Different execution plan when executing from SSMS

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

Contents

Issue

In some cases, when trying to testing the query performance, you might face situations where the execution is fast on SSMS and slow on application.

Sometimes the performance is not different, but the execution plan might be different of what you expect (for example, the Actual Execution plan that you got is different of what you see on Query Store)

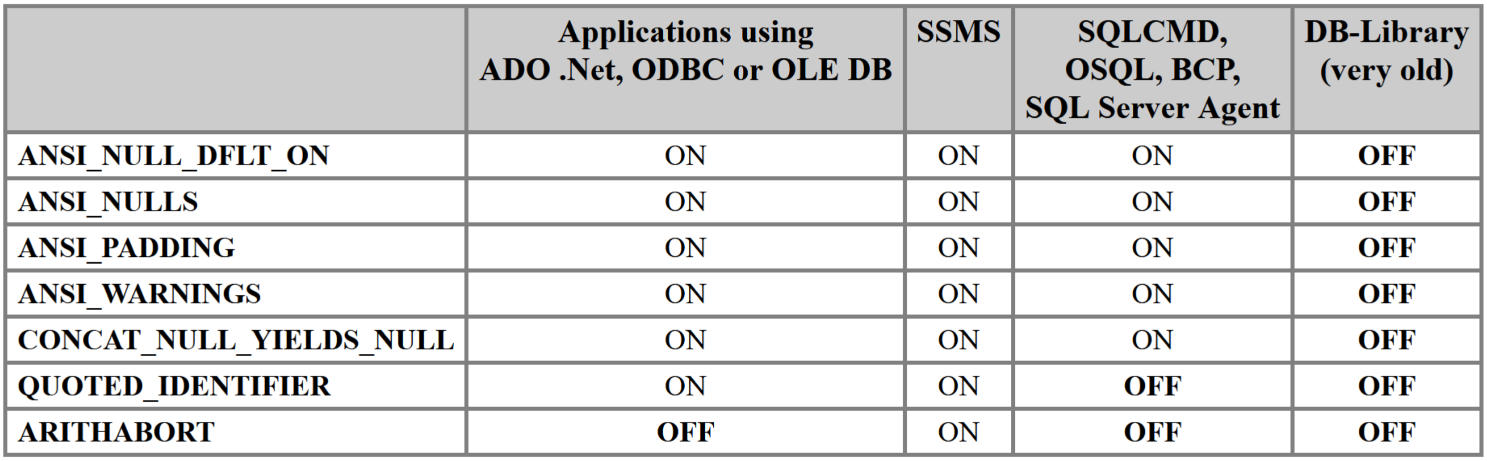
Investigation/Analysis

This can have different causes:

- your run compiled a new execution plan because the query text is not exactly the same as the one executed by the application (for example, variable names)
- statistics or schema changed so the plan recompiled. Also, maybe the plan is no longer in cache and on this new execution a compilation occurred.

