

How to Run SQL Profiler Trace to Capture Information

06/21/2024 6:48 am EDT

Issue:

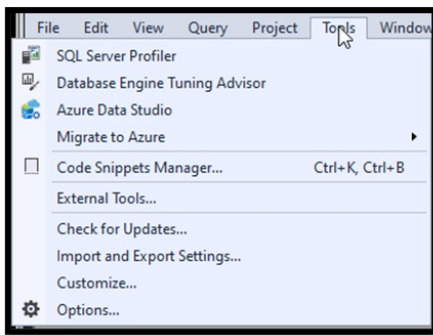
Customer is receiving errors in their SedonaOffice company. Is there a way to get mor information on why the errors are occurring?

Resolution:

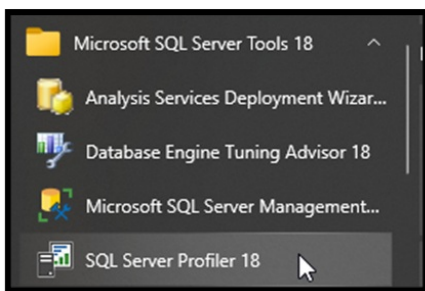
SQL server Profiler can be used to capture information from processes accessing the SQL server databases. Using SQL Server Profiler, you can create a trace that records and displays the commands sent to and the responses from the SQL server

Open SQL Server Management Studio

Connect to the correct server and select Tools then SQL Server Profiler.

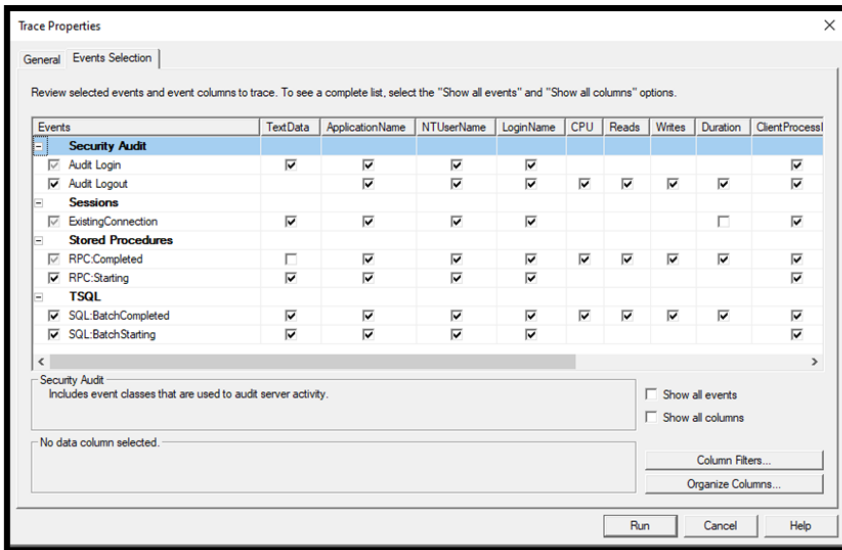


It can also be launched from the Windows Menu.



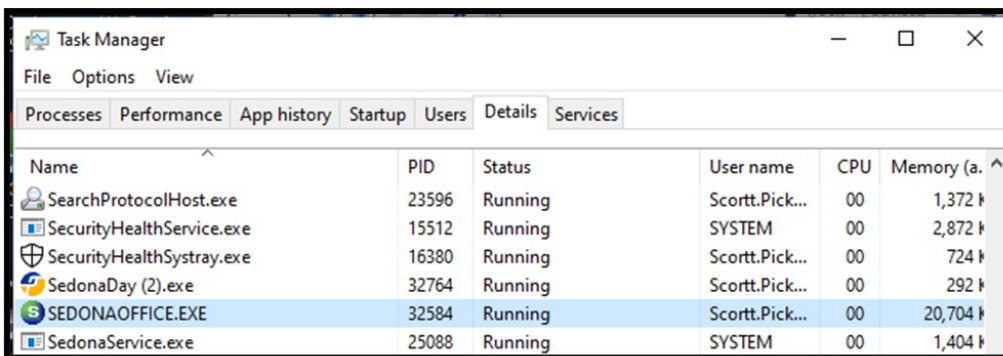
In the Events Selection Tab, you'll see the events it will monitor.

Be sure the options selected are the same as in the screenshot below.

