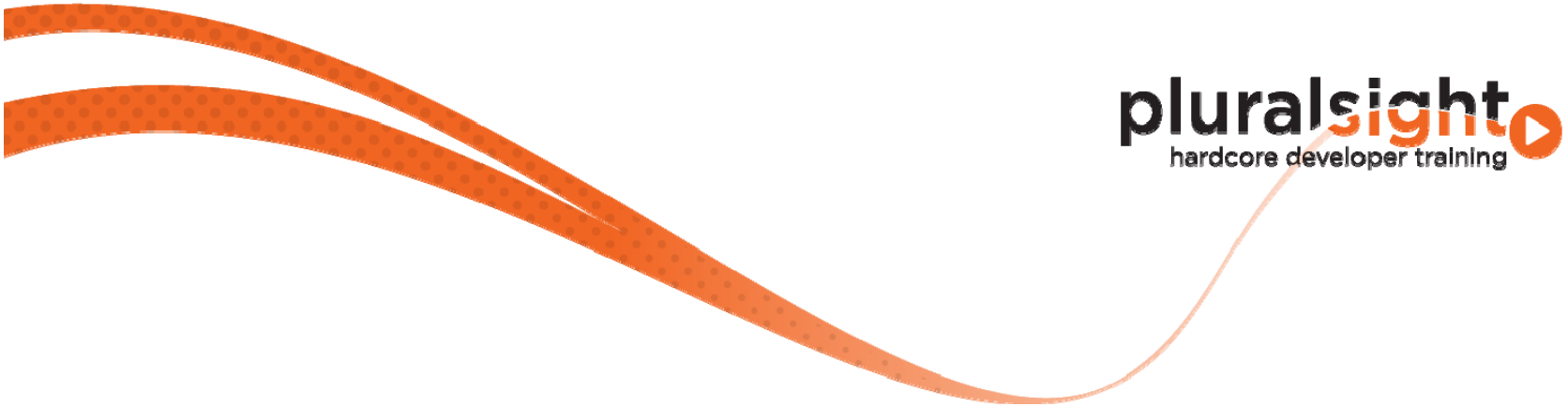
SQL Server: Advanced Extended Events

Module 4: Advanced Targets

Jonathan M. Kehayias
Jonathan@SQLskills.com



pluralsight
hardcore developer training

Introduction

- **Basic targets like the ring_buffer and event_file provide general event collection mechanisms for Extended Events, but require post collection processing to find important facts in the data**
- **Extended Events includes additional specialized targets that can be used to simplify diagnosing problems**

- **In this module we'll cover:**
  - The histogram target
  - The pair_matching target
  - The event_counter target
  - Integration with Event Tracing for Windows (ETW)

histogram Target

- **Prior to SQL Server 2012, this was implemented as two separate targets: synchronous_bucketizer and asynchronous_bucketizer**
- **Aggregates events into buckets creating a histogram (hence the name) of the frequency that events fire**
- **Only provides a count of occurrences, it does not allow for additional aggregation using operations like SUM, AVG, MIN, or MAX**
  - This level of analysis requires collecting the data to either the ring_buffer or event_file and post processing of the events
  - The UI can simplify this type of analysis as shown in Module 5 of this course

histogram Target Options

- **slots: the number of buckets or groupings to maintain**
  - The default value for this option is 256
  - Events that do not fit within an existing bucket value are ignored when the available slots are full for an event session
  - The value specified for this option is rounded up to the next power of 2
- **filtering_event_name: any event that is available in the session**
  - This option is optional, but specifies the specific event to bucket on
  - If specified the value must be in the format of package_name.event_name
- **source_type: specifies the type of object that the bucket is based on**
  - This option has two values:
    - 0 = an event in the session
    - 1 = an action in the session (the default)
- **source: the event column or action, in the format of package.action_name, that the buckets are based on**

pair_matching Target

- **Specialized target that matches events based on a specified criteria and discards the matched pairs, leaving unmatched events only in the data**
- **Careful selection of the pairing criteria has to be made or invalid event pairings will occur resulting in incorrect analysis**
  - E.g. lock_acquired and lock_released are not fired the same number of times when lock escalation occurs
- **Provides the same output XML as the ring_buffer for event data**

