

Base Language - Feature #4965

Parameter annotation - qualified

10/15/2020 02:18 AM - Marian Edu

Status:	Test	Start date:	
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
billable:	No	vendor_id:	GCD
Description			
Related issues:			
Related to Base Language - Feature #5146: record additional legacy class and ...			New

History

#1 - 10/15/2020 06:22 AM - Greg Shah

- Description updated

From Marian:

When a class is instantiated a 'constructor' method that matches the arguments is searched and then depending on the parameter mode there are some validations on the data type. If the parameter is object then the qualified annotation is used to find the actual class/interface, would it make sense to default to P.L.O if no annotation is present?

The issue we had is we have a variable defined as P.L.O that gets converted to `_BaseObject_` (interface), the parameter qualified annotation is also P.L.O only that result in `BaseObject` (base class) and the constructor is ignored because the interface is not assignable to the base class :(

Maybe the conversion can be changed to use `BaseObject` for P.L.O variables, meanwhile as a work around I'm changing the legacy class from `BaseObject` to `_BaseObject_` in this case - `ControlFlowOps`.

#2 - 10/21/2020 09:46 AM - Greg Shah

- Assignee set to Constantin Asofiei

#3 - 10/21/2020 09:49 AM - Constantin Asofiei

When a class is instantiated a 'constructor' method that matches the arguments is searched and then depending on the parameter mode there are some validations on the data type. If the parameter is object then the qualified annotation is used to find the actual class/interface, would it make sense to default to P.L.O if no annotation is present?

Do you have an example for this? The 'qualified' annotation must emit always.

#4 - 10/22/2020 02:06 AM - Marian Edu

Constantin Asofiei wrote:

Do you have an example for this?

The only one we stumble upon is for MapEntry that takes the "entry" object as input in constructor and that one is P.L.O, test is in oo/openedge/core/collections/map_entry/test_constructor.p.

There the method is qualified so it's not a problem, the problem is that _BaseObject_ (input) is not assignable from BaseObject (resolved qualified name).

The 'qualified' annotation must emit always.

This is unless one forgot to put the annotation in, just saying using P.L.O as annotation doesn't really make any difference cause it can be anything so using it as default seems like a harmless choice to me :)

#5 - 10/23/2020 09:51 AM - Constantin Asofiei

Marian Edu wrote:

There the method is qualified so it's not a problem, the problem is that _BaseObject_ (input) is not assignable from BaseObject (resolved qualified name).

Who/where is BaseObject resolved? Is this a runtime problem?

The 'qualified' annotation must emit always.

This is unless one forgot to put the annotation in, just saying using P.L.O as annotation doesn't really make any difference cause it can be anything so using it as default seems like a harmless choice to me :)

I'm OK with this as long as the default is not hard-coded at LegacyParameter or LegacySignature, but where the qualified annotation is used. There is more than one place which will need attention - look at where LegacySignature.qualified and LegacyParameter.qualified are read. Also, an warning should be logged that we default to progress.lang.object.

#6 - 02/19/2021 09:47 AM - Constantin Asofiei

Greg, you mentioned at some point that the method's return type is not in LegacySignature - where was this logged?

The missing part at making the 'qualified' default to PLO is that I don't have the return type at LegacySignature.

#7 - 02/19/2021 09:58 AM - Greg Shah

- Related to Feature #5146: record additional legacy class and method data in annotations added

#8 - 02/19/2021 09:59 AM - Greg Shah

You mentioned at some point that the method's return type is not in LegacySignature - where was this logged?

