

New York .NET & SQL Server User Groups

SQL Server: Are you talking to me?

Service Broker & Plan Explorer by Jonathan Kehayias
Statement Execution & Plan Cache by Kimberly L. Tripp



Our Hosts!

- **Microsoft**
 - Jesus Aguilar
 - <http://www.giventocode.com/>
- **SQL Server User Group**
 - Joe Lax
 - <http://nycsqlusergroup.com/>
- **.NET User Group**
 - Andrew Brust
 - <http://www.nycdotnetdev.com/>

- **Pluralsight** – www.pluralsight.com
 - Online training provider
 - More than 1,000 IT and dev courses – on demand, and on mobile devices!
 - SQLskills online training partner with 31 courses and over 100 hours of deep, technical training
 - Directly relevant to tonight:
 - Coming soon – courses on SQL Server Service Broker
 - [SQL Server: Using SQL Sentry Plan Explorer](#) by Jonathan Kehayias
 - [SQL Server: Optimizing Ad Hoc Statement Performance](#) by Kimberly L. Tripp
- **SQL Sentry** – www.sqlsentry.net
 - SQL Server Software vendor
 - Graphical Plan Analysis – **Plan Explorer (FREE)**, Plan Explorer Pro
 - Server performance – Performance Advisor (SQL and BI), Fragmentation Manager
 - Event Management – SQL Server, Oracle, SharePoint, and Windows

Our Conference: SQLintersection

Our Spring Show Lineup Speaks for Itself!

April 12-17, 2014 in Orlando, Florida



SQL POWER
intersection TO CREATE

April 13-16, 2014 JW Marriott Orlando, FL

"I'll be there"

"Show package" attendees who register early choose a Surface 2 or XBOX ONE.

203.264.8220

Powered by: Microsoft NextGen



Use this discount code
and save \$50

SQLskills

■ 50 SQL Sessions, 3 SQL keynotes, and 7 workshops:

- Saturday, April 12 – two pre-conference workshops:
 - ***50 Things All SQL Server Developers Need To Know!*** with Kevin Kline and Aaron Bertrand
 - ***What's New in SQL Server 2014*** with Bob Beauchemin
- Sunday, April 13 – three pre-conference workshops:
 - ***Developer's Guide to SQL Server Operations*** with Jeremiah Peschka and Kendra Little
 - ***Practical Disaster Recovery Techniques*** with Paul Randal
 - ***Queries Gone Wild: Real-world Solutions*** with Kimberly L. Tripp
- Thursday, April 17 – two post-conference workshops:
 - ***Make SQL Server Apps Go Faster*** with Brent Ozar, Jeremiah Peschka, and Kendra Little
 - ***Windows Azure SQL Database from A to Z*** with Bob Beauchemin

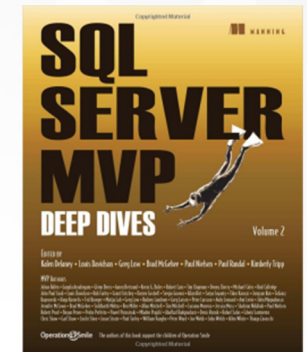
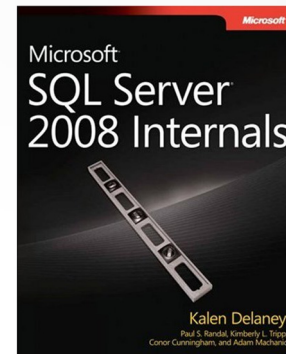
■ Industry-experts and Microsoft speakers, Sunday Eve-Wednesday, April 13-16

- Sessions on performance tuning, troubleshooting, coding/development, query tuning, architecture, fragmentation, & much, much more – learn real-world solutions & bring back immediate ROI
- Learn how to take advantage of SQL Server 2014 – Hekaton, Columnstore, and HA/VM/Hybrid with an entire track on SQL Server 2014 from early users & the SQL Server Development Team

