

## Base Language - Bug #7247

### Dataset:READ-XML does nothing

04/03/2023 04:36 PM - Vladimir Tsichevski

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Igor Skorniyakov	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Related issues:</b>			
Related to Base Language - Bug #7193: READ-JSON is not supported for the "LON...		<b>Closed</b>	
Related to Database - Feature #6444: dataset improvements		<b>Closed</b>	
Blocks Conversion Tools - Feature #6237: OEUnit support		<b>Closed</b>	

### History

#### #1 - 04/03/2023 04:40 PM - Vladimir Tsichevski

The test case is:

```
using OpenEdge.Core.Assert.  
using OpenEdge.Core.AssertionFailedError.  
  
routine-level on error undo, throw.  
  
class oo.TestDataset:  
  
    @Test.  
    METHOD PUBLIC VOID testDatasetFromXML()  
        DEFINE VARIABLE dsData AS HANDLE NO-UNDO.  
        DEFINE VARIABLE xml AS LONGCHAR NO-UNDO.  
  
        CREATE DATASET dsData.  
        xml = "<?xml version=~"1.0~"?">  
            + "<ProDataSet>  
            +   "<Customer>  
            +     "<customerName>John Smith</customerName>"  
            +     "<customerNum>10</customerNum>"  
            +   "</Customer>  
            +   "<Customer>  
            +     "<customerName>Freddie Mercury</customerName>"  
            +     "<customerNum>11</customerNum>"  
            +   "</Customer>  
            +   "<Order>  
            +     "<orderNum>1</orderNum>"  
            +     "<customerNum>10</customerNum>"  
            +     "<address>Address 1</address>"  
            +   "</Order>  
            + "</ProDataSet>".  
        Assert:IsTrue(dsData:READ-XML("longchar", xml, ?, ?, ?)).  
        Assert:Equals(2, dsData:NUM-TOP-BUFFERS).  
    END METHOD.  
  
end class.
```

The test compiles and runs OK. But in FWD the READ-XML method reads no data since the dataset has no buffers, so the second assertion fails. In OE the test passes OK.

**#2 - 04/03/2023 04:41 PM - Vladimir Tsichevski**

- *Blocks Feature #6237: OJUnit support added*

**#3 - 04/03/2023 04:42 PM - Vladimir Tsichevski**

- *Related to Bug #7193: READ-JSON is not supported for the "LONGCHAR" source type added*

**#4 - 04/03/2023 04:50 PM - Igor Skornyakov**

Vladimir Tsichevski wrote:

The test case is:

[...]

The test compiles and runs OK. But in FWD the READ-XML method reads no data since the dataset has no buffers, so the second assertion fails.

In OE the test passes OK.

This looks to be similar to the situation with READ-JSON. For the just created dynamic DATA-SET on READ-XML we have to infer its structure from the XML data. I'm working now on the same problem for READ-JSON (currently it is suspended). For RAD-XML this should be similar but more complicated because of XML prefixes/namespaces. I think that the same problem exists for TEMP-TABLE:READ-XML (already done for READ-JSON).

I've not considered these uses cases when working on [#6444](#), sorry

**#5 - 04/05/2023 02:00 AM - Igor Skornyakov**

- *Related to Feature #6444: dataset improvements added*

**#6 - 04/15/2023 02:32 PM - Igor Skornyakov**

- *Assignee set to Igor Skornyakov*

- *Status changed from New to WIP*

**#7 - 04/18/2023 11:02 AM - Igor Skornyakov**

Created task branch 7247a.

**#8 - 05/24/2023 03:50 PM - Igor Skornyakov**

Extracted common code in XmlImport/JsonImport to a base class.

Committed to 7247a/14585.

**#9 - 05/26/2023 11:26 AM - Igor Skornyakov**

Implemented inferring temp-table structure from XML.

Committed to 7247a/14588.

**#10 - 05/26/2023 11:33 AM - Igor Skornyakov**

Rebased 7247a to trunk/14590.  
Pushed up to revision 14592.

**#11 - 05/26/2023 01:06 PM - Greg Shah**

What is left to do in this task? Please set the %Done. Also, move it to review status if it is ready for review.

**#12 - 05/26/2023 01:21 PM - Igor Skornyakov**

- % Done changed from 0 to 60

Greg Shah wrote:

What is left to do in this task? Please set the %Done. Also, move it to review status if it is ready for review.

I'm working now on inferring DATASET structure from XML. After that I will need to add inferring of the NAMESPACE/PREFIX which should not be a big deal.

I've initially expected that it will be more or less the same as for READ-JSON (except namespace/prefix), but it appears to be more tricky, in particular for the nested relations.

I plan to work on it at the weekend and finish at the beginning of the next week.

**#13 - 05/29/2023 04:37 PM - Igor Skornyakov**

- % Done changed from 60 to 80

Inferring DATASET from XML is almost done.  
To be done:

1. Inferring RELATIONS
2. Supporting attributes
3. Supporting XML NAMESPACES.

**#14 - 05/30/2023 11:12 AM - Igor Skornyakov**

Added support for inferring DATASET/TEMP-TABLE structure from XML (w/o namespaces)

Committed to 7247a/14596.

The only remaining thing is inferring prefixes/namespaces. Working on this.

## #15 - 05/31/2023 08:57 AM - Igor Skornyakov

- Status changed from WIP to Review

Finished.

Committed to 7247a/14597.

Please review.

Thank you.

## #16 - 06/07/2023 01:54 PM - Greg Shah

- % Done changed from 80 to 100

Ovidiu: Please review 7247a.

