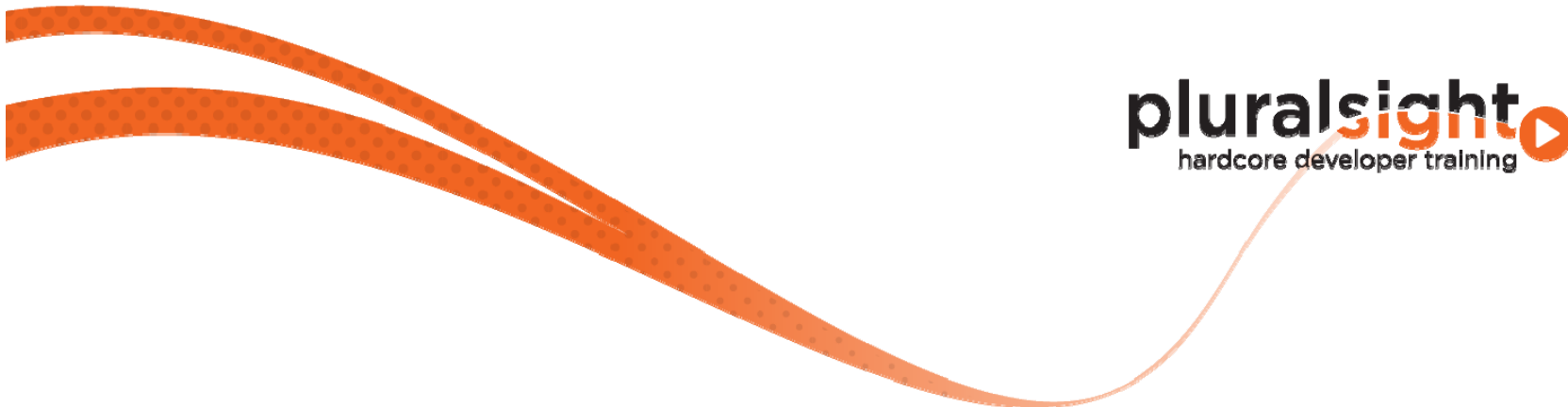


SQL Server: Advanced Extended Events

Module 2: Getting More From Event Sessions

Jonathan M. Kehayias
Jonathan@SQLskills.com



Introduction

- Event session design can maximize effectiveness and simplify problem identification
- Basic predicate definitions can limit the effectiveness of an event session and result in excessive event collection
- Improper use of actions can result in performance impacts and incorrect understanding of event data
- See Module 3 of the Introduction course for Extended Event object definitions

- In this module we'll cover:
 - Using textual comparators
 - Controlling event execution count
 - The sql_text action and its meaning
 - Determining execution paths
 - Leveraging side-impacting actions

Using Textual Comparators

- **Basic predicate definitions were introduced in Module 6 of the Introduction course**
 - When an event session is started, all of the basic predicates are converted to textual comparators by the Engine as shown in the event_predicate column output of sys.dm_xe_session_events
- **The 3000-character limitation on predicates may make arithmetic operators impractical for defining predicates on busy systems or for complex event sessions**
- **Textual comparators are available in the sqlserver and package0 packages and the appropriate comparator must be used for the data type of the data being predicated**
- **Textual comparators allow predicates to be built that can be shorter in length and that maintain state information about the last evaluation of the predicate**
 - Tracking long executions with package0.greater_than_equal_int64
 - Sampling a subset of session_ids using package0.divides_by_uint64

Controlling Event Execution Count

- In some cases it may only be necessary to collect a single instance or limited number of instances of an event occurring for diagnostics and troubleshooting
- The `package0.counter` predicate can be used to limit the number of times an event is fired by an active session
 - This can be useful with side-effecting actions to limit their impact
- For example, sampling of data can be accomplished using the `package0.divides_evenly_by_uint64` textual comparator with the `package0.counter` predicate source to collect every fifth execution of the event while also limiting the total execution count

The sql_text Action

- This is the InputBuffer for the executing session and not the actual statement text of the executing statement
- InputBuffer masking of sensitive data using keyword match list for words like “password” blocks data collection
- Use TRACK_CAUSALITY and sql_statement_starting/completed event collect_statement customizable column in SQL Server 2012 instead of the sql_text action for event correlation
- Large actions like sql_text may require additional consideration for event sizing during session configuration to prevent event loss
 - This is covered further in Module 3 of this course

Determining Execution Paths with `tsql_stack`

- **Identifying usage of shared objects in a database can be challenging**
 - SQL dependencies can be invalidated through DDL changes
 - Source control may allow for pattern searching but can still miss objects when consistency has not been applied to usage
- **Troubleshooting problems in commonly shared code can be simplified by collecting the `tsql_stack` action which contains the Transact-SQL call frames from the start of execution**
 - Frames contain the `sql_handle` and offset information for each level of the call stack to the current point of execution

Side-Impacting Actions

- **Allow side effects to occur when an event fires**
 - Performing a memory dump of a single thread
 - Performing a memory dump of all threads
 - Inserting a debug break – (Beware: this really does what it says!)
- **Provide a much more robust method of collecting memory dump information over DBCC DUMPTRIGGER**
 - Limited to a specific session_id or only performing the dump collection once with the package0.counter predicate
- **Keep in mind that these still execute synchronously and take extreme caution when using one of these in a production environment**
 - Generally these should only be used under the guidance of Product Support Services for troubleshooting problems with SQL Server

Summary

- Leveraging the predicates and actions correctly in event sessions can reduce the impact the session has on the instance as well as improve the diagnostic potential of the data being collected
- The next module will look at:
 - Event session options