Base Language - Feature #4658

OO serialization support

05/29/2020 07:34 AM - Greg Shah

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	10%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to Base Language - Feature #4373: finish core OO 4GL support	**New**
Related to Conversion Tools - Feature #6237: OEUnit support	**Closed**

History

#1 - 05/29/2020 08:25 AM - Greg Shah

OO classes and enums support serialization in the 4GL.

This includes:

- SERIALIZABLE/NON-SERIALIZABLE/SERIALIZE-NAME <name> syntax
  - for classes (CLASS statement) we must handle SERIALIZABLE (NON-SERIALIZABLE and SERIALIZE-NAME cannot be used in class defs)
  - enums are implicitly serializable (no syntax, it just is implemented)
  - all three keywords/clauses can be used with DEFINE DATASET, DEFINE TEMP-TABLE, DEFINE PROPERTY and DEFINE VARIABLE
  - This is a conversion thing and but also will need some runtime support.
  - All parent classes of a serializable class must also be serializable in the 4GL. This has class hierarchy/inheritance implications. It is a compile error if this is not the case.
  - It also may require generation of additional code to handle the actual serialization processing for private members.
- Runtime changes to properly flatten/restore serializable objects for I/O with:
  - Existing built-in OO classes which honor serializaton must have it added (e.g. the various *Error classes).
  - binary streams (check text output too), Progress.IO.BinarySerializer (this can read private members too?)
  - JSON
    - may be in many places, for example the handle based WRITE-JSON() method but also the OO builtin class Progress.IO.JsonSerializer
    - may only reads public members?
    - find the places where there is implicit JSON serialization support
  - appserver
    - the docs suggest this is only for 4GL clients, check this
    - this matters for both exception support (throwing/catching exceptions) and parameter passing
  - other usage? XML?

Some notes and questions about the standard Java Serializable vs the 4GL implementation:

- The implicit/built-in serialization support in Java can possibly be used for the binary form of 4GL serialization. 4GL tests will be needed to check if this can work. My guess is that it probably can work except if we need binary compatibility. I think we probably should have binary compatibility here because customers may have existing serialized data that must be read.
- I do think we probably should implement the Serializable marker interface even if we end up having a custom binary serialization.
, The JSON serialization will need something custom.
- For members that cannot or should not be serialized, there is the transient keyword in Java. Is NON-SERIALIZABLE the 4GL equivalent? This only matters for DEFINE DATASET, DEFINE TEMP-TABLE, DEFINE PROPERTY and DEFINE VARIABLE since NON-SERIALIZABLE cannot be used in a CLASS statement.
- In Java it is allowed to have a parent class that is not serializable, but in such a case the parent class must have a default constructor. In the 4GL the parent classes must be serializable. This is probably not a real issue since the 4GL code must already have dealt with this already. The only implication here is that the built-in OO 4GL class hierarchy will need to have serialization added where it exists in the 4GL.

#2 - 05/29/2020 08:25 AM - Greg Shah

- Related to Feature #4373: finish core OO 4GL support added

#3 - 02/08/2022 03:28 PM - Greg Shah

Thoughts from Constantin:

- the runtime can't use 'implements Serializable' as progress.lang.object IS serializable. In OE one needs to mark each and every class as 'serializable'. So we will need some kind of annotation to mark serializable classes.
- we will need to go through all skeleton classes and manually mark them (plus all their transient fields, although I don't know if there is this concept in OE, beside SERIALIZE-HIDDEN at temp-tables)
- some types will need special consideration, as they will need to be recreated, like memptr and object (this needs to be registered with ObjectOps)
- temp-tables - do these get serialized? Are there types which don't get serialized? Do we call getter/setter in property serialization?
- we should build a generic serialization technique to not rely on serializing each and every field, as with read/writeExternal
- types of serialization: JSON, file, etc. What compatibility do we provide?

It seems easy on a first glance, but the deeper I dig, it gets more complex. And here it shows that OE can have custom serializers:
https://programinprogress.com/object-serialization-in-progress-openedge/

#4 - 02/09/2022 01:39 PM - Constantin Asofiei

I assume the OE serializer is walking the object graph somehow. But what happens if the same legacy OO instance is referenced in a certain field, in this object graph?

#5 - 05/16/2022 01:34 PM - Constantin Asofiei

There are SERIALIZABLE and NON-SERIALIZABLE options at the property and variable definition.

In 6129a/13857 I've added LegacyResource annotation for the legacy classes. I have not touched the property/var options.

#6 - 06/01/2022 07:28 PM - Greg Shah

Marian: We have a new customer project that needs this support urgently. Implementing this properly will require testcases. How quickly can your team write testcases?

