

Base Language - Feature #5146

record additional legacy class and method data in annotations

02/12/2021 10:40 AM - Greg Shah

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>billable:</b>	No	<b>vendor_id:</b>	GCD
<b>Description</b>			
<b>Related issues:</b>			
Related to Base Language - Feature #4373: finish core OO 4GL support			<b>New</b>
Related to Base Language - Feature #4965: Parameter annotation - qualified			<b>Test</b>

History

#1 - 02/12/2021 10:41 AM - Greg Shah

- We should be able to record (in an annotation) whether the converted class was a CLASS, INTERFACE or ENUM in the 4GL. I'm not sure if LegacyResource is the right place, but that should be recorded somewhere. It can be inferred by Java reflection but that is messy.
- The LegacySignature annotation is missing the method's return type. Recall that type is actually an InternalEntry.Type instance which defines what kind of internal entry is being implemented by this Java method (procedure, OO 4GL method, property getter...). Although we can record the extent and the OO qualified name, we can't actually record the original return type. Again, we can "reverse" the Java return type via reflection and make some assumptions, but this is messy.

Both cases seem like information we should record at conversion time. I'm writing code to dump all this right now, so it is top of mind for me. It is not urgent since I have written the messy reflection approach to figuring out each of the above items. But I don't want that to be our long term approach.

#2 - 02/12/2021 10:41 AM - Greg Shah

- Related to Feature #4374: parameter validation for Progress.Lang.Class:invoke and new added

#3 - 02/12/2021 10:41 AM - Greg Shah

- Related to deleted (Feature #4374: parameter validation for Progress.Lang.Class:invoke and new)

#4 - 02/12/2021 10:41 AM - Greg Shah

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

#5 - 02/19/2021 09:58 AM - Greg Shah

- Related to Feature #4965: Parameter annotation - qualified added

#6 - 02/19/2021 10:55 AM - Constantin Asofiei

Regarding the return type for function and OO methods - my only problem is that there are ~3000 hand-written classes in p2j.oo package which would need to be updated. There may be a way to automate this.

**#7 - 02/21/2021 11:28 AM - Constantin Asofiei**

3821c rev 12042 to 12044 fixes the 'extent', 'qualified' and adds 'returns' annotation to all legacy p2j.oo methods. Also, fixes conversion rules to emit these for all Java methods. What I did not address are legacy variable and property signature.

Also, there is a tool to sanitize automatically the method signature for p2j.oo classes - see [#4965-13](#) for more details.