

Base Language - Bug #5202

Temp Table serialize-name/xml-node-name

03/16/2021 03:43 AM - Marian Edu

Status:	Test	Start date:	
Priority:	Normal	Due date:	
Assignee:	Ovidiu Maxiniuc	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#1 - 03/16/2021 03:44 AM - Marian Edu

While the Table annotation has both the xmlNodeName and serializeName info, that is not available in TempTableSchema - getName and getTableName returns the same name value. For TempTableSchema.Column it was taken a step further, we have getName and getFieldname but this does not allow having both xml-node-name and serialize-name to be set. While one can be used as a default for the other in 4GL and both defaults to the table/field name if not set, one can certainly set them to different values for xml/json serialization purposes.

This could probably be easily fixed/extended but maybe there is already fixed or someone else already working on it.

#2 - 03/16/2021 03:49 AM - Constantin Asofiei

Eric/Ovidiu: please comment what's the proper way of resolving the xmlNodeName, serializeName for a temp-table.

#3 - 03/16/2021 08:43 AM - Greg Shah

- Assignee set to Ovidiu Maxiniuc

#4 - 03/16/2021 11:23 AM - Ovidiu Maxiniuc

- Status changed from New to WIP

We do not use the Table annotation directly, because that is the static hardcoded value. Instead, the values of its attributes are used each time an instance of the table is created in order to configure the initial values for the runtime/dynamic attributes, which are stored in TableMapper structures.

The first thing the Column constructor does is:

```
SerializeOptions opts = field.getSerializeOptions();
```

The xmlNodeName and serializeName are members of this structure and they are used on next lines to compute the serialization name (serName) which is then saved into Column.name. The xmlNodeName and serializeName are mutable (for temp-tables) and can dynamically be altered by the 4GL programmer. Column objects are created each time the serialization occurs so the constructor always computes the updated value for its name field.

So the name field in TempTableSchema.Column is the *name of element or attribute in serialized form* (as documented in javadoc), not the actual name of the table to be serialized. Is this an issue? If so, please let me know your exact problem. There is a infinitesimal chance that the Column.name is not correctly computed in some cases; the algorithm for it value was developed upon both existing 4GL documentation and manual tests.

#5 - 03/16/2021 11:34 AM - Greg Shah

Aren't these changes in a very recent revision of 3821c? I think they are not present in the latest 4384i. Perhaps we need to merge the latest 4384i code into 3821c and make a 4384j.

#6 - 03/16/2021 11:51 AM - Ovidiu Maxiniuc

The code in Column constructor is as old as the class itself. What has changed recently is the data structure and the internal management of TableMapper. There was a flaw (and probably 4384i inherited it if it was not rebased to one of the latest 3821c) which caused some attributes (like xmlNodeName and serializeName) to leak from one temp-table to another, if their structure were close enough. Meaning setting one of these attribute on field f1 of table tt1 was causing the same attribute to be also visible in field f1 of table tt2 (see [#5176](#)). If this is the case for the initial issue of this task, then yes, rebasing or merging with 3821c should fix the issue.

#7 - 03/16/2021 01:02 PM - Marian Edu

Ovidiu Maxiniuc wrote:

The xmlNodeName and serializeName are members of this structure and they are used on next lines to compute the serialization name (serName) which is then saved into Column.name.

Yes but, you only have one property for this name to keep what it should be two different properties - xml-node-name and serialize-name. Both defaults to table/field name, serialize-name (if set) serves as default to xml-node-name but not the other way around.

So the name field in TempTableSchema.Column is the *name of element or attribute in serialized form* (as documented in javadoc), not the actual name of the table to be serialized. Is this an issue? If so, please let me know your exact problem.

Yes, there should be two separate attributes and preferable keep the names close to the 4GL ones - just like in Dataset, getName is imho misleading and I see method name much more useful than it's attached javadoc but it's probably just me :)

There is a infinitesimal chance that the Column.name is not correctly computed in some cases; the algorithm for it value was developed upon both existing 4GL documentation and manual tests.

As said this is an attribute that does not exist in 4GL, it's somehow mixing xml-nde-name and serialize-name giving precedence to xml-node-name... this should satisfy the needs of xml serialization but does not work for json.

