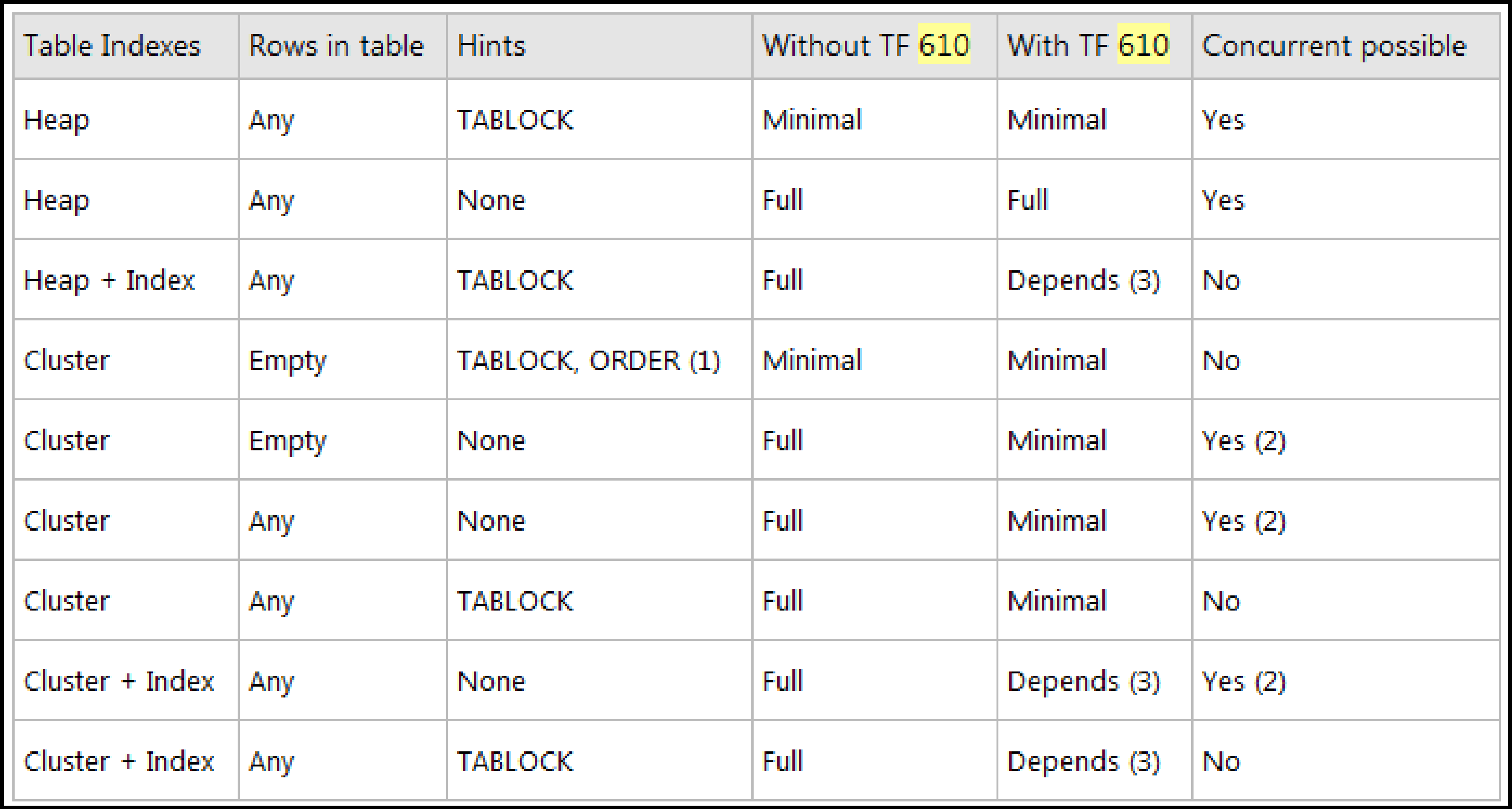


Table Indexes	Rows in table	Hints	Without TF 610	With TF 610	Concurrent possible
Heap	Any	TABLOCK	Minimal	Minimal	Yes
Heap	Any	None	Full	Full	Yes
Heap + Index	Any	TABLOCK	Full	Depends (3)	No
Cluster	Empty	TABLOCK, ORDER (1)	Minimal	Minimal	No
Cluster	Empty	None	Full	Minimal	Yes (2)
Cluster	Any	None	Full	Minimal	Yes (2)
Cluster	Any	TABLOCK	Full	Minimal	No
Cluster + Index	Any	None	Full	Depends (3)	Yes (2)
Cluster + Index	Any	TABLOCK	Full	Depends (3)	No

