SQL Server: Introduction to Extended Events

Module 8: Basic Targets

Jonathan M. Kehayias

Jonathan@SQLskills.com

pluralsight
hardcore developer training

Introduction

- **Targets are the data consumers for Extended Events, and two targets provide functionality similar to what was previously available in SQL Trace:**
  - The ring_buffer target provides an in-memory storage location for events being collected
  - The event_file target provides a file system storage location for events being collected

- **In this module we'll cover:**
  - ring_buffer target
  - ring_buffer target options
  - Event data XML
  - event_file target
  - event_file target options
  - Reading event_file data with Transact-SQL

ring_buffer Target

- **Memory-resident target that holds raw event data while a session is active**
- **Functions as a first-in-first-out (FIFO) queue for the data based on the target configuration options**
  - Strict FIFO – oldest event is removed when the memory buffers are full and a new event arrives (default)
  - Per Event FIFO – specifies the maximum number of occurrences to retain for each individual event in the session
- **DMV limitations in SQL Server 2008 may result in unreadable XML from sys.dm_xe_session_targets**
- **Ideal target for small data sets where event loss is acceptable**

ring_buffer Target Options

- **max_memory – the maximum amount of memory in KB to be used; older events are dropped when this is reached**
  - Default is 4MB of memory for the target
  - Specifies the size of binary data that is maintained, not the size of the XML document that is returned by the sys.dm_xe_session_targets DMV
- **occurrence – sets the preferred number of events of each type to keep**
  - Default is 0 which enforces strict FIFO
  - This option can be affected by the size of events that are being generated and may not result in the expected behavior under all workloads
- **max_event_limit – the maximum number of events to be kept in the target**
  - Default is 1000
  - Available in SQL Server 2012 and onwards

ring_buffer XML Output

- **The ring_buffer XML output is not schema bound but follows a generally predictable output**
- **The XML contains a single parent <RingBufferTarget> XML node that contains the attributes about the target operation since the event session was started**
  - eventsPerSec – the number of events processed per second
  - processingTime – the amount of time spent processing memory buffers
  - totalEventsProcessed – the total number of events processed
  - eventCount – the number of events in the ring_buffer
  - droppedCount – the number of memory buffers that were dropped
  - memoryUsed – the amount of memory currently used by the ring_buffer
- **Inside the <RingBufferTarget> parent node are the <event> nodes which contain the information returned by the events defined in the event session**

Event XML

- The XML for individual events does not conform to any schema, though the XML has a predictable format similar to:

```
<event name="" package="" timestamp="">
  <data name="">
    <type name="" package="" />
    <value> </value>
    <text />
    </data>
  <action name="" package="">
    <type name="" package="" />
    <value> </value>
    <text />
    </action>
</event>
```

Event XML (2)

- **The root <event> node contains the event name and package name associated with the event, and the timestamp for the date and time in GMT that the event fired on the server**
  - SQL Server 2008 will also contain the id and version for the event
- **The <event> node will have a separate <data> node for each of the columns that the event outputs**
  - The <data> nodes will be listed in column_id order in sys.dm_xe_object_columns for the data columns
- **The <event> node will have a separate <action> node for each of the actions that were added to the event**
  - The <action> nodes will be listed in the order the actions were defined for the event

Event XML (3)

- **The <data> nodes and <action> nodes share a similar XML format**
- **The <data> nodes contain a single attribute containing the name of the column**
- **The <action> nodes contain two attributes, one for the action name, and the other the package for the action**
- **Each <data> or <action> node will have a <type> node with the name and package of the data type of the value being returned**
- **The <data> or <action> nodes can also have two additional nodes:**
  - A <value> node which contains the value for the data being returned
  - A <text> node which will contain the lookup text for columns that correspond to maps in Extended Events

event_file Target

- **Similar to the trace file in SQL Trace, this target collects event data into a proprietary, binary file format in the file**
- **Supports maximum size and rollover file configurations for the event session**
- **Event data has the same XML format as individual events in the ring_buffer target**

- **Changes in SQL Server 2012:**
  - This target was the asynchronous_file_target in SQL Server 2008/2008R2
  - Event session DDL was updated to allow legacy event session DDL from 2008 to remap the target to the event_file internal in the engine
  - SQL Server 2008/2008R2 required the use of separate log and metadata files that were collapsed into a single file in SQL Server 2012
  - Introduction of a .NET API for reading the file data outside of the SQL Server Engine

event_file Target Options

- **filename – specifies the location and file name of the log file and is the only mandatory option for the target**
  - The filename specified is appended with _0_ and a long integer for the number of milliseconds since January 1, 1600 when the file is created
- **max_file_size – configures the maximum file size in MB**
  - Default value is 1GB
- **max_rollover_files – configures the maximum number of files to retain during file rollover**
- **increment – configures the size in MB to grow the log file each time the file runs out of space**
  - Default value is twice the size of the memory buffers for the session
- **metadatafile – specifies the location of the metadata file that corresponds to the log file for the session in SQL Server 2008/2008R2**
  - Is not used in SQL Server 2012

Reading event_file Data with Transact-SQL

- **The sys.fn_xe_file_target_read_file table-valued function can be used to read event_file data in Transact-SQL**
- **Parameters:**
  - path – the path to the files to read, which can contain wildcards and include the name of a file (no default)
  - mdpath – the path to the metadata file that corresponds to the file or files specified by the path argument (no default)
    - Maintained for backward compatibility in SQL Server 2012, for log files generated in previous versions of SQL Server
  - initial_file_name – the first file to read from path(no default)
    - If null is specified as the argument all the files found in path are read
  - initial_offset – the offset in the first file where reading begins
    - If null is specified as the argument the entire file will be read
  - initial_file_name and initial_offset are paired arguments; if a value is specified for either argument you must specify a value for the other argument

Summary

- **Two targets provide basic event data collection capabilities in Extended Events: the ring_buffer and event_file targets**
- **The ring_buffer target is best for small data sets that fit in memory, or where a fixed number of events or a fixed number of each event needs to be maintained**
- **The event_file target is best for large data sets or data sets that require all events to be maintained for later analysis**

- **The next module will look at:**
  - Basic troubleshooting scenarios