SQL Server Profiler

10/24/2016 • 9 minutes to read • Contributors

In this article

Where is the Profiler?
Capture and replay trace data
SQL Server Profiler
SQL Server Profiler concepts
SQL Server Profiler tasks
See also

THIS TOPIC APPLIES TO: SQL Server Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

SQL Server Profiler is an interface to create and manage traces and analyze and replay trace results. Events are saved in a trace file that can later be analyzed or used to replay a specific series of steps when trying to diagnose a problem.

> **IMPORTANT!!**
> We are announcing the deprecation of SQL Server Profiler for Database Engine Trace Capture and Trace Replay. These features **are** available in SQL Server 2016 but will be removed in a later version.
>
> The *Microsoft.SqlServer.Management.Trace* namespace that contains the Microsoft SQL Server Trace and Replay objects will also be deprecated.
> **Note** that SQL Server Profiler for the Analysis Services workloads is NOT being deprecated, and will continue to be supported.
>
> Submit your feedback and questions on our **Connect page.**

Where is the Profiler?

You can start the Profiler in a number of ways from within SSMS. Here is a topic that lists the ways to start the Profiler.

FEEDBACK

Capture and replay trace data

The following table shows the features we recommend using in SQL Server 2017 to capture and replay your trace data.

Feature\Target Workload	Relational Engine	Analysis Services
Trace Capture	Extended Events graphical user interface in SQL Server Management Studio	SQL Server Profiler
Trace Replay	Distributed Replay	SQL Server Profiler

SQL Server Profiler

Microsoft SQL Server Profiler is a graphical user interface to SQL Trace for monitoring an instance of the Database Engine or Analysis Services. You can capture and save data about each event to a file or table to analyze later. For example, you can monitor a production environment to see which stored procedures are affecting performance by executing too slowly. SQL Server Profiler is used for activities such as:

- Stepping through problem queries to find the cause of the problem.
- Finding and diagnosing slow-running queries.
- Capturing the series of Transact-SQL statements that lead to a problem. The saved trace can then be used to replicate the problem on a test server where the problem can be diagnosed.
- Monitoring the performance of SQL Server to tune workloads. For information about tuning the physical database design for database workloads, see Database Engine Tuning Advisor.
- Correlating performance counters to diagnose problems.

  SQL Server Profiler also supports auditing the actions performed on instances of SQL Server. Audits record security-related actions for later review by a security administrator.

SQL Server Profiler concepts

To use SQL Server Profiler, you need to understand the terms that describe the way the tool functions.

> **NOTE!** Understanding SQL Trace really helps when working with SQL Server Profiler. For more information, see SQL Trace.

Event
An event is an action generated within an instance of SQL Server Database Engine. Examples of these are:

- Login connections, failures, and disconnections.
- Transact-SQL SELECT, INSERT, UPDATE, and DELETE statements.
- Remote procedure call (RPC) batch status.
- The start or end of a stored procedure.
- The start or end of statements within stored procedures.
- The start or end of an SQL batch.
- An error written to the SQL Server error log.
- A lock acquired or released on a database object.
- An opened cursor.
- Security permission checks.

  All of the data generated by an event is displayed in the trace in a single row. This row is intersected by data columns that describe the event in detail.

  EventClass
  An event class is a type of event that can be traced. The event class contains all of the data that can be reported by an event. Examples of event classes are the following:

- **SQL:BatchCompleted**
- **Audit Login**
- **Audit Logout**
- **Lock:Acquired**
- **Lock:Released**

  EventCategory
  An event category defines the way events are grouped within SQL Server Profiler. For example, all lock events classes are grouped within the **Locks** event category. However, event categories only exist within SQL Server Profiler. This term does not reflect the way Engine events are grouped.

  DataColumn
  A data column is an attribute of an event classes captured in the trace. Because the event class determines the type of data that can be collected, not all data columns are applicable to all

event classes. For example, in a trace that captures the **Lock:Acquired** event class, the **BinaryData** data column contains the value of the locked page ID or row, but the **Integer Data** data column does not contain any value because it is not applicable to the event class being captured.

Template
A template defines the default configuration for a trace. Specifically, it includes the event classes you want to monitor with SQL Server Profiler. For example, you can create a template that specifies the events, data columns, and filters to use. A template is not executed, but rather is saved as a file with a .tdf extension. Once saved, the template controls the trace data that is captured when a trace based on the template is launched.

Trace
A trace captures data based on selected event classes, data columns, and filters. For example, you can create a trace to monitor exception errors. To do this, you select the **Exception** event class and the **Error**, **State**, and **Severity** data columns. Data from these three columns needs to be collected in order for the trace results to provide meaningful data. You can then run a trace, configured in such a manner, and collect data on any **Exception** events that occur in the server. Trace data can be saved, or used immediately for analysis. Traces can be replayed at a later date, although certain events, such as **Exception** events, are never replayed. You can also save the trace as a template to build similar traces in the future.

SQL Server provides two ways to trace an instance of SQL Server: you can trace with SQL Server Profiler, or you can trace using system stored procedures.

Filter
When you create a trace or template, you can define criteria to filter the data collected by the event. To keep traces from becoming too large, you can filter them so that only a subset of the event data is collected. For example, you can limit the Microsoft Windows user names in the trace to specific users, thereby reducing the output data.

If a filter is not set, all events of the selected event classes are returned in the trace output.