White Paper: The Data Loading Performance Guide – on MSDN

Minimally Logged Index Operations

- CREATE INDEX
- ALTER INDEX REBUILD or DBCC DBREINDEX
- DROP INDEX (to a heap)

Demo
MINIMAL LOGGING

Other Candidates for “Magic Switch”

- -T1118 – Forces Uniform Extent allocations in tempdb
  - http://blogs.msdn.com/b/psssql/archive/2008/12/17/sql-server-2005-and-2008-trace-flag-1118-t1118-usage.aspx

- -T4199 – Turns on some Query Optimizer changes
  - http://support.microsoft.com/kb/974006

- -T2301 – Spend more time optimizing complex queries
  - http://blogs.msdn.com/b/ianjo/archive/2006/04/24/582219.aspx

WORKING WITH TEMPDB

Why tempdb?

- Is it faster than a user database?
- Are pages even written to disk?
- When Is it used?

TEMPDB is Faster

- It’s not recovered!
  - No “Durability”
- Minimal Logging, usually just allocations
- Checkpoint doesn’t write the data pages to disk
- Fewer Log Records are written
  - i.e Updates – no after image rows needed

Demo – tempdb is Faster!

Table Variables

```
DECLARE @myTab as (a int primary key, b varchar(30))
```

- Best used at the interface level where required
  - Stored procedure parameters
  - Table-Valued-Function parameters

- Limited indexing and not statistics
- OK for passing a small number of rows between procs

- Prefer temp tables over temp variables when possible

#temp tables

- Are best most of the time.
- Particularly when the number of rows gets over 3000
- Can be indexed and have statistics
- Reused between function executions
- Best most of the time

Temp table caching

- Since SQL Server 2005 SP3, SQL Server caches a few pages of temp table definitions by renaming them.
- Saves around 2 milliseconds / table per procedure call.
- Important in very high activity sites.
- Restrictions
  - No DDL allowed on the table except “Drop Table …”
  - No named constraints
  - Less than 8 MB

EVERY TABLE SHOULD HAVE A PRIMARY KEY! RIGHT?

How about a clustered index?

LEARN TO LOVE THE HEAP!

© Alfred Molon www.molon.de

RAVENS
86
HEAP
86

Indexes on #temp tables?

- Great if they actually get used!
- If the index is used once for sorting. Consider a heap
- If the table has multiple inserts:

  - Create the index after the inserts
  - Before any use
- If the table has only one insert:

  - Create the index when the table is created

Demos – tempdb caching clustered indexes

SARGable searches

SARG able searches

- Watch for implicit conversions

  CONVERT_IMPLICIT() in query plans

- Types must match to use indexes without conversion.

  WHERE au_lname LIKE N’Bronson%’

- Watch out for string searches that require scans

  WHERE   au_lname LIKE ‘%NOVICK%’

BETTER LOGIC

Search for better logic

- Don’t do what does not have to be done.
- Don’t move data around that isn’t needed
- Use T-SQL for what it’s good at and no more.

