



Using Event Notifications in SQL Server 2005/2008

DBA-313s

Jonathan Kehayias
Principal Consultant
SQLskills.com

About Jonathan Kehayias

- Consultant/Trainer/Author/Speaker
- Principal Consultant, SQLskills.com

Email: jonathan@sqlskills.com

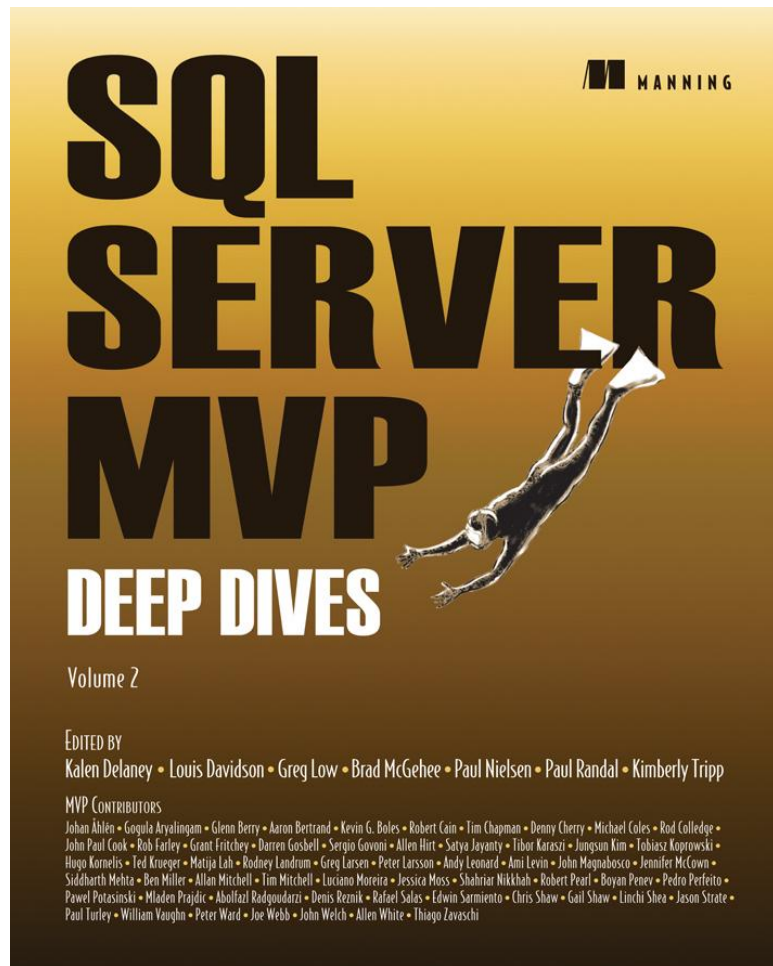
Blog: <http://sqlskills.com/blogs/jonathan>

Twitter: @SQLPoolBoy



- Author: SQL Server 2008 Internals and Troubleshooting
- Author: Troubleshooting SQL Server; A Guide for the Accidental DBA
- Microsoft Certified Master: SQL Server 2008, SQL Server MVP
- Author of Using Extended Events whitepaper on MSDN
- Regular presenter at PASS Summit, SQL Saturdays, SQLBits and SQL Connections conferences
- Developer Extended Events Manager SSMS Addin for SQL Server 2008 Open Source project on Codeplex
- Course instructor for *Microsoft Certified Master – Database and SharePoint* qualifications

SQL Server MVP Deep Dives, Volume 2



www.operationsmile.org
www.manning.com/delaney

Agenda

Overview of Event Notifications

- Why use them?
- Compared with SQL Trace and DDL Triggers
- Notification Events

Basics of Service Broker Architecture

- Components
- Understanding Activation
- Security

Questions

Overview of Event Notifications

Asynchronous execution outside of transaction scope

Respond to Trace and DDL Events

Perform additional processing immediately inside the Database Engine

Alternative to SQL Trace and DDL Triggers

Compared to SQL Trace

Pros

- Doesn't require trace files or tables for intermediate data
- Persist across server restarts
- Delayed consumption of Events