pair_matching Target Options

- **begin_event: any event that is present in the current session**
  - This option specifies the event name for the beginning event to be paired
- **end_event: any event that is present in the current session**
  - This option specifies the event name for the ending event to be paired
- **begin_matching_columns: comma-delimited and ordered list of column names for matching events on**
- **end_matching_columns: comma-delimited and ordered list of column names for matching events on**
- **begin_matching_actions: comma-delimited and ordered list of action names for matching events on**
- **end_matching_actions: comma-delimited and ordered list of action names for matching events on**

pair_matching Target Options (2)

- **respond_to_memory_pressure: specifies the target response to memory pressure**
  - Can be set to two values:
    - 0 = Do not respond to memory pressure (the default)
    - 1 = Respond to memory pressure by not adding new orphans to the list
- **max_orphans: specifies the maximum number of unpaired events that will be collected in the target**
  - Once the limit is reached, unpaired events are removed on a first-in, first-out (FIFO) basis
  - The default for this option is 10,000

Troubleshooting Invalid Pairs

- **Troubleshooting incorrect matching should be done using TRACK_CAUSALITY and adding a ring_buffer or event_file target to the event session to collect all of the events**
- **When an unmatched event occurs in the pair_matching target, the attach_activity_id guid can be used to lookup all of the events in the ring_buffer or event_file target data to determine whether the event was unmatched correctly or not**
- **An example of potential unmatched event collection in error is the statement_starting and statement_completed events when a recompile occurs**
  - In this scenario the statement_starting event actually fires twice, once for the recompile and once for the actual execution
  - The state column of the event specifies whether it was for a recompile or not and can be used to filter out the recompilation events from being a part of the session and causing unmatched events for a session

event_counter Target

- Counts how many events occurred for the event session
- Use this when defining an event session for an unknown workload to determine how many events you can expect based on your event session's definition, as a part of planning the appropriate targets to use for the session
- There are no configurable options for this target

etw_classic_sync_target Target

- **Integrates diagnostics data generated by SQL Server with Event Tracing for Windows (ETW)**
  - End-to-end data collection including kernel information possible
- **Leverages the classic ETW provider and not the newer Windows 2008+ manifest-based ETW provider in Windows**
  - Backwards compatibility with older versions of Windows
  - Classic provider data format does not integrate well with current ETW tools, which are based on the manifest provider format
- **Useful for developers and to understand how things work, but impractical for general administration work**

etw_classic_sync_target Target Options

- **default_xe_session_name: any string up to 256 characters**
  - This value is optional and has a default of XE_DEFAULT_ETW_SESSION
- **default_etw_session_logfile_path - any string up to 256 characters**
  - This value is optional and has a default of %TEMP%\ XEEtw.etl
- **default_etw_session_logfile_size_mb: Any unsigned integer in MB**
  - This value is optional and has a default of 20MB
- **default_etw_session_buffer_size_kb: Any unsigned integer in KB**
  - This value is optional and has a default of 128KB
- **retries: Any unsigned integer**
  - The number of times to retry publishing the event to the ETW subsystem before dropping the event
  - The default is 0

etw_classic_sync_target Target Limitations

- **SQL Server Service startup account must be a member of the Performance Log Users group**
- **The ETW target does not support asynchronous event publishing**
- **Creates a single ETW session in windows, the default is named XE_DEFAULT_ETW_SESSION, and only one exists for the entire server, making segregation of data from multiple instances impossible**
- **The ETW session must be stopped by the command line even if the event session in SQL is dropped**
  - logman stop XE_DEFAULT_ETW_SESSION -ets

Summary

- **The advanced targets provide aggregation of event data and additional diagnostic capabilities for events that were previously not possible**
- **It may be necessary to use traditional data collection with the ring_buffer or event_file targets to perform more in-depth analysis of the event data being collected by an event session**
- **Integration with ETW, while limited, provides the ability to do end-to-end tracing with Windows kernel level events to diagnose the root cause of performance problems in SQL Server**

- **The next module will look at:**
  - Advanced UI features