How to collect data for SSMS troubleshooting

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:32 AM PST

Contents

You might come across with issues where the customer reports that he is experiencing problems with SSMS. Note that the scope of an SSMS issue would be with the usage of the tool. For example:

- slowness on the GUI
- unexpected behavior of the GUI

How SSMS works in the background

SSMS will just perform T-SQL to populate the GUI. Like so, if for example the customer is complaining with timeouts and / or slowness on the GUI you might want approach the problem from a general performance view. A common scenario is for customers to report that they cannot expand a certain area on the SSMS Object Explorer, resulting on a timeout. Most of the times this caused by blocking, like so the problem is not with SSMS but with transaction that is causing blocking.

Try to reproduce the problem and collect details

Try to reproduce the problem, making sure that you are using the same SSMS version as the customer. In some cases this might be hard since it might be a combination of different variables. For example, number of databases, number of tables or other

Collect details

Like mentioned above, SSMS will execute queries to populate the GUI. For example, to populate the tables on Object explorer it will use several DMVs like **sys.tables** among others. Let's look at some of the ways that we can use to collect data that might be useful when troubleshooting. Also will save you time when engaging the PG.

Output

This window can be exposed on SSMS by going into **View** -> **Output**. This will basically output every query that is being executed by SSMS to populate the Object Explorer.

Profiler and xEvents

Sometimes we need more details on processes that run out of the Object Explorer. For example, from the query designer. Like so, and since SSMS uses T-SQL, all commands can be captured by using SQL Profiler or xEvents.

Internal reference

325753750

How good have you found this content?