See [#5146](#). It really is surprising that we forgot that part. :O

#9 - 02/19/2021 11:05 AM - Constantin Asofiei

It was easy to write a simple tool to check the p2j.oo methods which have LegacySignature. The list of problems is this:

```
'object' parameter 0 has no qualified annotation for method: protected void com.goldencode.p2j.oo.core.collect
ions.LegacyMapEntry.setOwningMap(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.core.collection
s.LegacyMapEntry.setValue(com.goldencode.p2j.util.object)
Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.uti
l.object com.goldencode.p2j.oo.lang.LegacyClass.getLegacyClass(com.goldencode.p2j.util.character)
Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.uti
l.object com.goldencode.p2j.oo.lang.LegacyClass.getLegacyClass(java.lang.String)
Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.objec
t com.goldencode.p2j.oo.lang.LegacyClass.new_(com.goldencode.p2j.util.object)
Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.objec
t com.goldencode.p2j.oo.lang.LegacyClass.new_()
Parameter count and signature parameters count do not match: public void com.goldencode.p2j.oo.net.http.HttpCl
ient.setLogger(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.Conten
tTypeHeader.setLogger(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.lib.so
ckets.LegacySocketLibrary.setLogger(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: public void com.goldencode.p2j.oo.net.http.Cookie
Jar.setLogger(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: protected void com.goldencode.p2j.oo.core.collect
ions.LegacyIterator.setOwnerCollection(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.Authen
ticationRequestEventArgs.setCredentials(com.goldencode.p2j.util.object)
Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.objec
t com.goldencode.p2j.oo.core.system.ApplicationError.clone()
Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.uti
l.object com.goldencode.p2j.oo.net.http.filter.writer.StatusCodeWriterBuilder.build(com.goldencode.p2j.util.ob
ject,com.goldencode.p2j.util.object)
Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.uti
l.object com.goldencode.p2j.oo.lang.LegacyEnum.toObject(com.goldencode.p2j.util.character,com.goldencode.p2j.u
til.character)
Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.uti
l.object com.goldencode.p2j.oo.lang.LegacyEnum.toObject(com.goldencode.p2j.util.character,com.goldencode.p2j.u
til.int64)
Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.objec
t com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.writeTo(com.goldencode.p2j.util.object)
Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.uti
l.object com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.build(com.goldencode.p2j.util.objec
```

t,com.goldencode.p2j.util.object)

Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.build(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)

Return type is 'object' but has no qualified annotation at its signature: protected com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.getWriter(com.goldencode.p2j.util.character)

Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.option(com.goldencode.p2j.util.character,com.goldencode.p2j.util.decimal)

Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.option(com.goldencode.p2j.util.character,com.goldencode.p2j.util.logical)

Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.option(com.goldencode.p2j.util.character,com.goldencode.p2j.util.character)

Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.option(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)

Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.MessageWriterBuilder.option(com.goldencode.p2j.util.character,com.goldencode.p2j.util.int64)

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.HttpResponseDecorator.setEntity(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.dataadmin.error.DataAdminError.setInnerError(com.goldencode.p2j.util.object)

Parameter count and signature parameters count do not match: public void com.goldencode.p2j.oo.web.WebHandler.setLogger(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: protected void com.goldencode.p2j.oo.core.collections.EntrySetIterator.setOwningMap(com.goldencode.p2j.util.object)

Parameter count and signature parameters count do not match: public abstract void com.goldencode.p2j.oo.net.http.ISupportCookies.setCookieJar(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.serverconnection.SocketReadEventArgs.setData(com.goldencode.p2j.util.object)

Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.BodyWriterBuilder.build(com.goldencode.p2j.util.character)

Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.BodyWriterBuilder.build(com.goldencode.p2j.util.object)

Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.BodyWriterBuilder.build_1(com.goldencode.p2j.util.object)

Parameter count and signature parameters count do not match: public abstract void com.goldencode.p2j.oo.logging.ISupportLogging.setLogger(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.HttpMessage.setEntity(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: public abstract void com.goldencode.p2j.oo.core.collections.ImapEntry.setValue(com.goldencode.p2j.util.object)

Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.object com.goldencode.p2j.oo.lang.BaseObject.getLegacyClass()

Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.object com.goldencode.p2j.oo.lang.BaseObject.clone()