Cons

- Additional overhead for XML generation
- Requires knowledge of Xquery to parse XML
- DDL Events not generated if rollback occurs
- Only a subset of Trace Events available

Compared to Triggers

Pros

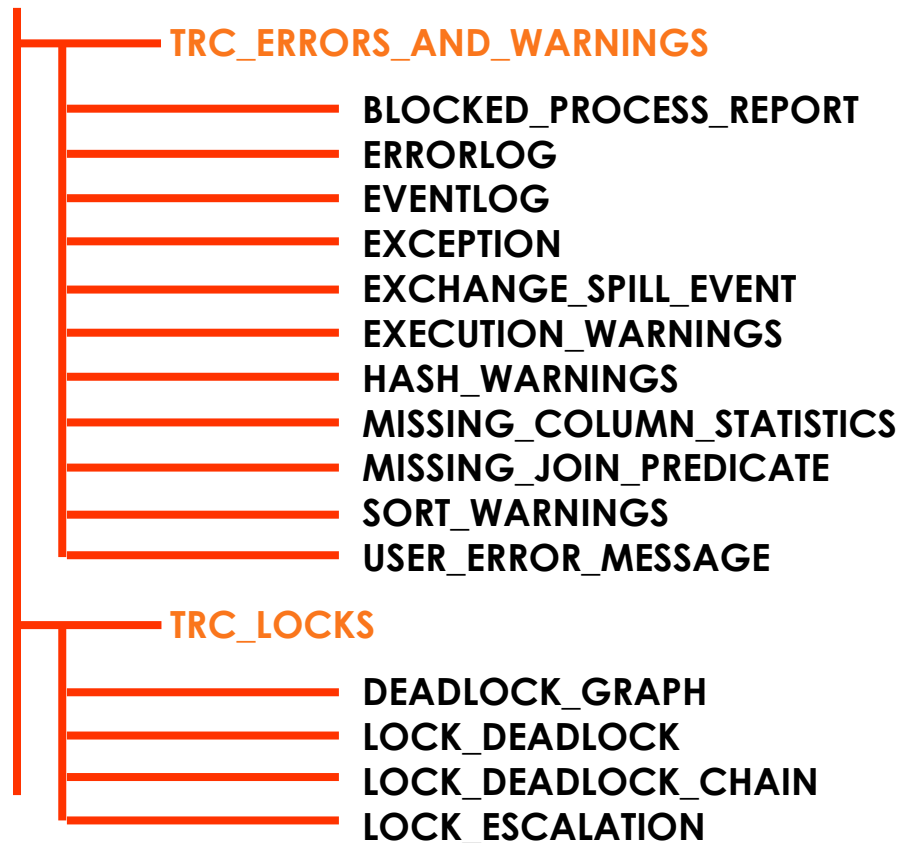
- Processing is decoupled from the event
- Can be processed on remote server (triggers are local only)
- Event Notifications can't be rolled back (Auditing)

Cons

- No DML Event Notifications only DDL Events
- Notifications only generate XML data, triggers can run code (Activation runs code)

Notification Events –Trace

Event Notification Trace Event Groups



Notification Events – DDL

DDL_DATABASE_LEVEL_EVENTS

DDL_TABLE_VIEW_EVENTS

DDL_TABLE_EVENTS

(CREATE_TABLE,ALTER_TABLE,DROP_TABLE)

DDL_VIEW_EVENTS

(CREATE_VIEW,ALTER_VIEW,DROP_VIEW)

DDL_INDEX_EVENTS

(CREATE_INDEX,ALTER_INDEX,DROP_INDEX)

DDL_STATISTICS_EVENTS

(CREATE_STATISTICS,ALTER_STATISTICS,DROP_STATISTICS)

Service Broker Overview

Queue based, durable messaging framework

Flexible conversation based architecture

Bidirectional messaging guarantees reliable delivery

Transactional processing ensures data integrity

Leverage existing technologies for processing (TSQL stored procedures, .NET, WCF, etc.)