#7 - 06/02/2022 04:31 AM - Marian Edu

Greg Shah wrote:

> Marian: We have a new customer project that needs this support urgently. Implementing this properly will require testcases. How quickly can your team write testcases?

We just need to setup an application server and start writing some testcases, just to be sure this is only for 4GL client -> AppSrv, has nothing to do with rest/soap tests that we've done before. It should probably be a matter of days, what is the OE version that we are testing against?

#8 - 06/02/2022 04:54 AM - Constantin Asofiei

Marian Edu wrote:

> just to be sure this is only for 4GL client -> AppSrv

Yes, just 4GL client -> AppSrv at this time, as what I need is the transport of an object to the appserver and back.

Please start with something simple, and in the end include all datatypes (including memptr/handle - how is handle serialized?), dataset, temp-table (what about PLO table fields? handle fields?), arrays, INPUT, OUTPUT, INPUT-OUTPUT modes, etc.

More complex tests should include nested objects (i.e. a class field or temp-table field is a object), circular references - what if the same 'this' reference is kept in another field somewhere?

A separate test suite should be for p2j.oo builtin classes, and exceptions (AppError, SysError).

#9 - 06/02/2022 05:13 AM - Constantin Asofiei

Marian Edu wrote:

> what is the OE version that we are testing against?

11.7.12

#10 - 06/02/2022 05:18 AM - Constantin Asofiei

Marian, the priority is a simple test with this:

- temp-table with data-types, including object fields, avoid handle type.
- class instance fields of different data-types, including object
- input, output, input-output modes
- object arrays in a parameter or temp-table field

Please provide this first, and expand after this to a more complete test suite.

#11 - 06/02/2022 05:22 AM - Constantin Asofiei

Constantin Asofiei wrote:

- object arrays in a parameter or temp-table field

Here I mean object arrays at the procedure parameter, temp-table field or class instance field.

#12 - 06/02/2022 05:50 AM - Marian Edu

Constantin Asofiei wrote:

> Marian Edu wrote:
>
> > what is the OE version that we are testing against?
>
> 11.7.12

That might be a problem for us, we have 12.2 installed and that doesn't include the 'classic appsrv' anymore... I'll check if we still have access to 11.7.

#13 - 11/15/2022 06:42 AM - Vladimir Tsichevski

- Related to Feature #6237: OEUnit support added

#14 - 11/15/2022 06:45 AM - Vladimir Tsichevski

We also need to provide Java serialization support for OO exceptions in order to implement unit-testing engine. See discussion around #6237-197.

#15 - 11/15/2022 07:32 AM - Greg Shah

Marian's team is (or will be) working on testcases.

However, I think you need the support now so you should look to provide some minimum level of support for the exception classes. Some of that can be done with manual edits to the built-in classes. Some initial provisional conversion support will also likely be needed so that application exceptions can be used in unit tests. Ask questions here and we will help you.

#16 - 11/15/2022 07:55 AM - Marian Edu

Greg Shah wrote:

> Marian's team is (or will be) working on testcases.
>
> However, I think you need the support now so you should look to provide some minimum level of support for the exception classes. Some of that can be done with manual edits to the built-in classes. Some initial provisional conversion support will also likely be needed so that application exceptions can be used in unit tests. Ask questions here and we will help you.

This is how it started in 4GL as well, when an error was thrown on the server side that was serialised as an error object on the client side (at some point)... I mean the exact error object instance not just an SysError.

#17 - 11/15/2022 07:57 AM - Vladimir Tsichevski

Greg Shah wrote:

> Marian's team is (or will be) working on testcases.
>
> However, I think you need the support now so you should look to provide some minimum level of support for the exception classes. Some of that can be done with manual edits to the built-in classes. Some initial provisional conversion support will also likely be needed so that application exceptions can be used in unit tests. Ask questions here and we will help you.

I am working on it now, and I'd like to commit the resulting change, so we will not do the same thing in parallel.

#18 - 11/15/2022 07:59 AM - Constantin Asofiei

Vladimir, what branch are you working on? There is some early support for OO serialization in 6129b, we need to work on top of that. If needed, the changes can be brought to 3812c...

#19 - 11/15/2022 08:27 AM - Vladimir Tsichevski

Constantin Asofiei wrote:

> Vladimir, what branch are you working on? There is some early support for OO serialization in 6129b, we need to work on top of that.

No special branch was created, I post the changes to #6237.

> If needed, the changes can be brought to 3812c...

It would be great!

#20 - 11/15/2022 09:01 AM - Vladimir Tsichevski

Constantin Asofiei wrote:

> Vladimir, what branch are you working on? There is some early support for OO serialization in 6129b, we need to work on top of that.

This branch is well behind the latest 3821c. Is it possible to update/rebase it?

