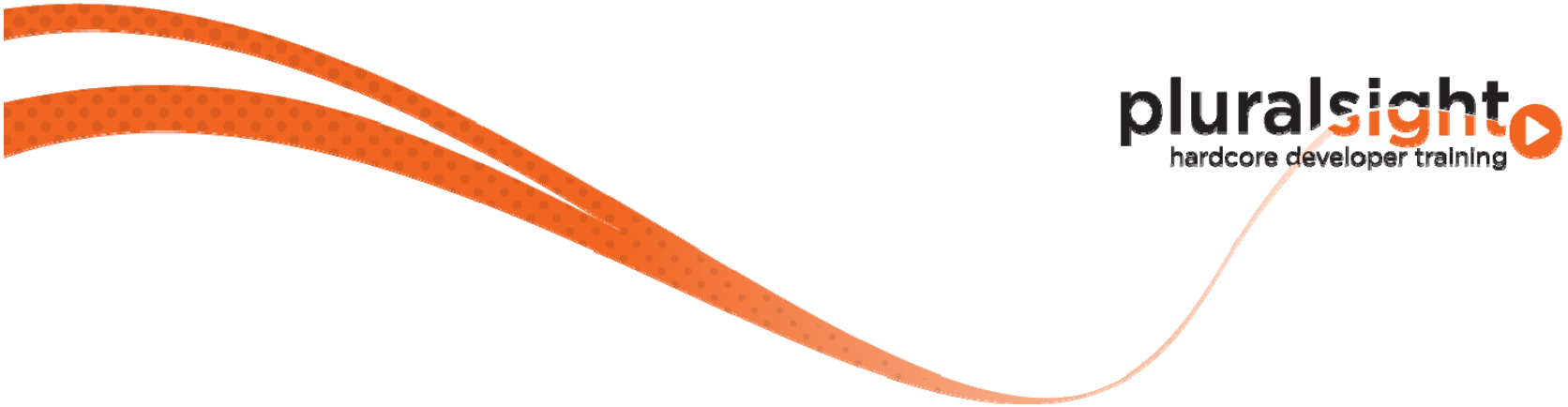
SQL Server: Advanced Extended Events

Module 3: Event Session Options

Jonathan M. Kehayias
Jonathan@SQLskills.com



Introduction

- **Event session options control the behavior of the event session once it is started inside the engine**
- **Configuring session options can affect the performance of the event session, how events are processed, and whether or not events will be lost under excessive event generation**
- **Default option values are generally acceptable for most event sessions**

- **In this module we'll cover:**
  - Session options
  - Default values
  - Reasons to change option values

Memory Buffer Configuration

- **MAX_MEMORY**
  - Configures the relative value for the maximum buffer size for the event session in either KB or MB
  - MAX_MEMORY is not the actual max memory for the event session since buffers are created in 64KB increments and therefore align on 64KB boundaries
  - When using the streaming target for Extended Events in SQL Server 2012, increasing the value for MAX_MEMORY can prevent the session from disconnecting under excessive event generation
    - Additional buffer space allows for more events to be buffered without the session disconnecting
  - The MAX_MEMORY session option directly correlates to the MEMORY_PARTITION_MODE option as shown later in this course
    - When logging to a file target, these two options can control the size of the file writes for event sessions that generate large amounts of data
  - Default value is 4MB for the event session

Memory Buffer Configuration (2)

- **MEMORY_PARTITION_MODE**
  - Determines the number of memory buffers created for an event session
    - NONE: 3 buffers
    - PER_NODE: 3 buffers per NUMA node on the server
    - PER_CPU: 2.5 buffers per SQL Server scheduler
  - For NUMA-based systems, partitioning per NUMA node results in node locality improvements for the event session that can improve performance slightly
  - When event generation is expected to be significant, using PER_CPU partitioning may allow the event session to retain events that would be dropped under other partitioning modes
  - If the MAX_MEMORY value is configured to a small value, the event session may not start due to size of the memory buffers being to small
  - Default value is NONE

Memory Buffer Configuration (3)

- **MAX_EVENT_SIZE**
  - Establishes the size of the large buffers that are associated with an event session for collecting events that are larger than the size of a standard buffer
    - When MAX_EVENT_SIZE is specified two buffers are created
    - The minimum value for MAX_EVENT_SIZE when specified is the value configured for MAX_MEMORY option for the event session
  - Large buffer memory allocations are in addition to the standard buffer size for an event session
    - May be used for buffering events if the standard buffers are full as a side effect of the large buffers existing
  - May be necessary for event sessions that capture large event definitions such as those that are collecting IOData from file I/O operations or capturing large batch text using the sql_text action
  - Default value is 0KB or no large buffers created

Event Dispatch and Retention

- **EVENT_RETENTION_MODE**
  - Determines whether single events, entire buffers, or no events can be lost by the event session
  - NO_EVENT_LOSS can impact performance under heavy event generation and can limit the events that can be added to a event session
  - Default value is ALLOW_SINGLE_EVENT_LOSS
- **MAX_DISPATCH_LATENCY**
  - Configures the maximum time an event will remain in a partially-full memory buffer before being dispatched to the asynchronous target(s) for consumption
  - May impact the dispatcher pool if multiple event sessions exist with small dispatch latency configurations
  - Default value is 30 seconds

Session Startup and Event Correlation

- **STARTUP_STATE**
  - Configures the event session to start automatically when SQL Server starts
  - Default is OFF
- **TRACK_CAUSALITY**
  - Adds the attach_activity_id and attach_activity_id_xfer action to every event in the session to allow for correlation of which events are related to each other and the order they fired
  - The attach_activity_id and attach_activity_id_xfer actions are comprised of two parts:
    - A guid: tracks events that relate to each other
    - A sequence number (seq): provides the order the events fired within a sequence.
  - An attach_activity_id_xfer is attached to events that transfer activities for end-to-end tracking of events for correlation
  - Default is OFF

Summary

- **Understanding the effect of session options is important for maximizing the performance of an event session and minimizing the risk of event loss for busy systems**
- **Incorrectly configuring session options can prevent an event session from being created or starting**

- **The next module will look at:**
  - Advanced Targets