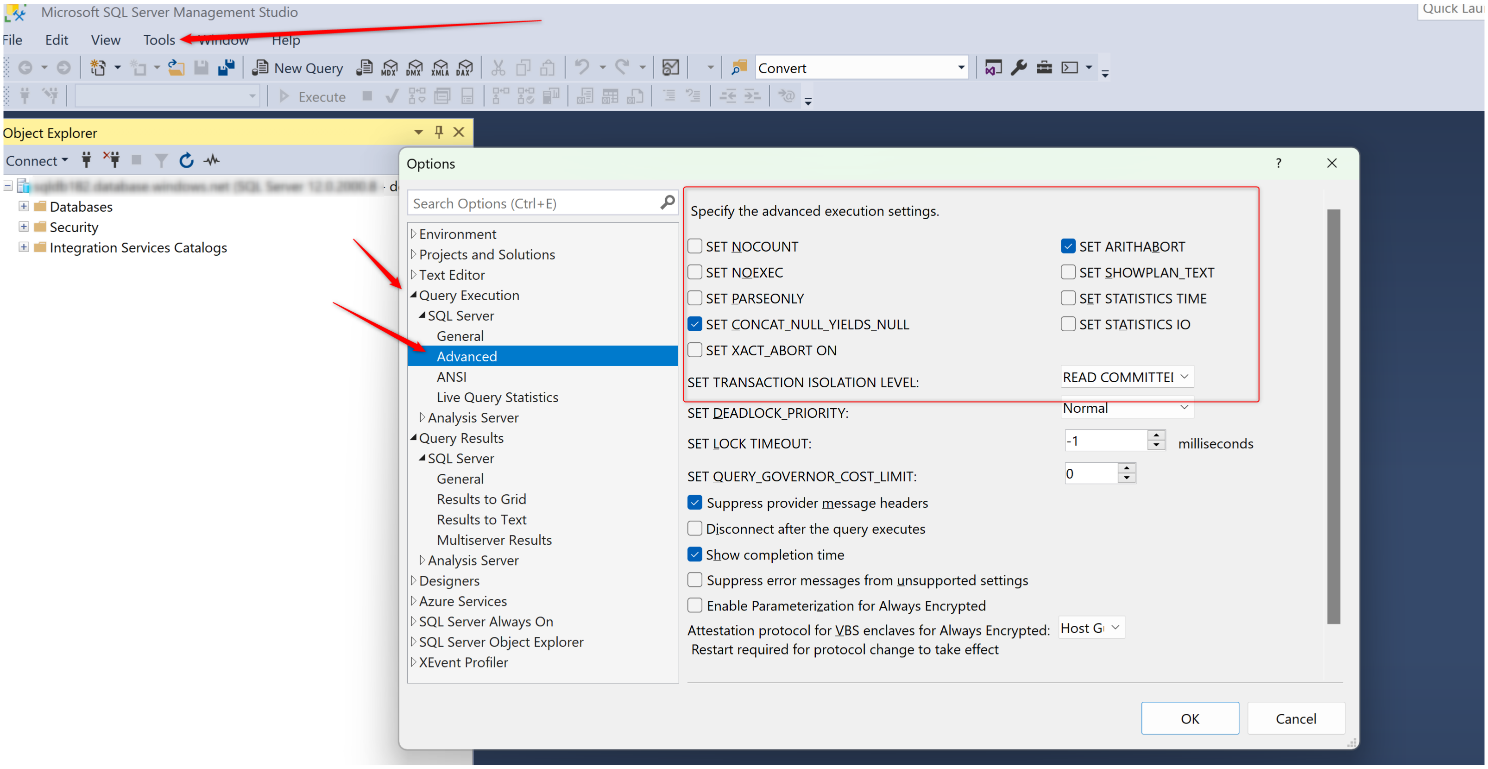
```
MonWiQueryParamData | where LogicalServerName == "v0lmyvu2oy" and logical_database_name == "hosting"
| where query_hash == ""
| project TIMESTAMP, plan_id, compile_code
```

- the SET operators are different - on SSMS the default set operators are different when compared with applications. The matrix looks like this (table taken from Slow in the Application, Fast in SSMS?):



	Applications using ADO .Net, ODBC or OLE DB	SSMS	SQLCMD, OSQL, BCP, SQL Server Agent	DB-Library (very old)
ANSI_NULL_DFLT_ON	ON	ON	ON	OFF
ANSI_NULLS	ON	ON	ON	OFF
ANSI_PADDING	ON	ON	ON	OFF
ANSI_WARNINGS	ON	ON	ON	OFF
CONCAT_NULL_YIELDS_NULL	ON	ON	ON	OFF
QUOTED_IDENTIFIER	ON	ON	OFF	OFF
ARITHABORT	OFF	ON	OFF	OFF

On SSMS you can change this settings by going to **Tools**-> **Options** -> **Query Execution** -> **SQL Server** -> **Advanced**



Specify the advanced execution settings.

☐ SET NOCOUNT
☐ SET NOEXEC
☐ SET PARSEONLY
☑ SET CONCAT_NULL_YIELDS_NULL
☐ SET XACT_ABORT ON
☑ SET ARITHABORT
☐ SET SHOWPLAN_TEXT
☐ SET STATISTICS TIME
☐ SET STATISTICS IO

SET TRANSACTION ISOLATION LEVEL: READ COMMITTEI

SET DEADLOCK_PRIORITY: Normal

SET LOCK TIMEOUT: -1 milliseconds

SET QUERY_GOVERNOR_COST_LIMIT: 0

☑ Suppress provider message headers
☐ Disconnect after the query executes
☑ Show completion time
☐ Suppress error messages from unsupported settings
☐ Enable Parameterization for Always Encrypted

Attestation protocol for VBS enclaves for Always Encrypted: Host G
Restart required for protocol change to take effect

Just an example on how ARITHABORT might affect the troubleshooting results (and some performance baselines that you might want to set), with SET ARITHABORT ON the cached plan that exists with SET ARITHABORT OFF will not be reused. Like so you might not get the same execution plan.

Mitigation

- make sure that the query text is exactly the same
- make sure that there wasn't a plan recompile due to schema or statistics change.
- make sure that SET ARITHBORT value reflects what is executed on the application.

More information

Slow in the Application, Fast in SSMS?

Public Doc Reference

ARITHABORT

How good have you found this content?

🙂 🙁