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.serverconnection.ClientSocket.setLogger(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.serverconnection.ClientSocket.setConnectionParameters(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.serverconnection.ClientSocket.setServer(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.serverconnection.ClientSocket.writeData(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: protected void com.goldencode.p2j.oo.core.collections.ArrayIterator.setIteratedArray(com.goldencode.p2j.util.object)

'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.web.WebResponse.__web_WebResponse_constructor__(com.goldencode.p2j.util.object)

Parameter count and signature parameters count do not match: public void com.goldencode.p2j.oo.net.http.ProxyHttpClient.setProxyUri(com.goldencode.p2j.util.object)

Return type is 'object' but has no qualified annotation at its signature: protected com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.DefaultHeaderBuilder.newHeader()

Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.EntityWriterBuilder.build(com.goldencode.p2j.util.character)

Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.filter.writer.EntityWriterBuilder.build(com.goldencode.p2j.util.object)

Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.uti

```

1.object com.goldencode.p2j.oo.net.http.filter.writer.EntityWriterBuilder.build_1(com.goldencode.p2j.util.objec
ct)
'object' parameter 0 has no qualified annotation for method: public static void com.goldencode.p2j.oo.net.http
.HttpHeader.setNullHeader(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.HttpRe
questDecorator.setEntity(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.HttpRe
questDecorator.setUri(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public abstract void com.goldencode.p2j.oo.net.ht
tp.IHttpRequest.setEntity(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: public void com.goldencode.p2j.oo.net.http.Statef
ulHttpClient.setCookieJar(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: public void com.goldencode.p2j.oo.net.http.ProxyH
ttpRequest.setProxyUri(com.goldencode.p2j.util.object)
Return type is 'object' but has no qualified annotation at its signature: public com.goldencode.p2j.util.objec
t com.goldencode.p2j.oo.core.collections.StringKeyedMap.get(com.goldencode.p2j.util.character)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.MessagePart
.setBody(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.MessagePart
.setHeaders(com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: public abstract void com.goldencode.p2j.oo.net.ht
tp.ISupportProxy.setProxyUri(com.goldencode.p2j.util.object)
'object' parameter 1 has no qualified annotation for method: public com.goldencode.p2j.util.object com.goldenc
ode.p2j.oo.net.http.RequestBuilder.addCallback(com.goldencode.p2j.util.object,com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: public static com.goldencode.p2j.util.object com.
goldencode.p2j.oo.net.http.RequestBuilder.put(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object
)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.Authen
ticatedRequest.setCredentials(com.goldencode.p2j.util.object)
Return type is 'object' but has no qualified annotation at its signature: public abstract com.goldencode.p2j.u
til.object com.goldencode.p2j.oo.core.collections.Iterable.iterator()
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.http.HttpRe
quest.setCookie(com.goldencode.p2j.util.object)
Return type is 'object' but has no qualified annotation at its signature: public static com.goldencode.p2j.uti
l.object com.goldencode.p2j.oo.net.http.filter.writer.SetHeaderMessageWriterBuilder.build(com.goldencode.p2j.u
til.object,com.goldencode.p2j.util.object)
'object' parameter 0 has no qualified annotation for method: protected void com.goldencode.p2j.oo.web.WebRespo
nseWriter.setResponse(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: public com.goldencode.p2j.util.object com.goldenc
ode.p2j.oo.web.RemoteWebRequest.setHeader(com.goldencode.p2j.util.character,com.goldencode.p2j.util.character)
'object' parameter 0 has no qualified annotation for method: public void com.goldencode.p2j.oo.net.serverconne
ction.ClientSocketConnectionParameters.setUri(com.goldencode.p2j.util.object)

```

Some other issue is that for a setter no 'qualified' annotation is emitted by the conversion rules, for its parameter. I haven't checked OO event methods.

At this time, I would prefer to not go down the rabbit hole of adding the return type for each OO method/function/getter, but instead add this tool and sanitize the p2j.oo code.

