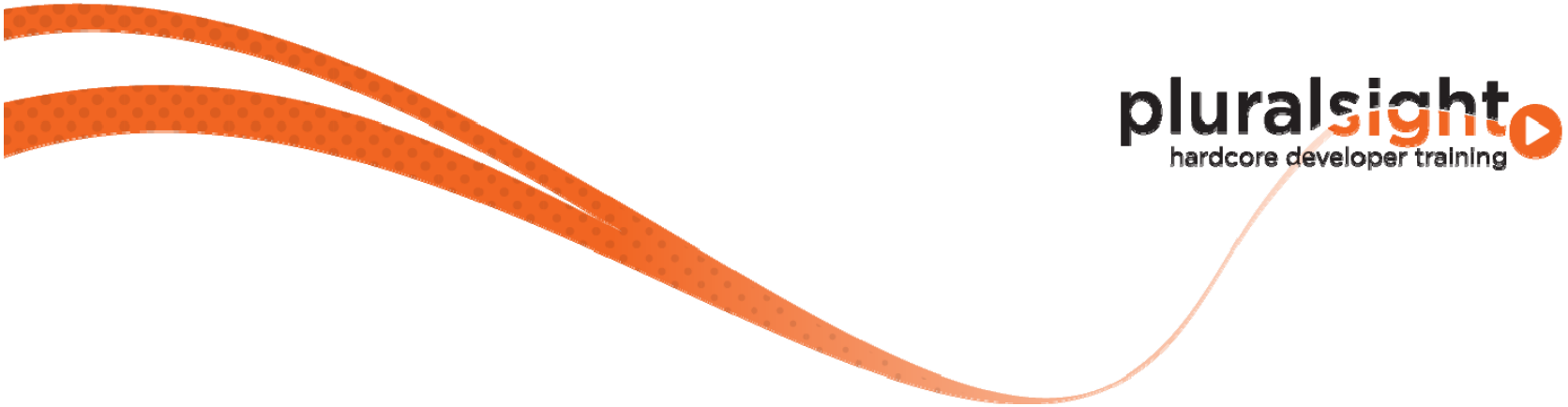
SQL Server: Advanced Extended Events

Module 6: Extended Events .NET API and PowerShell Integration

Jonathan M. Kehayias
Jonathan@SQLskills.com



Introduction

- **Extended Events can be managed outside of Transact-SQL for programmatic access to the core functionality**
- **The .NET API can be used with any of the .NET languages to manage event sessions and read event data**
- **The PowerShell provider can be used to manage event sessions from the command line**

- **In this module we'll cover:**
  - Considerations for leveraging the .NET API
  - Extended Events object model
  - Using the Extended Events reader
  - Using the PowerShell provider

Considerations for Leveraging the .NET API

- **Necessary assemblies are not a part of SQL Management Objects (SMO) or included in the SQL Server Feature Pack downloads**
  - Creates licensing challenges for redeploying applications that depend on the assemblies required
- **Prior to SQL Server 2012 Service Pack 1 only the x86 versions of the assemblies were shipped**
- **Requires referencing the following assemblies**
  - Microsoft.SqlServer.Management.XEvent
  - Microsoft.SqlServer.Management.XEventEnum
  - Microsoft.SqlServer.Management.Sdk.Sfc

Extended Events Object Model

- **The top level object in the model is XEStore**
- **The XEStore object provides access to two sources:**
  - Metadata – includes the packages and associated metadata for the Extended Events objects available in the instance
  - Runtime – the event sessions that are defined on the instance
- **The intention behind the object model is to support the programmatic creation of sessions from SSMS only**
  - Creating sessions from Transact-SQL DDL commands is often faster and requires less code than creating them programmatically inside of .NET

Using the Extended Events Reader

- **Provides a mechanism for programmatically consuming event data from files generated by the event_file target or from live event sessions by attaching the streaming provider to the session**
- **The streaming provider implements shadow copies of the event session buffers for consumption by the reader when open**
  - The streaming provider changes the MAX_DISPATCH_LATENCY to 3 seconds for the event session once connected to provide a "live" view of the data, but the stream is still asynchronously served by the session
    - The MAX_DISPATCH_LATENCY will automatically revert back to the session-configured value when the stream is closed
    - If multiple streams are opened, the first one to close will revert the MAX_DISPATCH_LATENCY value delaying processing for other streams
  - Failure to consume the events efficiently will result in the streaming provider being closed by the session to prevent performance impact to SQL Server

Using the Extended Events Reader (2)

- **Eliminates the need to manually process XML data from files or the live event stream providing object properties for the data and actions generated by events firing in the Engine**
- **Requires referencing the Microsoft.SqlServer.XEvent.Linq.dll**
  - Prior to SQL Server 2012 Service Pack 1 only the x86 version exists:
    - C:\Program Files (x86)\Microsoft SQL Server\110\Tools\Binn\ManagementStudio\Extensions\Application
  - After SQL Server 2012 Service Pack 1 separate versions exist for X86 and x64 usage in the following paths:
    - X86 - C:\Program Files (x86)\Microsoft SQL Server\110\Tools\Binn\ManagementStudio\Extensions\Application
    - X64 - C:\Program Files\Microsoft SQL Server\110\Shared
  - Application must reference the Microsoft.SqlServer.XEvent.Linq namespace
- **The reader is exposed through the QueryableXEventData object in the object model**

Reading Files with QueryableXEventData

- **The QueryableXEventData object is overloaded to provide multiple methods of reading file data from Extended Events**
  - Single string to a single path
    - Can be an explicit file path to a single file
    - Can be a wild card string to process all files matching the wild-card expression
  - Single string array containing a list of multiple XEL files to process
  - Two string arrays containing a list of multiple XEL files to process along with a list of multiple XEM files to process for the XEL files
    - Provides backwards compatibility to SQL Server 2008 and 2008 R2 files
    - Can also contain wild-card paths to process all files matching the expression

Reading Live Data with QueryableXEventData

- **The QueryableXEventData object can also read event data from an active session asynchronously by connecting an event_stream target to the active session**
  - The event_stream target can only be connected through the use of the .NET API, it can't be configured for use through standard DDL statements
    - The stream can be read in binary through Transact-SQL but this is not recommended
- **The QueryableXEventData object constructor provides an overload specifically for attaching a stream to an event session which requires the following fields to be specified:**
  - connectionString – the connection string to connect to the server
  - source – the event session to attach the stream to
  - EventStreamSourceOptions – the source of the event stream
  - EventStreamCacheOptions – specifies the cache options of the event stream

Using the PowerShell Provider

- **Extended Events expose a PowerShell provider, implemented as a PSDrive, to manage event sessions inside SQL Server**
  - SQLSERVER: drive provides access through the XEvent directory
  - From Object Explorer in SQL Server Management Studio, Start PowerShell on the Extended Events folder
  - Full path for a instance - SQLSERVER:\XEvent\ServerName\InstanceName
- **Event sessions can be accessed from the Sessions folder under the instance**
- **New sessions can be created using the same objects provided by the .NET API for Extended Events**
  - Creating sessions from Transact-SQL DDL commands is often faster and requires less code than creating them programmatically inside of Powershell

Summary

- **SQL Server 2012 provides programmatic access to Extended Events for .NET applications and PowerShell**
- **Managing event sessions with Transact-SQL DDL commands is often faster and requires less code using .NET or PowerShell**
- **The extended events reader provides the ability to read "live" event data from an active event session or to process files generated by the event_file target**

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