See more information online at www.SQLintersection.com

Author/Instructor: Kimberly L. Tripp

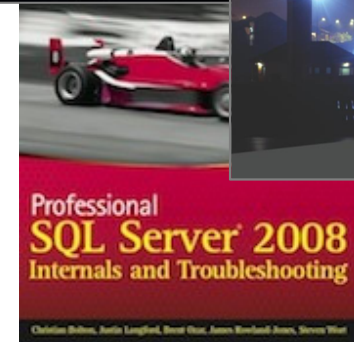
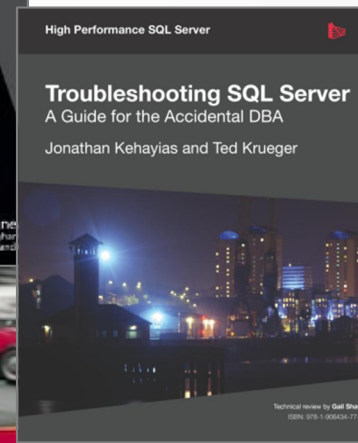
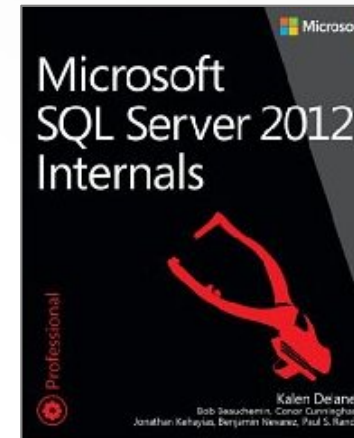
- Consultant/Trainer/Speaker/Writer
- President/Founder, SYSolutions, Inc.
 - e-mail: Kimberly@SQLskills.com
 - blog: <http://www.SQLskills.com/blogs/Kimberly>
 - Twitter: @KimberlyLTripp
- Author/Instructor for SQL Server Immersion Events
- Instructor for multiple rotations of both the SQL MCM & Sharepoint MCM
- Author/Manager of SQL Server 2005 & 2008 Launch Content
- Author/Speaker at Microsoft TechEd, SQL Connections, SQLPASS, ITForum, TechDays
- Author of several SQL Server Whitepapers on MSDN/TechNet: Partitioning, Snapshot Isolation, Manageability, SQLCLR for DBAs
- Author/Presenter for more than 25 online webcasts on MSDN and TechNet
- Co-author MPress Title: SQL Server 2008 Internals, the SQL Server MVP Project (1 & 2), and SQL Server 2000 High Availability
- Presenter/Technical Manager for SQL Server HOL DVDs
- Owner and Manager of the SQLIntersection conference



Author/Instructor: Jonathan Kehayias



- Consultant/Trainer/Speaker/Author
- Principal Consultant, [SQLskills.com](http://www.SQLskills.com)
 - Email: Jonathan@SQLskills.com
 - Blog: <http://www.SQLskills.com/blogs/Jonathan>
 - Twitter: @SQLPoolBoy
- Contributing Author: *SQL Server 2008 Internals and Troubleshooting*, *Microsoft SQL Server 2012 Internals*
- Microsoft Certified Master: SQL Server 2008
- SQL Server MVP since October 2008
- Author of Using Extended Events whitepaper on MSDN
- Developed Extended Events Manager SSMS Addin for SQL Server 2008 Open Source project on Codeplex
- Member of the Tampa SQL Server User Group and PASS



Overview

- **Jonathan Kehayias**
 - **Part I: Service Broker**
 - **Part II: SQL Sentry Plan Explorer**
- **Kimberly L. Tripp**
 - Part III: Statement Execution
 - Part IV: Plan Cache
- **Raffle for prizes**
 - One SQL Sentry Plan Explorer Pro License
 - Two Pluralsight Annual Plus Subscriptions
 - One voucher for \$100 off (per day) for an Immersion Event of your choice (maximum value \$500 for a 5-day event)
 - Everyone – 30-day trial subscription to Pluralsight
- **Open Q & A**

Service Broker Usage Scenarios

- **Centralized data collection**
 - Multiple sites process transactions locally and use Service Broker to send information to a central office
 - Asynchronous message delivery allows local sales transactions to continue even if a site loses connectivity to the central office
- **Asynchronous trigger execution**
 - Complex logic in DML triggers affect OLTP performance
 - Queuing a message requesting work from a Service Broker service to offload the complex logic into a separate transaction asynchronously
- **Reliable processing of queries**
 - Single transaction is used to read a message, run a query, and return results to the initiating service
 - During a service failure the transaction would roll back, return the message to the queue, and restart processing after recovery

