Error 13609: JSON text is not properly formatted

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

Contents

Issue

When running a JSON formatted query on SQL Database it fails with a 13609 error:

"Msg 13609, Level 16, State 1, Line 41 JSON text is not properly formatted. Unexpected character '>' is found at position 0."

Investigation/Analysis

This issue was first noticed on September 3, 2022 and is caused by a regression introduced in T65. Customers confirmed that the affected queries were previously running without problems before T65 was rolled-out.

An example query:

```
DECLARE @ValuationDate datetime=null
SET @ValuationDate='2022-08-31'
   SELECT [ProjectID],
          [ProjectName],
          [ProjectDesc],
          [Author],
          [Notes]
    FROM Project(NOLOCK)
    WHERE Replace(JSON_VALUE(DataStructure,'$.ValuationDate'),'T00:00:00','')
            = COALESCE(@ValuationDate, Replace(JSON_VALUE(DataStructure,'$.ValuationDate'),'T00:00:00','') )
    AND IsActive = 1
```

Mitigation

There is a T65 reset build (with the hotifx - 16.0.816.214) that contains the corrected code and is currently being rolled out worldwide (as of October 6, 2022).

If the customer is impatient for the mitigation the Product Group can also apply the hotfix directly to the affected databases. This requires an IcM to the **SQL DB Perf: XML/JSON** team.

RCAs

Public RCA JSON functions are implemented in SQL Server 2016 on top of NVARCHAR type to support Unicode characters (per JSON standard). JSON text is represented and processed as NVARCHAR type in SQL Server engine. Using VARCHAR type requires an implicit conversion to NVARCHAR type. Conversion from VARCHAR might not be correct and depending on the input sequence, the SQL Server engine might represent non-unicode characters as '????' or produce unexpected characters. This is well known conversion issue in SQL Server engine, and the recommendation is to always use NVARCHAR to represent JSON text.

Due to a bug in the code, conversion from VARCHAR into NVARCHAR (Unicode UTF-16) strings in JSON_VALUE function was skipped under a set of specific circumstances, which caused parsing of JSON text to fail. Using NVARCHAR type or LOBs (e.g., VARCHAR(MAX) or NVARCHAR(MAX)) would avoid conversion errors. We have identified the problem and reverted the code that introduced it. The code fix with the correct conversion logic will be (re)deployed in the following days.

We apologize for any inconvenience or disruption this problem may have caused.

Engineering RCA There is regression introduced to JSON_VALUE function, when we tried to address another defect where JSON object larger than 4k characters and stored as VARCHAR(8000) (or smaller) gets silently truncated upon implicit conversion to NVARCHAR(4000). This conversion is required as JSON parser expects wide string input. In regressed code, we tried to handle this case and make conversion within the function, only if the input was of type VARCHAR (non-LOB). To check input argument type, we used wrong and unreliable function CXVariant::Xvt(). If this function returns wrong type, even if it is VARCHAR, parsing of JSON will fail with the error message 13609, as conversion will not happen, and JSON parser expects wide strings.

The fix has been coded and ported to T65. There is a new build with the fix and the T65 will be reset.

Internal Reference

IcMs
IcM 334496118: Issue with the JSON format after the patch upgrades happened on Azure SQL Server
Umbrella ICM, for linking mitigated incidents
IcM 337820286: Cx is getting error on queries that using JSON_VALUE function in WHERE condition after planned deployment today
IcM 337899555: Stored procedures failing after code package version change
IcM 338103905: Sql error number: 13609. Error Message: JSON text is not properly formatted. Unexpected character '>' is found at position 0
IcM 338527973: SQL Exceptions started occuring after a health Event. JSON text is not properly formatted. Unexpected character \'(ก)\' is found

Engineering Work Items
2008890: Extend test coverage for JSON_VALUE to include different input sources
1979962: Avoid direct usage of Xvt from CXVariant in Retail code inside the intrinsics

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:
/Root Cause: Azure SQL v3/Performance/SQL Server Engine bug/Other

How good have you found this content?