#10 - 02/19/2021 11:09 AM - Greg Shah

I already have code to calculate the return type using reflection and the existing annotations. I will check it in shortly. It is a poor substitute for the missing data because it is over complicated and will be more fragile than using an annotation. I do prefer to fix this properly. Before we decide one way or the other, please review my code to see what you think.

#11 - 02/19/2021 11:38 AM - Greg Shah

I already have code to calculate the return type using reflection and the existing annotations.

3821c rev 12032 includes this code (see `SupportLevelDocumentationGenerator.reverseType()`).

#12 - 02/19/2021 03:11 PM - Constantin Asofiei

Greg Shah wrote:

I already have code to calculate the return type using reflection and the existing annotations.

3821c rev 12032 includes this code (see `SupportLevelDocumentationGenerator.reverseType()`).

Yes, that's helpful.

I agree that the return type should be added. But the approach to 'default to PLO' if the qualified annotation is missing can lead to weird problems, as this is used to match and validate arguments (is important especially for dynamic calls). So I'm thinking of a way to fix these and 'inject' the qualified annotation for the parameter and method's return type automatically in the source file of p2j.oo classes. And use the same approach to inject the return type where is possible.

I'm not sure about the getters and setters - these aren't OO methods, we just emulate them as such. But I'll see if these can be safely added there, too (I mean, it doesn't affect the runtime or if it does, it can be fixed easily).

#13 - 02/21/2021 09:27 AM - Constantin Asofiei

I've built a tool which automatically edits the source code for the `com.goldencode.p2j.oo` classes and injects qualified annotation (for object method or parameter, if is missing) and returns annotation (for all non-void methods, if is missing). It also reports missing extent annotation for extent

method/parameter and parameter count mismatch between LegacySignature and Java method signature.

After running it, what's left are these which will need to be fixed manually:

```
Method return type is character[] but has no 'extent' annotation: com.goldencode.p2j.oo.core.util.ConfigBuilder.getOptionStringArrayValue(com.goldencode.p2j.util.character)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.net.http.CookieJar.setLogger(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.web.WebHandler.setLogger(com.goldencode.p2j.util.object)
Method return type is character[] but has no 'extent' annotation: com.goldencode.p2j.oo.core.util.BuilderRegistry.getKeys()
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.logging.ISupportLogging.setLogger(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.net.http.ProxyHttpClient.setProxyUri(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.net.http.RequestBuilder.put(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.net.http.StatefulHttpClient.setCookieJar(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.net.http.ProxyHttpRequest.setProxyUri(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.web.RemoteWebRequest.setHeader(com.goldencode.p2j.util.character,com.goldencode.p2j.util.character)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.net.http.ISupportCookies.setCookieJar(com.goldencode.p2j.util.object)
Parameter count and signature parameters count do not match: com.goldencode.p2j.oo.net.http.HttpClient.setLogger(com.goldencode.p2j.util.object)
```

This tool can be extended to sanitize the LegacySignature against the Java method signature (if they match or not).

I've also added the 'default to PLO' changes, I'll commit these and this tool (p2j.uast.LegacyBuiltInClassSanitizer) first.

Second, I'll make the manual changes in the list above.

Third, I'll run the tool, re-check the changes in p2j.oo package and commit these.

One thing to note: currently I'm not treating the java.lang.Object LegacyClass.invoke case, which is polymorphic. Or ParameterList.setParameter(..., Object val), which has a polymorphic argument. How should these be encoded?

#14 - 02/21/2021 11:27 AM - Constantin Asofiei

- Status changed from New to WIP

- % Done changed from 0 to 100

All changes are in 3821c revs 12042 to 12044.