Use T-SQL for what it’s good at

- SQL Server is great at array calculations

- Not so good when a CURSOR is required

Some quick and easy things to do

- Set NOCOUNT ON

- Schema qualify all object references

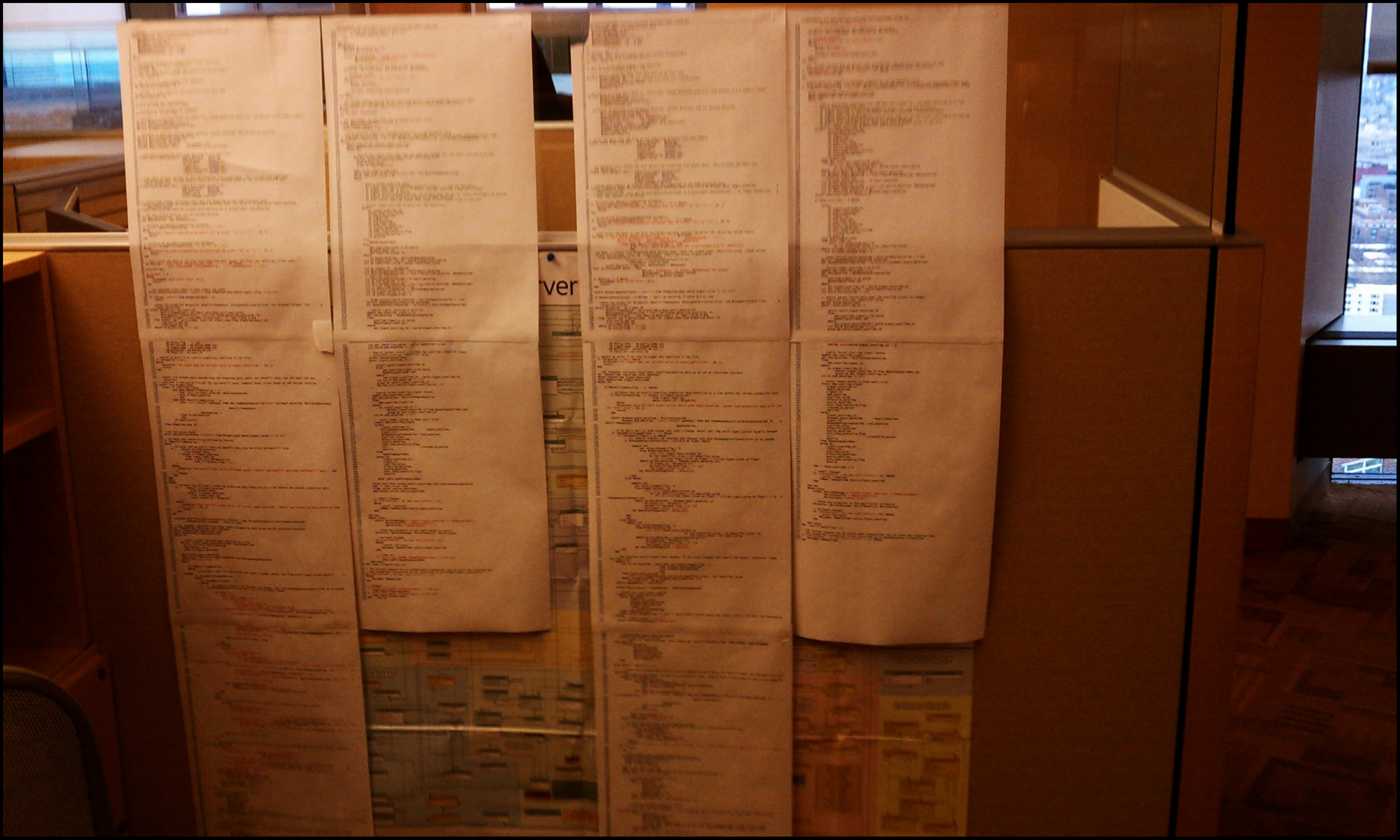
  FROM authors → FROM dbo.authors

- Never begin the name of a procedure with sp_

  sp_GetMyData → MyDataGet

- Minimize DDL and move it to the front of the procedure.
  - Not as important as it use to be

Live with your code!



