Subject: Re: ODBC Assertion failed
Posted by mirek on Tue, 22 Dec 2020 17:22:49 GMT
View Forum Message <> Reply to Message

Giorgio wrote on Tue, 22 December 2020 11:44mr_ped wrote on Fri, 18 December 2020 14:02Giorgio wrote on Fri, 18 December 2020 08:26I've found out that the problem arises only in debug mode. The problem is the ASSERT at line 200 and 215 in ODBC/ODC.cpp. I can live without having the capability to debug that part of code, although it's very handy when there are issues (a problem with that part of the code was the reason I turned on the debugger in the first place).

Usually ASSERT compiles to nothing in release, and to debug check under debug.

So "arises only in debug mode" means that you see the warning in debug mode, but in release the same condition, which would trigger assert, does nothing and continues further.

ASSERT usually ensures conditions which "must be true" (and there is no need to check it at runtime) and the code after them usually is written in such a way that it may often crash hard if the condition is false.

So you get assert warning only in debug mode, but most likely the real problem is happening in release too, it's just not catched early by ASSERT, but causes havoc further down the line. In some situations you may be lucky that the code after is robust enough to survive even when condition is false, and the assert is rather just information that something is seriously broken in the code, way beyond the author expectations.

I think I found the problem. In the ODBC there is a member called tmode of type "TransactionMode". TransactionMode can be NORMAL or IMPLICIT. There is also the related method SetTransactionMode to modify tmode. For some reason it was set to NORMAL and that caused the issue. Setting it to IMPLICT fixed the problem.

Unfortunately I did not found any reference to SetTransactionMode in the forum so the only way to understand the problem was inspecting the ODBC.h and ODBC.cpp code. The problem was specifically related to MSSQLSession, so nothing in the query itself was wrong and changing it would not have helped. I do not think that my code was "seriously broken in the code, way beyond the author expectations" and I do think that who wrote that piece of code could have easily given some hint about the origin of the problem.

Sorry for the late reply, only now I have checked all details.

The ASSERT should basically mean that there is mismatch between Begin and Commit/Rollback pairs.

NORMAL transaction mode means you are using Begin/Commit/Rollback. Without Begin, there is commit after each statement. Transactions can be nested (it that case, only top-level transaction is relevant).

In IMPLICIT mode, each commit or rollback starts another transaction. Begin is the same as commit, mismatch between Begin/Commit is ignored. That is why you are not seeing the problem with IMPLICIT mode...

That said, I cannot construct a scenario which would lead to failure you describe. Would it be possible to insert DDUMP(tlevel) before both ASSERTs? (and post here the result...)

Mirek