I did not touch the poly methods/parameters:

```
Method return type is Object but has no 'returns' annotation: com.goldencode.p2j.oo.lang.LegacyClass.getProper
tyValue(com.goldencode.p2j.util.character)
Method return type is Object but has no 'returns' annotation: com.goldencode.p2j.oo.lang.LegacyClass.getProper
tyValue(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
Method return type is Object but has no 'returns' annotation: com.goldencode.p2j.oo.lang.LegacyClass.getProper
tyValue(com.goldencode.p2j.util.object,com.goldencode.p2j.util.character)
Method return type is Object but has no 'returns' annotation: com.goldencode.p2j.oo.lang.LegacyClass.getProper
tyValue(com.goldencode.p2j.util.object,com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
Method return type is Object but has no 'returns' annotation: com.goldencode.p2j.oo.lang.LegacyClass.invoke(co
m.goldencode.p2j.util.character)
Method return type is Object but has no 'returns' annotation: com.goldencode.p2j.oo.lang.LegacyClass.invoke(co
m.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
Method return type is Object but has no 'returns' annotation: com.goldencode.p2j.oo.lang.LegacyClass.invoke(co
m.goldencode.p2j.util.object,com.goldencode.p2j.util.character)
Method return type is Object but has no 'returns' annotation: com.goldencode.p2j.oo.lang.LegacyClass.invoke(co
m.goldencode.p2j.util.object,com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
```

#15 - 02/21/2021 08:14 PM - Greg Shah

Perhaps the default value for LegacySignature.returns() should be "VOID".

As for the poly cases, I think they should be marked "POLYMORPHIC".

#16 - 02/22/2021 03:08 AM - Marian Edu

Constantin Asofiei wrote:

I agree that the return type should be added. But the approach to 'default to PLO' if the qualified annotation is missing can lead to weird problems, as this is used to match and validate arguments (is important especially for dynamic calls).

Constantin, just to clarify, the default to P.L.O is not for the (newly added) return type but for the 'qualified' when the return type is set to `object`. Imho it serve no purpose to qualify to the default base object hence there is no real qualification there.

@Greg, since there seems to be some automatic changes applied to 'oo' packages should we make a new branch to incorporate those on our side? I'm set to resume work on this next week on Monday.

#17 - 02/22/2021 04:09 AM - Constantin Asofiei

Marian Edu wrote:

Constantin, just to clarify, the default to P.L.O is not for the (newly added) return type but for the 'qualified' when the return type is set to `object`. Imho it serve no purpose to qualify to the default base object hence there is no real qualification there.

For reporting and runtime consistency, the sanitize tool automatically injects the qualified annotation as needed (in case of object parameters or method return type). The conversion rules were also changed to emit this for converted 4GL code, as needed.

I've added the 'default to PLO' for both return type and parameters (if they are object), but this should be considered 'for development purpose only'. As new classes/methods are added, the sanitize tool should be used to check and fix any problems, before considering the code 'production ready'.

#18 - 02/22/2021 06:55 AM - Greg Shah

since there seems to be some automatic changes applied to 'oo' packages should we make a new branch to incorporate those on our side?

Yes, agreed. I've pushed 3821c revision 12046 to xfer. Do you want to branch from that, apply your changes and provide a bsr send so we can merge your changes into 3821c?

Or do you just want a branch based on the latest 3821c? For example, this might be useful if your changes are not ready for merge. In this case we can replace 4384h or add an 4384i branch.

#19 - 02/22/2021 07:18 AM - Marian Edu

Greg Shah wrote:

since there seems to be some automatic changes applied to 'oo' packages should we make a new branch to incorporate those on our side?

Yes, agreed. I've pushed 3821c revision 12046 to xfer. Do you want to branch from that, apply your changes and provide a bsr send so we can merge your changes into 3821c?

Or do you just want a branch based on the latest 3821c? For example, this might be useful if your changes are not ready for merge. In this case we can replace 4384h or add an 4384i branch.