Service Broker Usage Scenarios (2)

■ Distributed server-side processing

- Online ticketing application processes orders using Service Broker to exchange data between payment processing, CRM, and ticket printing applications
- Allows orders to continue to be accepted even if the payment processing, CRM, or ticket printing application is offline for maintenance

■ Batch processing changes

- Applications store data to be processed into a queue within Service Broker allowing a program to periodically read and process the data
- Parallel processing from the queue can handle larger volumes of data quickly and efficiently
- Conversation groups and dialogs bundle data allowing multiple rows to be handled in a single RECEIVE command

Advantages of Service Broker

- **Database integration**

- Eliminates the overhead of a separate distributed transaction coordinator
- The message store and database are maintained at the same transactional point-in-time, eliminating the need for reconciliation processes when one of the two has to be restored

- **Ordering of messages**

- Messages sent within the Service Broker architecture are received once and only once in the order in which the messages were sent

- **Message locking or related messages**

- Service Broker prevents out of order processing of message within the same conversation group by locking the conversation group during processing
- Multiple threads can process different conversation groups concurrently increasing scalability and performance

Advantages of Service Broker (2)

■ Loose coupling of applications

- The initiating application and target application are loosely coupled, allowing the initiating application to continue processing after sending a message to the target application
- Loose coupling provides scheduling flexibility, allowing parallel processing
- Background processing of requests can mitigate high load times against source and target servers through queuing requests for processing

■ Automatic activation

- Activation allows Service Broker to automatically scale to match the volume of messages arriving in a given service queue
- Allows programs running inside the database as well as external to the database to take advantage of queue activation
- Automatically starts a new queue reader when there is work to be performed

Plan Explorer Overview

- **Setup requirements and installation**
- **Comparison to Management Studio**
 - Plan layout
 - Performance issue identification
 - Missing indexes
 - Large execution plans
- **Demos**

Setup Requirements and Installation

- All SQL Server versions from SQL Server 2005 onward are supported
- .NET 4.0 must be installed before installing Plan Explorer
- Plan Explorer can be freely downloaded from the SQL Sentry website:
 - 64-bit download (<http://bit.ly/1aXzJaL>)
 - 32-bit download (<http://bit.ly/1c81EWP>)
 - No information is collected as a part of the download, so nobody will email or contact you in any way about the Free Edition
- Plan Explorer can be integrated directly within SSMS by installing the optional SSMS Add-in (<http://bit.ly/1cNuJER>)
 - Note: Plan Explorer must be installed before the add-in can be installed

Plan Layout

■ Management Studio

- ❑ Large spacing between operators causes wasted space that generally requires scrolling to view the entire plan
- ❑ Zooming in or out through the right click context menu only
- ❑ Object and index names are truncated and require viewing the Properties window or tooltip for the operator to view the full names

■ Plan Explorer

- ❑ Optimizes spacing between operators to minimize wasted space, specifically vertical space, minimizing the need to scroll
- ❑ Dynamic zooming can be performed using a Ctrl + mouse scroll wheel
- ❑ Object and index names are displayed fully for operators on the screen
- ❑ Sortable grids simplify identifying high cost operators in complex large execution plans

Performance Issue Identification

■ Management Studio

- No highlighting of potentially problematic operations
- Important information about possible problems is hidden in tooltips
 - Estimated and actual row counts for cardinality estimation issues
 - Per-thread row counts for parallel tasks for uneven distributions
- Runtime metrics are not collected as a part of the execution plan

■ Plan Explorer

- Performs color-scaling highlighting of expensive operators for easy identification of possible problems
- Plan Tree grid shows important information about operators and highlights potentially problematic issues automatically
- Runtime metrics are collected for actual plans created inside of Plan Explorer to simplify analysis

Missing Index Information

- **Management Studio**

- Missing index information is shown for every statement plan in a multi-statement batch and not just for the statement plan that contains the missing index information
- Only the first missing index group information in the plan is displayed, even if multiple statement plans contain different recommendations

- **Plan Explorer**

- Only shows missing index warnings on the plan where the missing index details were actually included in the plan
- Adding the Missing Indexes column to the Statement Tree simplifies identifying the statement(s) that have missing index information available for review