Event Notifications Basic Service Broker Components

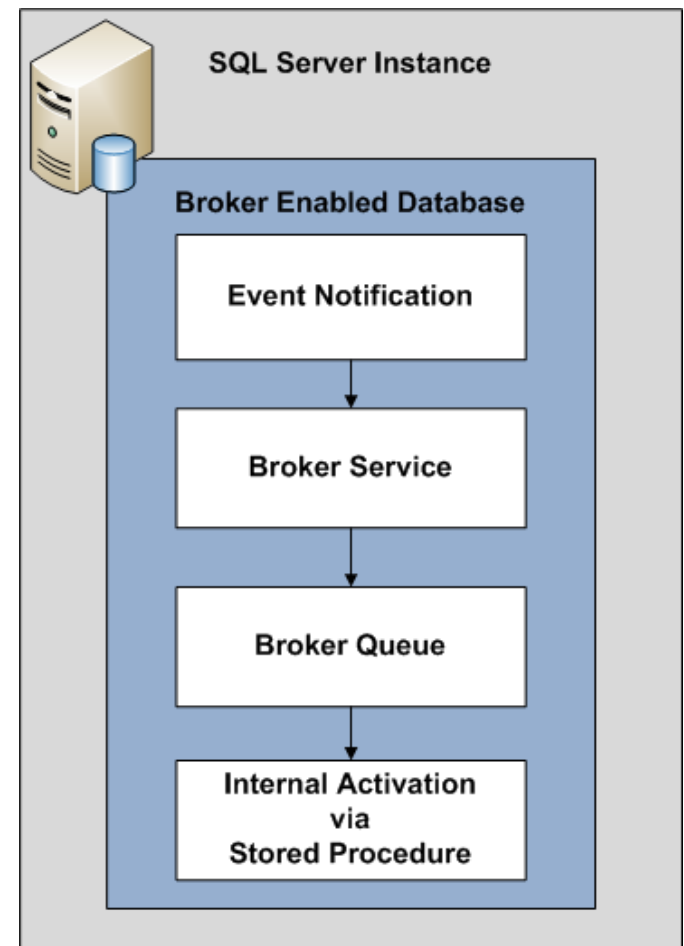
Event Notification

Broker Service

- Routes

Broker Queue

Optional - Activation Stored



Creating Event Notifications

Create a QUEUE

Create a SERVICE on a QUEUE

Create a ROUTE for the SERVICE

Create an EVENT NOTIFICATION to a SERVICE

Create a APPLICATION to process notification events from the QUEUE

***Note:** No Service Broker Endpoint or networking is required for Event Notifications

Creating an Event Notification

```
CREATE EVENT NOTIFICATION name  
ON { SERVER | DATABASE | QUEUE }  
[ WITH FAN_IN ]  
FOR { event_type | event_group } [ ,...n ]  
TO SERVICE broker_service  
{ 'broker_instance_specifier' | 'current  
database' }
```



Demo Event Notifications Basics

October 11-14, Seattle, WA



Event Notifications | DMV's

sys.event_notification_event_types contains a list of all events available with Event Notifications

sys.server_event_notifications contains a row for each server-level event notification object

sys.server_events contain a row for each event available for use with Event Notifications

sys.service_queues contains a row for each service broker queue in the current database