#21 - 11/15/2022 09:48 AM - Greg Shah

- % Done changed from 0 to 10

#22 - 11/15/2022 11:11 AM - Vladimir Tsichevski

- % Done changed from 10 to 0

I am working on serialization of OO error classes. One of them is SoapFaultError, which has a field of the handle type. I wonder if this can (and should) be serialized.

#23 - 11/15/2022 11:11 AM - Vladimir Tsichevski

- % Done changed from 0 to 10

#24 - 11/15/2022 11:11 AM - Vladimir Tsichevski

- Status changed from New to WIP

#25 - 11/15/2022 11:21 AM - Vladimir Tsichevski

I wonder if there is a way to test OO serialization from 4gl, so I could write unit test?

#26 - 11/15/2022 11:24 AM - Vladimir Tsichevski

Greg Shah wrote:

> - enums are implicitly serializable (no syntax, it just is implemented)

The LegacyEnum Java class is not serializable. Or, do you mean something else?

#27 - 11/15/2022 11:33 AM - Greg Shah

Vladimir Tsichevski wrote:

> Greg Shah wrote:
>
> > - enums are implicitly serializable (no syntax, it just is implemented)

The LegacyEnum Java class is not serializable. Or, do you mean something else?

I mean that in the 4GL, their enums are serializable without any additional 4GL syntax (there is no use of the SERIALIZABLE keyword but the enums are still serializable). We need to add this to our enum implementation.

#28 - 11/15/2022 01:20 PM - Constantin Asofiei

Vladimir, I'll try to bring the changes from 6129b to 3821c tomorrow. The files in question are, but some of them contain other changes:

```
added:
  src/com/goldencode/p2j/util/LegacySerializable.java
  src/com/goldencode/p2j/util/LegacyObject.java
modified:
  rules/convert/base_structure.xml
  rules/convert/control_flow.rules
  src/com/goldencode/p2j/persist/TableWrapper.java
  src/com/goldencode/p2j/util/Agent.java
  src/com/goldencode/p2j/util/AppServerHelper.java
  src/com/goldencode/p2j/util/ObjectOps.java
  src/com/goldencode/p2j/util/object.java
```

#29 - 11/15/2022 02:53 PM - Vladimir Tsichevski

- File 4658.diff added

I've added Java serialization to the following classes (see 4658.diff):

```
src/com/goldencode/p2j/oo/core/AssertionFailedError.java
src/com/goldencode/p2j/oo/core/system/ApplicationError.java
src/com/goldencode/p2j/oo/dataadmin/error/DataAdminError.java
src/com/goldencode/p2j/oo/json/JsonParserError.java
src/com/goldencode/p2j/oo/lang/AppError.java
src/com/goldencode/p2j/oo/lang/LegacyEnum.java
src/com/goldencode/p2j/oo/lang/ProError.java
src/com/goldencode/p2j/oo/lang/SoapFaultError.java
src/com/goldencode/p2j/oo/net/http/HttpRequestError.java
src/com/goldencode/p2j/oo/web/SendExceptionError.java
src/com/goldencode/p2j/util/object.java
```

Please, review.

#30 - 11/16/2022 01:27 PM - Vladimir Tsichevski

I think, I found an error in skeleton: class DataAdminError has wrong class names in constructors: OperationError, must be DataAdminError:

```
class OpenEdge.DataAdmin.Error.DataAdminError
inherits Progress.Lang.AppError:
   def public property HTTPErrorNum as int get. set.

   def public property InnerError as Progress.Lang.Error get. set.

   constructor public OperationError(input v as char).
   end.

   constructor public OperationError(input v as char, err as Progress.Lang.Error).
   end.

end.
```

I think, it is save to fix it and commit the fix to skeleton?

#31 - 11/16/2022 01:30 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> I think, I found an error in skeleton: class DataAdminError has wrong class names in constructors: OperationError, must be DataAdminError:

BTW, the conversion complains a little while it process the file, but in the end generates wrong (but compilable) Java code.

#32 - 11/17/2022 07:21 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> I think, I found an error in skeleton: class DataAdminError has wrong class names in constructors: OperationError, must be DataAdminError:
>
> [...]
>
> I think, it is save to fix it and commit the fix to skeleton?