I have no uncommitted changes, 4384h is current so we can go either way, your call since I don't have that much experience with bazaar and it looks like those long running operations are timing out on our side for whatever reason :(

#20 - 02/22/2021 07:20 AM - Greg Shah

I've pushed 3821c revision 12046 to xfer. Do you want to branch from that, apply your changes and provide a bsr send so we can merge your changes into 3821c?

OK, let's go this way. Can you branch from the latest 3821c?

#21 - 02/23/2021 02:46 AM - Marian Edu

- File 3821c.4384h.patch added

Greg Shah wrote:

I've pushed 3821c revision 12046 to xfer. Do you want to branch from that, apply your changes and provide a bsr send so we can merge your changes into 3821c?

OK, let's go this way. Can you branch from the latest 3821c?

Attached the patch, more work to do the merge than manually adding the darn `return` annotations or we should have run that after an intermediate merge... anyway, hope my eyes didn't miss anything and the merge is correct :(

#22 - 02/23/2021 12:04 PM - Constantin Asofiei

Marian, 4384h on xfer is still on rev 12004 - although I think is a little old, is still good, because I could use it as a reference for your patch.

You can checkout 4384h from xfer in a separate folder and do bsr diff -r 11992 > a.txt then meld a.txt with the 3821c.4384h patch, if you want to double-check. The patch looks OK to me. I've merged to 3821c as rev 12050.

Greg: I think we need a 4384i branch?

I've looked over the changes and in FileInputStream constructor and MessageWriter.writeFileStream you are using FileSystemOps.initFileInfo(fileName);.

I'm not sure this is OK. FileSystemOps.initFileInfo is emitted by conversion rules for FILE-INFO:FILE-NAME, and calling this API will overwrite any existing FILE-INFO state set before the method call.

#23 - 02/23/2021 12:12 PM - Marian Edu

Constantin Asofiei wrote:

Marian, 4384h on xfer is still on rev 12004 - although I think is a little old, is still good, because I could use it as a reference for your patch.

I don't have the windows laptop with me but nothing changed on our side the last two weeks soo yes, a little old maybe :)

You can checkout 4384h from xfer in a separate folder and do `bzr diff -r 11992 > a.txt` then meld a.txt with the 3821c.4384h patch, if you want to double-check. The patch looks OK to me. I've merged to 3821c as rev 12050.

Greg: I think we need a 4384i branch?

I've looked over the changes and in `FileInputStream` constructor and `MessageWriter.writeFileStream` you are using `FileSystemOps.initFileInfo(fileName);`.

I'm not sure this is OK. `FileSystemOps.initFileInfo` is emitted by conversion rules for `FILE-INFO:FILE-NAME`, and calling this API will overwrite any existing `FILE-INFO` state set before the method call.

Yeah, this is the only way we can get the file size in 4GL... maybe in FWD we can call something directly so we don't change the file-name, otherwise we could just save the value and reset it back but I'm willing too bet that 'reset' behaviour is present in 4GL implementation, I will check first just in case (maybe we need to keep it as-is to be compatible).

#24 - 02/23/2021 12:18 PM - Constantin Asofiei

Marian, there are some compile errors -

```
[exec] [ant:javac] p2j/src/com/goldencode/p2j/oo/net/http/filter/writer/MessageWriterBuilder.java:188: er
ror: REGISTRY has private access in EntityWriterRegistry
[exec] [ant:javac]         object<? extends BuilderRegistry> registry = EntityWriterRegistry.REGISTRY.get
();
[exec] [ant:javac]                                     ^
[exec] [ant:javac] p2j/src/com/goldencode/p2j/oo/net/http/filter/writer/RequestWriterBuilder.java:181: er
ror: REGISTRY has private access in EntityWriterRegistry
[exec] [ant:javac]         object<? extends BuilderRegistry> registry = EntityWriterRegistry.REGISTRY.ge
t();
```

Is this a typo? Shouldn't each class access its own `REGISTRY` member? I'll remove the `EntityWriterRegistry` and commit this to 3821c.

