Viewing Extended Events

Last updated by | Peter Hewitt | Jan 2, 2023 at 6:34 PM PST

Contents

Summary

This articles covers viewing Extended Events (XEvents) session data.

Introduction

XEvents is a lightweight performance monitoring system that enables users to collect data needed to monitor and troubleshoot problems in SQL Server. One of the functionalities is that it can track every query that goes into your Azure SQL database. SQL Server Management Studio (SSMS) provides a graphical user interface for XEvents to create and modify sessions, and display and analyze session data. For more details, refer to Extended Events overview.

XEvents replace the deprecated SQL Trace and SQL Server Profiler features.

Caution when using XEvents in SQL Database

For SQL Database, XEvents require the use of remote Azure storage since the end user doesn't have direct access to local storage. The volume of data generated is directly related to how busy the server is and the filters applied to the XEvents. If the application has a high volume of queries, it's best filter specifically only for the data required. A high volume (busy) database generates many events, and a large volume of unfiltered tracing could cause delays writing to remote storage and queries to slowdown.

While doing tracing, monitor the application behavior very closely. Keep tracing as short as possible and stop the XEvent as soon as you have what you need.

Collecting XEvents

The Collecting XEvents article details how to configure and create an XEvent session in SQL Database.

Viewing XEvents data

SQL Server Management Studio (SSMS) can be used to view the session data. There are several advanced features in the SSMS UI that can be used to view the data captured by an XEvent for analysis.

The Advanced Viewing of Target Data from Extended Events in SQL Server provides extensive information about the advanced features of SSMS to view the target data from XEvents in rich detail. It explains how to:

- Open and view the target data, in various ways.
- Export the target data to various formats, by using the special menu or toolbar for Extended Events.
- Manipulate the data while viewing, or before exporting.

This article is written from the perspective of SQL Server. Note the differences between SQL Server and Azure SQL Database.

Differences with Azure SQL Database

There is a high degree of parity in the implementation and capabilities of XEvents in the two products Microsoft SQL Server and Azure SQL Database. But there are some differences that affect the SSMS UI (user interface).

- For SQL Database, the package0.event_file target cannot be a file on the local disk drive. Instead, an Azure Storage container must be used. Therefore when you are connected to SQL Database, the SSMS UI asks for a storage container, instead of a local path and file name.
- In the SSMS UI when you see the check box **Watch live data** is grayed and disabled, it is because that feature is not available for SQL Database.
- A few Extended Events sessions are installed with SQL Server. Under the **Sessions** node we can see **AlwaysOn_health** plus a couple others. These are not visible when connected to SQL Database because they do not exist for SQL Database.

For documentation about XEvents that is specific to Azure SQL Database, see:

- Extended Events in Azure SQL Database

Public Doc Reference

Extended events in Azure SQL Database
Event File target code for extended events in Azure SQL Database and SQL Managed Instance
Ring Buffer target code for extended events in Azure SQL Database
Advanced Viewing of Target Data from Extended Events in SQL Server

How good have you found this content?