I have no commit access to the repo, can anyone help me committing the following change, please?

```
=== modified file 'oo4gl/OpenEdge/DataAdmin/Error/DataAdminError.cls'
--- oo4gl/OpenEdge/DataAdmin/Error/DataAdminError.cls    2019-05-07 15:57:06 +0000
+++ oo4gl/OpenEdge/DataAdmin/Error/DataAdminError.cls    2022-11-16 18:27:53 +0000
@@ -4,10 +4,10 @@

     def public property InnerError as Progress.Lang.Error get. set.

-    constructor public OperationError(input v as char).
+    constructor public DataAdminError(input v as char).
     end.

-    constructor public OperationError(input v as char, err as Progress.Lang.Error).
+    constructor public DataAdminError(input v as char, err as Progress.Lang.Error).
     end.

 end.
```

#33 - 11/17/2022 09:03 AM - Greg Shah

Are you using ~/secure/code/p2j_repo/skeleton? You should have full commit access to that repo. It is OK to commit your change.

#34 - 11/17/2022 11:27 AM - Vladimir Tsichevski

Greg Shah wrote:

> Are you using ~/secure/code/p2j_repo/skeleton? You should have full commit access to that repo. It is OK to commit your change.

Done, rev. 106.

Previously, I used the repo xfer.goldencode.com/opt/fwd/skeleton/, and when I tried to commit, I was asked for my password at xfer first, then for the password of Constantin (???).

#35 - 11/17/2022 11:29 AM - Greg Shah

If a repo exists on both devsrv01 and on xfer, then the devsrv01 version is authoritative and the xfer version is a copy that is sync'd when needed. FWD, the Hotel projects, the skeletons all fall into this category.

If a repo only exists on either devsrv01 or xfer but not both, then that repo is by definition authoritative. The testcases project is in this category.

#36 - 11/17/2022 02:19 PM - Vladimir Tsichevski

Greg Shah wrote:

> If a repo exists on both devsrv01 and on xfer, then the devsrv01 version is authoritative and the xfer version is a copy that is sync'd when needed. FWD, the Hotel projects, the skeletons all fall into this category.
>
> If a repo only exists on either devsrv01 or xfer but not both, then that repo is by definition authoritative. The testcases project is in this category.

Thank you, Greg.

#38 - 10/09/2023 02:21 PM - Greg Shah

> And here it shows that OE can have custom serializers: https://programinprogress.com/object-serialization-in-progress-openedge/

This site doesn't exist now. Do you recall what technique was used? I assume you mean they have something like Externalizable. My review of the 4GL docs didn't highlight any features that provide this.

#39 - 10/09/2023 02:46 PM - Constantin Asofiei

Greg Shah wrote:

> > And here it shows that OE can have custom serializers: https://programinprogress.com/object-serialization-in-progress-openedge/
>
> This site doesn't exist now. Do you recall what technique was used? I assume you mean they have something like Externalizable. My review of the 4GL docs didn't highlight any features that provide this.

I don't recall what that site had, and wayback machine doesn't have it. But I'm sure it was not about Externalizable, maybe it was some example using reflection to walk the object graph.

#40 - 10/09/2023 03:54 PM - Greg Shah

> And here it shows that OE can have custom serializers: https://programinprogress.com/object-serialization-in-progress-openedge/

> This site doesn't exist now. Do you recall what technique was used? I assume you mean they have something like Externalizable. My review of the 4GL docs didn't highlight any features that provide this.

> I don't recall what that site had, and wayback machine doesn't have it. But I'm sure it was not about Externalizable, maybe it was some example using reflection to walk the object graph.

We are implementing reflection separately. For this serialization task, I only want to consider features in which the runtime implements serialization or provides some mechanism for the programmer to get a call back that implements serialization.

Marian: Do you know of any such feature (like Java's Externalizable) in the 4GL?

#41 - 10/10/2023 02:13 AM - Marian Edu

Greg Shah wrote:

> Marian: Do you know of any such feature (like Java's Externalizable) in the 4GL?

The keyword is SERIALIZABLE and can be used on class objects and their properties (non-static) including temp-table and datasets. Initially there were no serializers provided as the use case was to only pass objects between client and application server (first requirement was to get the error objects across the wire). Since then there are now two serializers provided in Progress.IO package - binary and json, most probably the binary one is what they are using internally to pass objects between client/appsrv but that's a black box.

More info about this here [[https://docs.progress.com/bundle/openedge-oo-abl-develop-applications-117/page/Serializing-an-instance-to-file.html]] and about passing objects between client and appsrv -[[https://docs.progress.com/bundle/openedge-oo-abl-develop-applications-117/page/Passing-object-reference-parameters.html]].

I would say some form of reflection needs to be used here - aka, the object has no control of how it's data is being serialised, other than defining what is/isn't serialisable.

#42 - 10/10/2023 08:31 AM - Greg Shah

> the object has no control of how it's data is being serialised, other than defining what is/isn't serialisable

Perfect! That reduces our scope since there is no custom serialization logic/callback.

We will need testcases to be quite complete. What is the status of the tests?

Files

4658.diff	41.5 KB	11/15/2022	Vladimir Tsichevski