Also, I see lots of classes have null as the referent in i.e. its `BlockManager.function(null, ...)` - see `RequestWriterBuilder.getRegistry()`. And you've

removed the `ObjectOps.load(RequestWriterBuilder.class); call`. A first static access of this API, will not load the legacy class. You either leave the `ObjectOps.load(RequestWriterBuilder.class); call` or set the first argument (referent) to the class, like `function(RequestWriterBuilder.class, ...)`.

#25 - 02/23/2021 12:21 PM - Greg Shah

Greg: I think we need a 4384i branch?

Yes. Please make one as soon as you are happy with the 3821c starting point.

#26 - 02/24/2021 07:55 AM - Marian Edu

Constantin Asofiei wrote:

Marian, there are some compile errors -
[...]

Is this a typo? Shouldn't each class access its own REGISTRY member? I'll remove the `EntityWriterRegistry` and commit this to 3821c.

Also, I see lots of classes have null as the referent in i.e. its `BlockManager.function(null, ...)` - see `RequestWriterBuilder.getRegistry()`. And you've removed the `ObjectOps.load(RequestWriterBuilder.class); call`. A first static access of this API, will not load the legacy class. You either leave the `ObjectOps.load(RequestWriterBuilder.class); call` or set the first argument (referent) to the class, like `function(RequestWriterBuilder.class, ...)`.

This is what happens when doing changes during merge in a non-java project :(
How should we proceed were, a new merge or just wait for the new 4384 branch and fix whatever needs to be fixed there?

#27 - 02/24/2021 08:43 AM - Constantin Asofiei

Marian Edu wrote:

How should we proceed were, a new merge or just wait for the new 4384 branch and fix whatever needs to be fixed there?

There is the 4384i branch available on xfer - see [#4384-450](#)

#28 - 02/25/2021 04:32 AM - Greg Shah

- Status changed from WIP to Test

#29 - 03/01/2021 03:48 AM - Marian Edu

Constantin Asofiei wrote:

Also, I see lots of classes have null as the referent in i.e. its BlockManager.function(null, ...) - see RequestWriterBuilder.getRegistry().

Yes, this is often the case with the 'stubs' we've found in place and as a rule of thumb I've tried not to update code that 'seemed' to work so those were left as-is.

And you've removed the ObjectOps.load(RequestWriterBuilder.class); call. A first static access of this API, will not load the legacy class. You either leave the ObjectOps.load(RequestWriterBuilder.class); call or set the first argument (referent) to the class, like function(RequestWriterBuilder.class, ...).

This is part of the 'OO SDK' and as per your suggestion everything under com.goldencode.p2j.oo is registered in ObjectOps static ctor, should this cover it?

One thing that I'm not sure if was introduced by latest changes in our new #4384i branch is that when we send the result of a method directly to another method that input seems not to be valid (aka, not found in local objects map). If I'm using an intermediate variable then all works.

```
// this does not work
wObj.ref().setEntity(ByteBucket.instance());

// this works
object<? extends ByteBucket> bbObj = TypeFactory.object(ByteBucket.class);
bbObj.assign(ByteBucket.instance());
wObj.ref().setEntity(bbObj);
```

The static instance method of ByteBucket does use the regular ObjectOps.newInstance call to instantiate the object and there it's reference is clearly valid as it is 'initialized' but then somehow along the ObjectVar that is passed along becomes invalid... any idea what could be wrong here?

#30 - 03/01/2021 03:57 AM - Constantin Asofiei

Marian Edu wrote:

This is part of the 'OO SDK' and as per your suggestion everything under com.goldencode.p2j.oo is registered in ObjectOps static ctor, should this cover it?

Sorry, I forgot about that, thanks for the reminder. So there is no problem then.