SQL Server Profiler tasks

Task description	Topic
Lists the predefined templates that SQL Server provides for monitoring certain types of events, and the permissions required to use to replay traces.	SQL Server Profiler Templates and Permissions
Describes how to run SQL Server Profiler.	Permissions Required to Run SQL Server Profiler
Describes how to create a trace.	Create a Trace (SQL Server Profiler)

Describes how to specify events and data columns for a trace file.	Specify Events and Data Columns for a Trace File (SQL Server Profiler)
Describes how to save trace results to a file.	Save Trace Results to a File (SQL Server Profiler)
Describes how to save trace results to a table.	Save Trace Results to a Table (SQL Server Profiler)
Describes how to filter events in a trace.	Filter Events in a Trace (SQL Server Profiler)
Describes how to view filter information.	View Filter Information (SQL Server Profiler)
Describes how to Modify a Filter.	Modify a Filter (SQL Server Profiler)
Describes how to Set a Maximum File Size for a Trace File (SQL Server Profiler).	Set a Maximum File Size for a Trace File (SQL Server Profiler)
Describes how to set a maximum table size for a trace table.	Set a Maximum Table Size for a Trace Table (SQL Server Profiler)
Describes how to start a trace.	Start a Trace
Describes how to start a trace automatically after connecting to a server.	Start a Trace Automatically after Connecting to a Server (SQL Server Profiler)
Describes how to filter events based on the event start time.	Filter Events Based on the Event Start Time (SQL Server Profiler)
Describes how to filter events based on the event end time.	Filter Events Based on the Event End Time (SQL Server Profiler)
Describes how to filter server process IDs (SPIDs) in a trace.	Filter Server Process IDs (SPIDs) in a Trace (SQL Server Profiler)
Describes how to pause a trace.	Pause a Trace (SQL Server Profiler)
Describes how to stop a trace.	Stop a Trace (SQL Server Profiler)
Describes how to run a trace after it has been paused or stopped.	Run a Trace After It Has Been Paused or Stopped (SQL Server Profiler)
Describes how to clear a trace window.	Clear a Trace Window (SQL Server Profiler)
Describes how to close a trace window.	Close a Trace Window (SQL Server Profiler)
Describes how to set trace definition defaults.	Set Trace Definition Defaults (SQL Server Profiler)

Describes how to set trace display defaults.	Set Trace Display Defaults (SQL Server Profiler)
Describes how to open a trace file.	Open a Trace File (SQL Server Profiler)
Describes how to open a trace table.	Open a Trace Table (SQL Server Profiler)
Describes how to replay a trace table.	Replay a Trace Table (SQL Server Profiler)
Describes how to replay a trace file.	Replay a Trace File (SQL Server Profiler)
Describes how to replay a single event at a time.	Replay a Single Event at a Time (SQL Server Profiler)
Describes how to replay to a breakpoint.	Replay to a Breakpoint (SQL Server Profiler)
Describes how to replay to a cursor.	Replay to a Cursor (SQL Server Profiler)
Describes how to replay a Transact-SQL script.	Replay a Transact-SQL Script (SQL Server Profiler)
Describes how to create a trace template.	Create a Trace Template (SQL Server Profiler)
Describes how to modify a trace template.	Modify a Trace Template (SQL Server Profiler)
Describes how to set global trace options.	Set Global Trace Options (SQL Server Profiler)
Describes how to find a value or data column while tracing.	Find a Value or Data Column While Tracing (SQL Server Profiler)
Describes how to derive a template from a running trace.	Derive a Template from a Running Trace (SQL Server Profiler)
Describes how to derive a template from a trace file or trace table.	Derive a Template from a Trace File or Trace Table (SQL Server Profiler)
Describes how to create a Transact-SQL script for running a trace.	Create a Transact-SQL Script for Running a Trace (SQL Server Profiler)
Describes how to export a trace template.	Export a Trace Template (SQL Server Profiler)
Describes how to import a trace template.	Import a Trace Template (SQL Server Profiler)
Describes how to extract a script from a trace.	Extract a Script from a Trace (SQL Server Profiler)
Describes how to correlate a trace with Windows performance log data.	Correlate a Trace with Windows Performance Log Data (SQL Server Profiler)
Describes how to organize columns displayed in a trace.	Organize Columns Displayed in a Trace (SQL Server

	Profiler)
Describes how to start SQL Server Profiler.	Start SQL Server Profiler
Describes how to save traces and trace templates.	Save Traces and Trace Templates
Describes how to modify trace templates.	Modify Trace Templates
Describes how to correlate a trace with Windows performance log data.	Correlate a Trace with Windows Performance Log Data
Describes how to view and analyze traces with SQL Server Profiler.	View and Analyze Traces with SQL Server Profiler
Describes how to analyze deadlocks with SQL Server Profiler.	Analyze Deadlocks with SQL Server Profiler
Describes how to analyze queries with SHOWPLAN results in SQL Server Profiler.	Analyze Queries with SHOWPLAN Results in SQL Server Profiler
Describes how to filter traces with SQL Server Profiler.	Filter Traces with SQL Server Profiler
Describes how to use the replay features of SQL Server Profiler.	Replay Traces
Lists the context-sensitive help topics for SQL Server Profiler.	SQL Server Profiler F1 Help
Lists the system stored procedures that are used by SQL Server Profiler to monitor performance and activity.	SQL Server Profiler Stored Procedures (Transact-SQL)

See also

Locks Event Category
Sessions Event Category
Stored Procedures Event Category
TSQL Event Category
Server Performance and Activity Monitoring

> **Note**
>
> The feedback system for this content will be changing soon. Old comments will not be carried over. If content within a comment thread is important to you, please save a copy. For more information on the upcoming change, **we invite you to read our blog post**.