Andy Novick On-Line

- What’s new in SQL Server 2008 R2 and

  SearchSQLServer.com

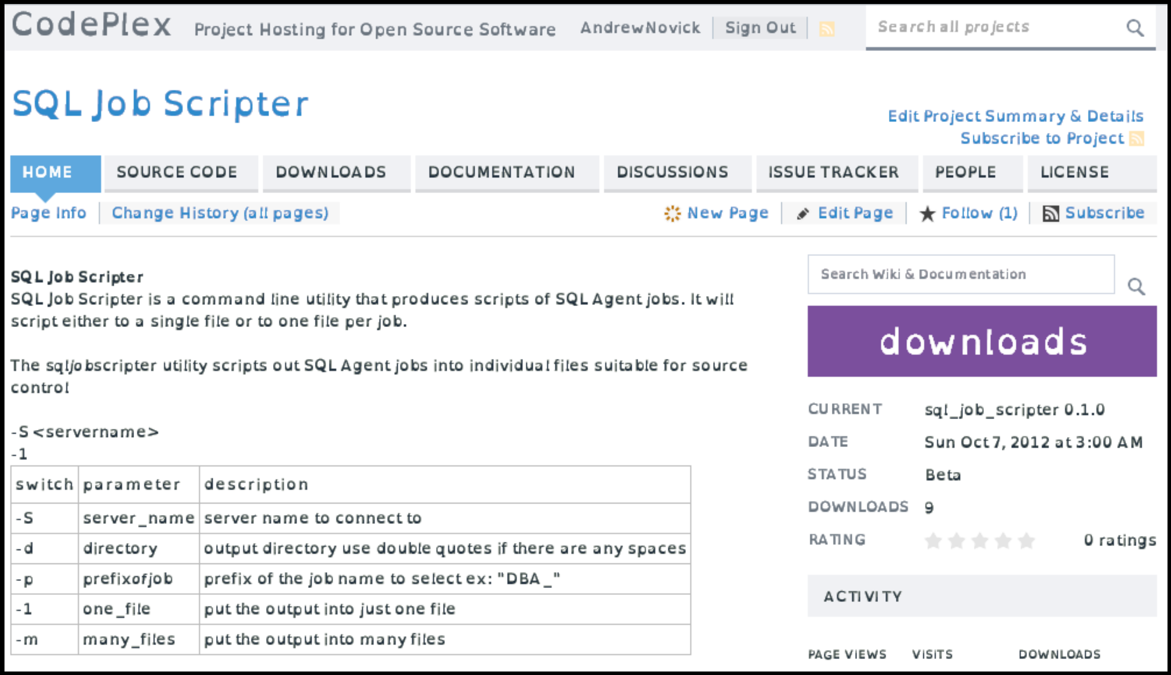
- Developing and Testing Databases in Visual Studio 2011 (Data Dude)

  DNRtv.com

- http://www.NovickSoftware.com

SQL Job Scripter

- Script out SQL Agent Jobs
- Open source
  - https://sqljobscripter.codeplex.com/



SQL Job Scripter

SQL Job Scripter is a command line utility that produces scripts of SQL Agent jobs. It will script either to a single file or to one file per job.

The sqljobscripter utility scripts out SQL Agent jobs into individual files suitable for source control

-S <servername>
-1

switch	parameter	description
-S	server_name	server name to connect to
-d	directory	output directory use double quotes if there are any spaces
-p	prefixofjob	prefix of the job name to select ex: "DBA_"
-1	one_file	put the output into just one file
-m	many_files	put the output into many files

CURRENT	sql_job_scripter 0.1.0
DATE	Sun Oct 7, 2012 at 3:00 AM
STATUS	Beta
DOWNLOADS	9
RATING	☆☆☆☆☆ 0 ratings

New England Microsoft Developers

- 1st Thursday of every month @ 6:30
- 201 Jones Rd. Waltham, MA - 1st floor

- .Net Focus with other technologies in the mix
  - Nov 1 - Visual Studio 2012 – Ben Day
  - Dec 6 – MVC vs Web Forms – Carl Bergenhem
  - Jan 3 – Building Windows Store Applications – Kevin Ford

- http://www.meetup.com/NE-MSFT-Devs

SQL PASS 2012
The Biggest Loser – Database Edition

Thanks for Coming

Andy Novick

▪

anovick@NovickSoftware.com

▪

www.NovickSoftware.com