Large Execution Plans

■ Management Studio

- ❑ Large statement text and execution plans will often freeze SSMS
- ❑ Identifying the most expensive operation in a very large plan is nearly impossible and requires scrolling through every branch looking at operator costs
- ❑ Lacks the ability to show cumulative costs per branch

■ Plan Explorer

- ❑ Optimized to handle large execution plans without performance problems
- ❑ Statement Tree and Plan Tree simplify identification of the most expensive statement and operation within a large complex plan
- ❑ Provides the ability to show cumulative costs per branch to drill down into the most expensive portion of a plan easier
- ❑ Pro Edition features further improve large plan analysis with collapsible parent operators

Overview

- Jonathan Kehayias
 - Part I: Service Broker
 - Part II: SQL Sentry Plan Explorer
- **Kimberly L. Tripp**
 - **Part III: Statement Execution**
 - **Part IV: Plan Cache**
- Raffle for prizes
 - One SQL Sentry Plan Explorer Pro License
 - Two Pluralsight Annual Plus Subscriptions
 - One voucher for \$100 off (per day) for an Immersion Event of your choice (maximum value \$500 for a 5-day event)
 - Everyone – 30-day trial subscription to Pluralsight
- Open Q & A

Different Ways to Execute SQL Statements

- **Ad hoc statements**
 - Possibly, as auto-parameterized statements
- **Dynamic string execution (DSE)**
 - *EXECUTE (@string)*
- **sp_executesql (forced statement caching)**
- **Prepared queries (forced statement caching through “parameter markers”)**
 - Client-side caching from ODBC and OLEDB (parameter via question mark)
 - Exposed via *SQLPrepare / SQLExecute* and *ICommandPrepare*
- **Statements within procedural code**
 - Functions
 - Stored procedures (and triggers)

Demo

Statement Execution & Cost-based Estimates

Statistics [histogram v. density_vector]

Heuristics

Demo: Key Points

- **Estimates come from:**

- Statistics – if they exist or if they can be [auto] created, using:
 - The histogram: when the value can be “sniffed” (literals & parameters)
 - NOTE: Stored procedures and sp_executesql can “sniff” parameters but BOTH save the plan and reuse the plan... even though it might not be good for subsequent executions resulting in PSP (parameter sniffing problems)
 - The density vector: when the value cannot be “sniffed” (variables)
- Heuristics – if there are no statistics available and SQL Server cannot auto create them
 - These are internal rules – sometimes called “magic” numbers (cannot be changed)
 - They often result in very poor plans (LEAVE AUTO_CREATE_STATS ON)

- **Statistics have to be reasonably small to be fast/useful**

- **They're just estimates**

- They're not always guaranteed to be accurate
- They're just meant to get us closer to the right value

Plan Cache Usage/Limits

- The “plan cache” (a.k.a. “procedure cache”)
- Uses “stolen” pages from the buffer pool (data pages)
- View cached plans: `sys.dm_exec_cached_plans`
- Plan cache memory limits:
 - SQL Server 2008+ and SQL Server 2005 SP2
 - 75% of visible target memory from 0-4GB
 - + **10%** of visible target memory from 4Gb-64GB
 - + **5%** of visible target memory > 64GB
 - SQL Server 2005 RTM and SQL Server 2005 SP1
 - 75% of visible target memory from 0-8GB
 - + **50%** of visible target memory from 8Gb-64GB
 - + **25%** of visible target memory > 64GB
 - SQL Server 2000
 - SQL Server 2000 4GB upper cap on the plan cache
 - 32-bit? AWE memory is not “visible” to the plan cache (plan cache has to live in “real memory”)

2008/2005 SP2	
Memory	Plan Cache
4GB	3.0GB
8GB	3.5GB
16GB	4.2GB
32GB	5.8GB
64GB	9.0GB
128GB	12.2GB
256GB	30.0GB

Statement Auto-Parameterization

Much Ado About Nothing!