sys.services contains a row for each service

sys.routes contains a row for each service broker route



Demo Event Notifications DMV's

October 11-14, Seattle, WA



Consuming Event Notification Data

Manually querying Queue vs using RECEIVE
Internal (Event Based) Activation

- Activation Stored Procedure

External Activation through
QUEUE_ACTIVATION event notification



Demo Consuming Event data

Remote Event Notifications

Requires Service Broker TCP Endpoints
Full Dialog or Service Account Security

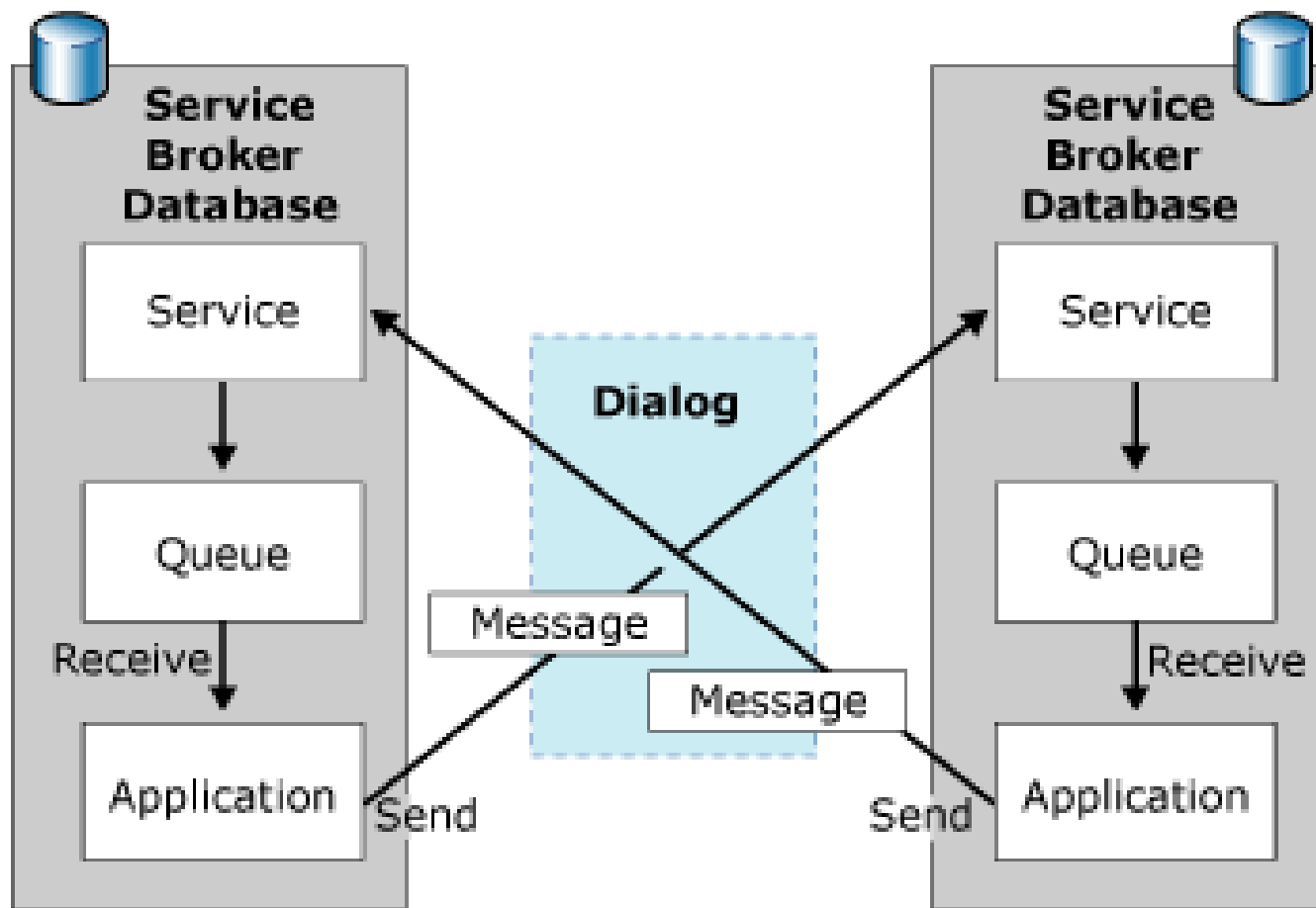
```
CREATE ENDPOINT SSBENDPOINT  
STATE = STARTED,  
AS TCP (LISTENER PORT = 50196)  
FOR SERVICE_BROKER  
(AUTHENTICATION = WINDOWS,  
ENCRYPTION = SUPPORTED)
```

Remote Event Notifications (Cont'd)

Requires Outbound and Return Routes for conversations to succeed.