## #17 - 06/07/2023 09:31 PM - Ovidiu Maxiniuc

Review of 7247a/r14621.

Fully unifying the de/serialization of XML and JSON is an ambitious task I think of for some time, but I did not have time to do it myself. Nice job so far!

- `StaticTempTable.java`: The new method `readTable()` declares it throws `PersistenceException` but in fact a new `IllegalStateException()` is thrown. I understand that invoking this method for a `StaticTempTable` is never expected, but the current code will abend the client in the event that happens. Throwing a `PersistenceException` is likely to be handled at some superior level so we should throw it instead. Adding the `@Override` to this method would be also recommended.
- `LookAheadXmlStreamReader.java`: please move/align the throws options on its own line in methods declaration (lines 197, 277, 398, 423, 463, 527, 542, 550) like you did for `peek()` at line 185. The methods from lines 194 to EoF have only `@see` tag in the javadoc. While this might seem enough for methods like `close()`, a short description is recommended. There are some methods whose implementation is more than delegating the call to original XML stream reader;
- `Importer.java`:
  - line 346: please add a comment with the text of error being recorded. It is easier for the reader to understand what 15366 error is;
  - lines 442, 613: the TODO was left behind. Let's treat these by throwing `PersistenceException` with an appropriate messages;
- `XmlImport.java`
  - in `readTable()` (line 865): I was about to ask if we extract the `ctx.ttName` length check before assigning `groupSep` and `decSep` but I saw that the finally not only restores the numeric format possibly altered by `readRecord()` but also invokes `popContext()`; What happens with the context when `!setParentValues(tableName)` (line 859)? Maybe we should move that too in the try block?
  - method `typeOf()`: the throw should be on separate line. Is it normal to always return `ParmType.CHAR` here? In other words, are all the inferred fields character typed in XML as opposed to JSON where there are half a dozen?

Other issues related to code formatting:

- separate individual methods (and fields) by an empty line;
- in method javadocs, separate the main comment, `@param`, `@return` and `@throw` sections by empty lines;
- the parameters and their description (also `@return` and `@throw` text) must be aligned at a TAB length (but NOT the 'hard' tab (`\t`) character).

## #18 - 06/08/2023 05:05 AM - Igor Skornyakov

Ovidiu Maxiniuc wrote:

Review of 7247a/r14621.

Fully unifying the de/serialization of XML and JSON is an ambitious task I think of for some time, but I did not have time to do it myself. Nice job so far!

- `StaticTempTable.java`: The new method `readTable()` declares it throws `PersistenceException` but in fact a new `IllegalStateException()` is thrown. I understand that invoking this method for a `StaticTempTable` is never expected, but the current code will abend the client in the event that happens. Throwing a `PersistenceException` is likely to be handled at some superior level so we should throw it instead. Adding the `@Override` to this method would be also recommended.

Fixed

- `LookAheadXmlStreamReader.java`: please move/align the throws options on its own line in methods declaration (lines 197, 277, 398, 423, 463, 527, 542, 550) like you did for `peek()` at line 185. The methods from lines 194 to EoF have only `@see` tag in the javadoc. While this might seem enough for methods like `close()`, a short description is recommended. There are some methods whose implementation is more than delegating the call to original XML stream reader;

Fixed.

- `Importer.java`:
  - line 346: please add a comment with the text of error being recorded. It is easier for the reader to understand what 15366 error is;
  - lines 442, 613: the TODO was left behind. Let's treat these by throwing `PersistenceException` with an appropriate messages; \*

Fixed.

`XmlImport.java`

- in `readTable()` (line 865): I was about to ask if we extract the `ctx.ttName` length check before assigning `groupSep` and `decSep` but I saw that the finally not only restores the numeric format possibly altered by `readRecord()` but also invokes `popContext()`;: What happens with the context when `!setParentValues(tableName)` (line 859)? Maybe we should move that too in the try block?

Yes, you're right, thank you. Fixed.

- method `typeOf()`: the throw should be on separate line. Is it normal to always return `ParmType.CHAR` here? In other words, are all the inferred fields character typed in XML as opposed to JSON where there are half a dozen?

Yes, this is how 4GL works (tested with all data types).

Other issues related to code formatting:

- separate individual methods (and fields) by an empty line;
- in method javadocs, separate the main comment, `@param`, `@return` and `@throw` sections by empty lines;
- the parameters and their description (also `@return` and `@throw` text) must be aligned at a TAB length (but NOT the 'hard' tab (`\t`) character).

Fixed where noticed.

Committed to 7247a/14622.

Can I merge 7247a to the trunk?

Thank you.

**#19 - 06/08/2023 06:37 AM - Greg Shah**

Ovidiu: Is 7247a OK to merge?

**#20 - 06/08/2023 01:40 PM - Ovidiu Maxiniuc**

Igor,  
I have fixed some formatting issues and enhanced javadoc texts for some methods for you in r14623. If you agree with my changes, you can merge to trunk.

**#21 - 06/08/2023 02:11 PM - Igor Skornyakov**

Ovidiu Maxiniuc wrote:

Igor,  
I have fixed some formatting issues and enhanced javadoc texts for some methods for you in r14623. If you agree with my changes, you can merge to trunk.

Thank you, Ovidiu!

7247a had been merged to the trunk rev. 14619 and archived.

**#22 - 06/12/2023 07:42 AM - Vladimir Tsichevski**

All related unit tests pass for reading dataset from XML.

**#23 - 06/12/2023 08:55 AM - Greg Shah**

- *Status changed from Review to Closed*