

Base Language - Support #5111

TempTableSchema & BufferField

01/27/2021 12:17 PM - Marian Edu

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 01/27/2021 04:47 PM - Greg Shah

- Description updated

From Marian:

This might be an easy one but damn if I can figure out what the connection is between those dmo temp-table schema/buffer and the buffer/bufferfield handles. Currently although omit initial values parameter exists in json serialization routines that seems to be simply ignored for now hence our tests fails so thought might be a good idea to implement that while at it but then how one could tell if a particular TempTableSchema.Column in a TempRecord has the default value?

In 4GL we would check buffer-field default-value against buffer-value and after going through all unwrapping to buffer then buffer field it looks like the code gets converted in FWD to something like this:

```
if (!_isEqual(htt.unwrapBuffer().bufferField(1).unwrapBufferField().getDefaultValue(), htt.unwrapBuffer(
).bufferField(1).unwrapBufferField().value()))
{
    message("not default");
}
```

Problem is, how to get from those 'persist' Temporary objects back to the buffer handle, or is there a better (easier) way?

#2 - 01/27/2021 04:47 PM - Greg Shah

Ovidiu/Eric: Please advise.

#3 - 01/27/2021 05:16 PM - Ovidiu Maxiniuc

The TempTableSchema.Column has a method isChanged(BaseDataType). If you pass the current value of the field to this method of its Column, it will return whether the field was altered from its initial value.

However, this method is already used in JsonExport.writeRecord() for the exact goal:

```
for (TempTableSchema.Column column : schema.columns())
{
    [...]
    if (noInit)
    {
        if (!column.isChanged((BaseDataType) column.getGetter().invoke(dmo)))
        {
            continue; // skip to next column
        }
    }
    writeDatum(json, column, dmo, null);
    [...]
}
```

#4 - 01/28/2021 01:45 AM - Marian Edu

Ovidiu Maxiniuc wrote:

However, this method is already used in JsonExport.writeRecord() for the exact goal:

This is exactly where I was looking to use it, this is probably from a newer version as that heck on noInit is clearly missing in our version. The last revision entry in JsonExport is this:

```
007 CA 20200927 Use IdentityHashMap instead of plain map when the key is a Class.
```

Greg, maybe it's time for a new branch, it looks like there could potentially be some merge conflicts anyway :(

#5 - 01/28/2021 09:03 AM - Greg Shah

maybe it's time for a new branch, it looks like there could potentially be some merge conflicts anyway :(

Yes.

Ovidiu's recent changes from [#4397](#) and [#3574](#) are in 4397a. The change mentioned above may be there. Either way, I think we should get the changes from 4397a into 3821c if it is safe.

I would like to get this merge done and then push that code to 3821c on xfer. You could take that branch and merge your 4384g changes there, then overwrite the 4384g branch. I can then easily merge that branch back into our 3821c and make a 4384h from that.

#6 - 01/28/2021 10:12 AM - Ovidiu Maxiniuc

There is a typo. Actually the task is [#4397](#) and branch 4397a. The current revision 11986 is generally stable. There are a couple of known issues I am actively working on. I expect my branch will be merged into 3821c by Monday.

#7 - 01/28/2021 11:04 AM - Greg Shah

Are these known issues something that will make 3821c worse? Or are they also broken already in 3821c?

#8 - 01/28/2021 11:20 AM - Ovidiu Maxiniuc

No, although I never converted and ran a large customer application I do not think merging 4397a will worsen 3821c. It has lots of fixes and improvements like the one which started this thread.

#9 - 01/28/2021 11:33 AM - Greg Shah

OK, unless Eric objects, please go ahead with the merge today. This will allow 2 things:

1. Eugenie can do a full conversion on a large customer application. If there are issues, post them here and Ovidiu will be quick to respond. :)
2. We can move ahead more quickly with the [#4384](#) merge.

#10 - 01/28/2021 12:42 PM - Eric Faulhaber

I will do a code review of the latest changes in 4397a.

#11 - 01/29/2021 07:33 PM - Eugenie Lyzenko

Greg Shah wrote:

OK, unless Eric objects, please go ahead with the merge today. This will allow 2 things:

1. Eugenie can do a full conversion on a large customer application. If there are issues, post them here and Ovidiu will be quick to respond. :)

We have conversion failure. Not sure if I can publish customer specific log here. Will be sent additionally.

The stack trace:

```
...
[java] EXPRESSION EXECUTION ERROR:
[java] -----
[java] ddl.generateTableDDLs(ddlSchema)
[java] ^ { java.lang.NullPointerException }
[java] -----
[java] ERROR:
[java] com.goldencode.p2j.pattern.TreeWalkException: ERROR! Active Rule:
[java] -----
[java]         RULE REPORT
[java] -----
[java] Rule Type : POST
[java] Source AST: [ mtc ] DATA_MODEL/ @0:0 {116964844371969}
[java] Copy AST : [ mtc ] DATA_MODEL/ @0:0 {116964844371969}
[java] Condition : ddl.generateTableDDLs(ddlSchema)
[java] Loop      : false
[java] --- END RULE REPORT ---
[java]
[java]
[java]
[java]     at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:1070)
[java]     at com.goldencode.p2j.convert.TransformDriver.processTrees(TransformDriver.java:570)
[java]     at com.goldencode.p2j.convert.ConversionDriver.generateDMOs(ConversionDriver.java:732)
[java]     at com.goldencode.p2j.convert.ConversionDriver.middle(ConversionDriver.java:465)
[java]     at com.goldencode.p2j.convert.TransformDriver.executeJob(TransformDriver.java:954)
[java]     at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:1024)
[java] Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:5
[java]     at com.goldencode.expr.Expression.execute(Expression.java:484)
[java]     at com.goldencode.p2j.pattern.Rule.apply(Rule.java:497)
[java]     at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:745)
[java]     at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:712)
[java]     at com.goldencode.p2j.pattern.Rule.apply(Rule.java:534)
[java]     at com.goldencode.p2j.pattern.RuleContainer.apply(RuleContainer.java:585)
[java]     at com.goldencode.p2j.pattern.RuleSet.apply(RuleSet.java:98)
[java]     at com.goldencode.p2j.pattern.PatternEngine.apply(PatternEngine.java:1652)
[java]     at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1531)
[java]     at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1479)
[java]     at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:1034)
[java]     ... 5 more
[java] Caused by: java.lang.NullPointerException
[java]     at com.goldencode.p2j.persist.orm.DDLGeneratorWorker$Helper.generateWordTable(DDLGeneratorWorker.java:1468)
[java]     at com.goldencode.p2j.persist.orm.DDLGeneratorWorker$Helper.generateTableDDLImpl(DDLGeneratorWorker.java:1421)
[java]     at com.goldencode.p2j.persist.orm.DDLGeneratorWorker$Helper.lambda$generateTableDDLs$0(DDLGeneratorWorker.java:997)
[java]     at com.goldencode.p2j.persist.orm.DDLGeneratorWorker$Helper.forAllDialects(DDLGeneratorWorker.java:1168)
[java]     at com.goldencode.p2j.persist.orm.DDLGeneratorWorker$Helper.generateTableDDLs(DDLGeneratorWorker.java:976)
[java]     at com.goldencode.expr.CE5809.execute(Unknown Source)
[java]     at com.goldencode.expr.Expression.execute(Expression.java:391)
[java]     ... 15 more
...
```