- Remote Service Name to send to
- Remote Broker Instance for the service
- Remote Address to the service broker endpoint
- CREATE ROUTE ExpenseRoute
- WITH
- SERVICE_NAME = '//Adventure-Works.com/Expenses',
- BROKER_INSTANCE = 'D8D4D268-00A3-4C62-8F91-634B89C1E315',
- ADDRESS = 'TCP://192.168.10.2:1234'

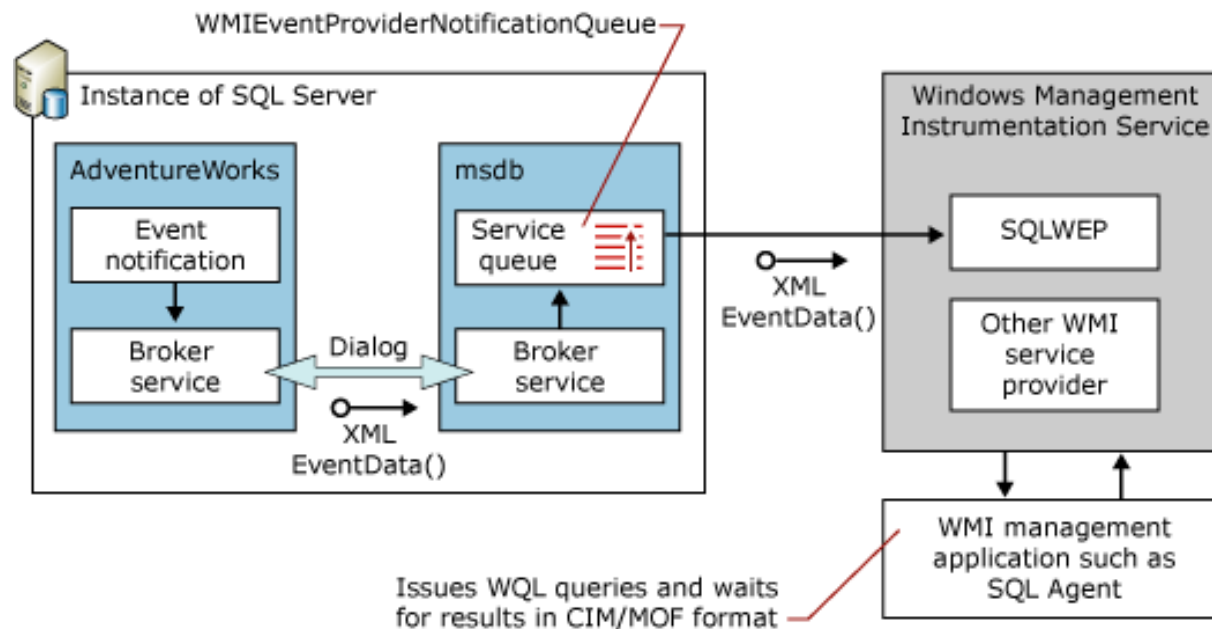
Remote Event Notification Architecture



WMI Events for SQL Server Architecture

Based on Event Notifications and Service Broker

Utilize a specialized Queue in msdb





Questions?

Complete the Evaluation Form to Win!

Win a Dell Mini Netbook – every day – just for handing in your completed form. Each session evaluation form represents a chance to win.

Pick up your evaluation form:

- In each presentation room
- Online on the PASS Summit website

Drop off your completed form:

- Near the exit of each presentation room
- At the Registration desk
- Online on the PASS Summit website

Sponsored by Dell





Thank you

for attending this session and the
2011 PASS Summit in Seattle