- Lots of hype, not really a lot of benefits...
- Idea: automatically parameterize search arguments and place statement in cache for future re-use of plan
- Problem: queries are only parameterized and reused when they're "safe"
 - Needs to be a VERY straightforward statement/plan.
 - Many limitations:
 - FROM clause cannot have more than one table
 - WHERE clause cannot have expressions joined by OR
 - WHERE clause cannot have an IN clause
 - Statement cannot contain a sub-query
 - VERY restrictive (see Appendix A in whitepaper for complete list)
 - Parameters do not change plan choice
- Problem: queries that are "unsafe" bloat the cache

Query Plans/Plan Cache Problems

- Plan cache pollution and single-use plans
- Take the following identical query (except for the SARG):
 - SELECT ... FROM member WHERE lastname = 'Tripp'
 - SELECT ... FROM member WHERE lastname = 'Tripps'
 - SELECT ... FROM member WHERE lastname = 'Tripped'
 - SELECT ... FROM member WHERE lastname = 'Tripper'
 - SELECT ... FROM member WHERE lastname = 'Falls'
- Each plan takes 24,385 bytes (or roughly 24KB)
- This “query class” is harder to track because each is listed in the cache...
- Query sys.dm_exec_query_stats for query_hash and aggregate over query_hash, query_plan_hash to see how many plans a particular query might have in the cache... if there's only one then the statement may in fact be stable

The “Cumulative Effect” of Queries

- SQL Server 2008 added `query_hash` and `query_plan_hash` to `sys.dm_exec_query_stats`
 - Aggregate by `query_hash` to find similar queries
 - Aggregate by `query_hash`, `query_plan_hash` to find similar queries along with their plans (possibly more than one per `query_hash` which is why the statement wasn't safe)
- SQL templatises the parameters – similar to `sp_get_query_template`
- See the queries in the blog posts...
- Check out BOL: Finding and Tuning Similar Queries by Using Query and Query Plan Hashes
 - <http://bit.ly/QfUmRY>

Reducing Plan Cache Pollution

- **Server setting: Optimize for ad hoc workloads**
 - On first execution, only a compiled plan **stub** will go into cache. For the prior query this is only 380 bytes (compared to the 24KB plan)
 - Ultimately you will end up with a lot of statements in cache – you'll still need to periodically clear cache
 - On second execution (if), the plan will be placed in cache
- **Create more consistent plans with covering indexes – might make the plan more stable (possibly even safe)!**
 - NOTE: Only single-table statements can be parameterized/safe. There are **a lot** of limitations to SQL Server detecting a stable statement as safe (see Appendix A of the *Plan Caching in SQL Server 2008* whitepaper)
- **If the plans are actually stable...**
 - Use `sp_executesql` or a stored procedure instead
 - Consider database setting: forced parameterization (not as likely – definitely watch the Pluralsight course before you enable this)

Overview

- **Jonathan Kehayias**
 - Part I: Service Broker
 - Part II: SQL Sentry Plan Explorer
- **Kimberly L. Tripp**
 - Part III: Statement Execution
 - Part IV: Plan Cache
- **Raffle for prizes**
 - One SQL Sentry Plan Explorer Pro License
 - Two Pluralsight Annual Plus Subscriptions
 - One voucher for \$100 off (per day) for an Immersion Event of your choice (maximum value \$500 for a 5-day event)
 - Everyone – 30-day trial subscription to Pluralsight
- **Open Q & A**

Overview

- **Jonathan Kehayias**
 - Part I: Service Broker
 - Part II: SQL Sentry Plan Explorer
- **Kimberly L. Tripp**
 - Part III: Statement Execution
 - Part IV: Plan Cache
- **Raffle for prizes**
 - One SQL Sentry Plan Explorer Pro License
 - Two Pluralsight Annual Plus Subscriptions
 - One voucher for \$100 off (per day) for an Immersion Event of your choice (maximum value \$500 for a 5-day event)
 - Everyone – 30-day trial subscription to Pluralsight
- **Open Q & A (including @PaulRandal)**

Potential Topics

- Data/log file management
- Recovery model choice
- VLF fragmentation
- Logging and recovery
- Tempdb/temp tables
- Indexing strategies
- Fragmentation
- Index maintenance
- Query plans
- Plan caching
- DMVs
- Virtualization
- I/O subsystems
- Wait/latch/spinlock statistics
- Locking/blocking/deadlocks
- Statistics
- Database mirroring
- Database snapshots
- Partitioning
- Data compression
- Resource governor
- Extended events
- Performance tuning
- Instant initialization
- Backup compression
- Heaps vs. clustered indexes
- Data type choice
- Columnstore
- Scale-out
- Monitoring

Questions?