One thing that I'm not sure if was introduced by latest changes in our new #4384i branch is that when we send the result of a method directly to another method that input seems not to be valid (aka, not found in local objects map). If I'm using an intermediate variable then all works.

I think this is most likely because the instance is left 'unreferenced' in ByteBucket.instance and then passed directly as a parameter to the setEntity call:

```
object<? extends ByteBucket> bb = ObjectOps.newInstance(ByteBucket.class);
bb.ref().initialize();
returnNormal(bb);
```

Each object var definition should be declared outside the top-level block (in the Java method, before the i.e. return function(...) via TypeFactory.object and a ObjectOps.register call for it needs to be in the Block.init method.

Please change the ByteBucket.instance code as above and try again with the wObj.ref().setEntity(ByteBucket.instance()); call. If is still not working, I'll take a deeper look.

#31 - 03/01/2021 05:17 AM - Marian Edu

Constantin Asofiei wrote:

Each object var definition should be declared outside the top-level block (in the Java method, before the i.e. return function(...) via TypeFactory.object and a ObjectOps.register call for it needs to be in the Block.init method.

While I do understand what you are saying I'm pretty sure this was working fine before, the ByteBucket.instance is used as default entity for many writers in .net package. It's the method return value, how could that be invalid and why would one need to register the return value? If really an ObjectVar is needed instead of the default object instance then maybe something can be done in returnNormal to fix that. But then again, something

changed in that area and I'm not sure our tests that were previously marked as 'passing' are still valid now :(

Please change the ByteBucket.instance code as above and try again with the wObj.ref().setEntity(ByteBucket.instance()); call. If is still not working, I'll take a deeper look.

Did that but it didn't help, apparently adding a initInput on setEntity method was in order too which makes it even more confusing as of why using an intermediate variable did work even without initInput?

#32 - 03/01/2021 05:28 AM - Constantin Asofiei

Please post a small test which shows this problem. The reference is decremented somewhere where it should not be.

#33 - 03/10/2021 05:51 AM - Marian Edu

Marian Edu wrote:

Constantin Asofiei wrote:

Each object var definition should be declared outside the top-level block (in the Java method, before the i.e. return function(...) via TypeFactory.object and a ObjectOps.register call for it needs to be in the Block.init method.

Ok, think I've found a way here... TypeFactory.object it's actually making an ObjectVar and later when assign is called the object is incremented. Returning directly an object seems not to be an option as that one is not 'valid' outside of the block, have to revisit methods that returns objects and see if we have any other situations like this :(

However, what I don't understand is why should the register of the variable be done in Init block? Seems to work just fine without it, and I don't see that being used in any methods apart from when an output parameter is used (not return) and sporadically in execute method when all object properties are registered - this is from conversion, only a few classes actually have this. Why would those object properties need to be registered in execute, those are clearly invalid at that time anyway?

#34 - 03/10/2021 06:23 AM - Constantin Asofiei

Marian, you mean a return new progress.lang.object() will work in 4GL but will not work in FWD? If yes, then this is a FWD bug and needs to be fixed.

#35 - 03/10/2021 07:44 AM - Marian Edu

Constantin Asofiei wrote:

Marian, you mean a return new progress.lang.object() will work in 4GL but will not work in FWD? If yes, then this is a FWD bug and needs to be fixed.

No, what I'm saying is when an intermediate local variable is used as return value the conversion puts the variable definition out of the function block and then it does add that (Init) block to register it. But, as far as I can see this is not done at all in FWD code. Unless the method has output parameters - where initOutput is used - or a few cases where execute does register all/some of the class properties in that Init block this is not used for any method that returns an object. So, right now I'm trying to figure this out as of why isn't that present more in the codebase - there must be more places where local variables are used as return value. And last, why ByteBucket.instance method now works even without that Init block where the ObjectOps.register(\$local_var) should be added?

Files

3821c.4384h.patch	160 KB	02/23/2021	Marian Edu
-------------------	--------	------------	------------