#8 - 03/18/2021 03:25 AM - Marian Edu

OK, actually it does look like the temptable is implementing both NamedSerializable and XmlNode so the info should be there only it isn't... StaticTempTable does get the buffer with associated Table annotation (DMO) only it does nothing with it beside setting the table name, adding one extra line to setSerializeName in ctor solves my issue but then guess there is a bunch of info in that DMO that should have set so right now just applied the fix locally to see JSON serialization works as expected if serialize-name is set.

#9 - 03/18/2021 03:03 PM - Ovidiu Maxiniuc

Nice catch! Indeed, the StaticTempTable does not initialize correctly those attributes. They need to be added. Also specialized testcases in xfer repository (attribute sections) would be also useful.

#10 - 03/22/2021 10:12 AM - Marian Edu

Ovidiu Maxiniuc wrote:

Nice catch! Indeed, the StaticTempTable does not initialize correctly those attributes. They need to be added. Also specialized testcases in xfer repository (attribute sections) would be also useful.

Hi Ovidiu, just checking if this is something you will be handling or expect us to do it? :)

For temp-table field I do think the name property in TempTableSchema.Column is not enough as we need both xml-node-name and serialize-name which in 4GL can be set to different values.

Thanks

#11 - 03/22/2021 12:32 PM - Ovidiu Maxiniuc

Marian Edu wrote:

Hi Ovidiu, just checking if this is something you will be handling or expect us to do it? :)

We will do it since this is more persistence-related.

For temp-table field I do think the name property in TempTableSchema.Column is not enough as we need both xml-node-name and serialize-name which in 4GL can be set to different values.

I understand the precomputed values in TempTableSchema.Column are not enough and also does not give you a great deal of flexibility. So, for this second issue we will keep the reference to LegacyFieldInfo from Column c'tor, avoiding copying the help/format/etc attribute to local fields. The getters for legacy-name, xml-node-name and serialize-name will delegate to that reference. Also the existing Column field will be delegated, but this will be transparent when the API is called from JSON/XML serializers.

#12 - 04/01/2021 09:14 AM - Greg Shah

Is this task complete as of 3821c rev 12222?

#13 - 04/01/2021 10:39 AM - Ovidiu Maxiniuc

Yes, the requested data is available in TempTableSchema.Column starting from rev 12222. Short description of the methods of interest:

- getName() - unchanged. It will return the old pre-compiled value, to keep the existing code working. At first my intention was to mark it as deprecated but I postponed the idea;
- getFieldName() - new. You can use this to obtain the legacy field NAME as it was declared in 4GL;
- getXmlNodeName() - new. Returns the XML-NODE-NAME attribute. If not set, null is returned;
- getSerializeName() - new. Returns the SERIALIZE-NAME attribute. If not set, null is returned.

#14 - 04/01/2021 11:28 AM - Greg Shah

- Status changed from WIP to Test

- % Done changed from 0 to 100

#15 - 04/01/2021 12:42 PM - Marian Edu

Ovidiu Maxiniuc wrote:

Yes, the requested data is available in TempTableSchema.Column starting from rev 12222. Short description of the methods of interest:

- getName() - unchanged. It will return the old pre-compiled value, to keep the existing code working. At first my intention was to mark it as deprecated but I postponed the idea;
- getFieldName() - new. You can use this to obtain the legacy field NAME as it was declared in 4GL;
- getXmlNodeName() - new. Returns the XML-NODE-NAME attribute. If not set, null is returned;

this should default to getSerializeName if not set.

- getSerializeName() - new. Returns the SERIALIZE-NAME attribute. If not set, null is returned.

this should default to getFieldName if not set.

@Greg, are those changes available already in our branch or it's a newer revision and we have to merge back?

#16 - 04/01/2021 01:40 PM - Greg Shah

The changes are not in your branch yet.

Ovidiu: Please make the additional change for the default values and then we can push the latest 3821c to xfer.

Marian: From there, you can rebase your changes and bsr send the update for us to merge back. Then we will open the next branch.

#17 - 04/01/2021 02:30 PM - Ovidiu Maxiniuc

Done. Committed as revision 12237.

#18 - 04/01/2021 03:24 PM - Greg Shah

I've pushed 3821c rev 12240 to xfer.goldencode.com.

Marian: Let's go ahead with the [#4384](#) merge so that you are on the latest code.