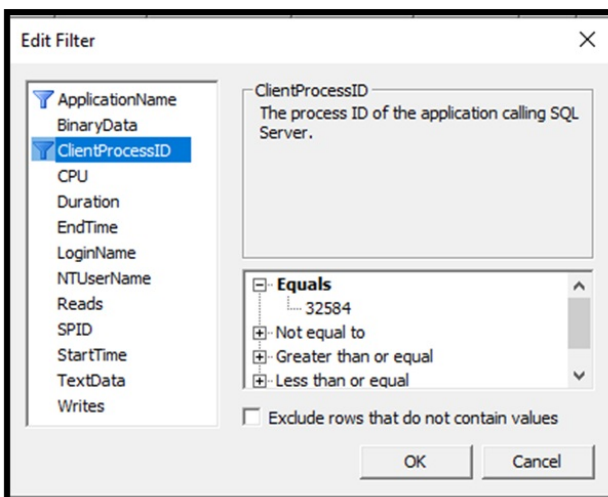


To limit the results to the machine having the issue we use the ProcessID from the application.

You can get this from the Windows Task Manager on the Details tab.



Enter the PID from the application in the Column Filter for ClientProcessID.



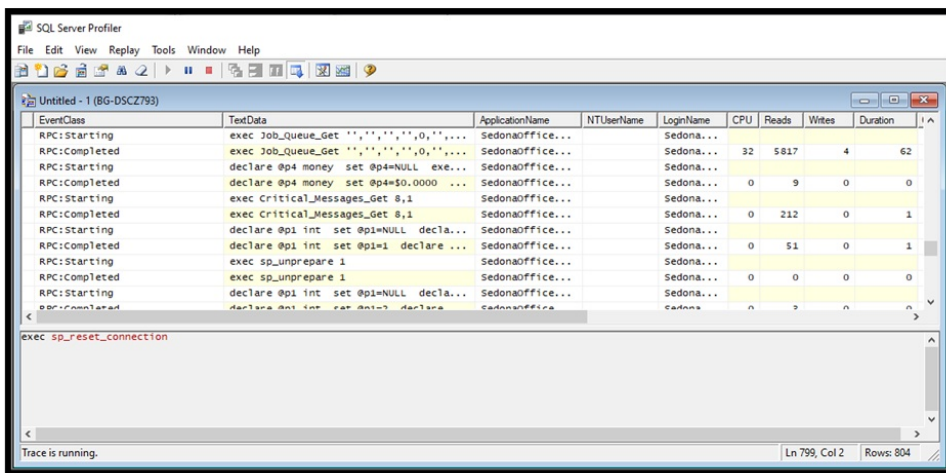
Click OK

Once the events and filters are correct click Run on the Trace Properties.

Once running it will capture the activity between the application and the server.

Perform the action in the application that is causing the error. Once the error is received you can Stop the Trace and save the file.

The trace will show every transaction between the application and server and can be used to see what event caused the errors. The saved trace file can be attached to the support case or emailed to Support for analysis.



The screenshot shows the SQL Server Profiler interface with a trace running. The main window displays a table of trace events. The columns are: EventClass, TextData, ApplicationName, NTUserName, LoginName, CPU, Reads, Writes, and Duration. The trace is currently running, as indicated by the status bar at the bottom which says "Trace is running." and shows "Ln 799, Col 2" and "Rows: 804".

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration
RPC:Starting	exec Job_queue_Get ''','','','0','...	SedonaOffice...		Sedona...				
RPC:Completed	exec Job_queue_Get ''','','','0','...	SedonaOffice...		Sedona...	32	5817	4	62
RPC:Starting	declare @p4 money set @p4=NULL exe...	SedonaOffice...		Sedona...				
RPC:Completed	declare @p4 money set @p4=50.0000 ...	SedonaOffice...		Sedona...	0	9	0	0
RPC:Starting	exec Critical_Messages_Get 8,1	SedonaOffice...		Sedona...				
RPC:Completed	exec Critical_Messages_Get 8,1	SedonaOffice...		Sedona...	0	212	0	1
RPC:Starting	declare @p1 int set @p1=NULL decla...	SedonaOffice...		Sedona...				
RPC:Completed	declare @p1 int set @p1=1 declare ...	SedonaOffice...		Sedona...	0	51	0	1
RPC:Starting	exec sp_unprepare 1	SedonaOffice...		Sedona...				
RPC:Completed	exec sp_unprepare 1	SedonaOffice...		Sedona...	0	0	0	0
RPC:Starting	declare @p1 int set @p1=NULL decla...	SedonaOffice...		Sedona...				
RPC:Completed	declare @p1 int set @p1=1 declare ...	SedonaOffice...		Sedona...	0	9	0	0

exec sp_reset_connection

Trace is running. Ln 799, Col 2 Rows: 804