User Interface - Bug #3592

shared frame interface needs to use the entire frame structure, not from each frame reference

05/28/2018 07:17 AM - Constantin Asofiei

Status:	New	Start date:		
Priority:	Normal	Due date:		
Assignee:		% Done:	0%	
Category:		Estimated time:	0.00 hour	
Target version:				
billable:	No	case_num:		
vendor_id:	GCD	version:		
Description				
Related issues:				
Related to Database -	Bug #4450: implement shared guery and s	shared browse	Closed	

History

#2 - 05/28/2018 07:22 AM - Constantin Asofiei

A frame definition may be split between multiple UI statements, like this:

1. master reference:

```
def new shared frame f1.
def var x as int.
def var y as int.
form x y with frame f1.
```

2. child reference:

```
def shared frame f1.
def var x as int.
def var y as int.
form x with frame f1.
form y with frame f1.
```

We need to capture all frame references which affect its structure (similar to how we build the frame in frame_generator.xml), and use FrameAstKey with this definition.

To bypass this issue, I've changed frame_generator_pre.xml to avoid processForm if fAlloc already has fr_interface annotation, but this is just a patch and it works only if first frame reference in each program have the same structure.

05/21/2024 1/3

#3 - 05/28/2018 07:28 AM - Greg Shah

Questions:

- 1. Are these in separate programs?
- 2. What is the problem that this causes for the frame? Does it have 4 widgets (prog1_x, prog1_y, prog2_x, prog2_y) in the 4GL but just 2 in FWD?

#4 - 05/28/2018 07:32 AM - Constantin Asofiei

Greg Shah wrote:

Questions:

1. Are these in separate programs?

Yes.

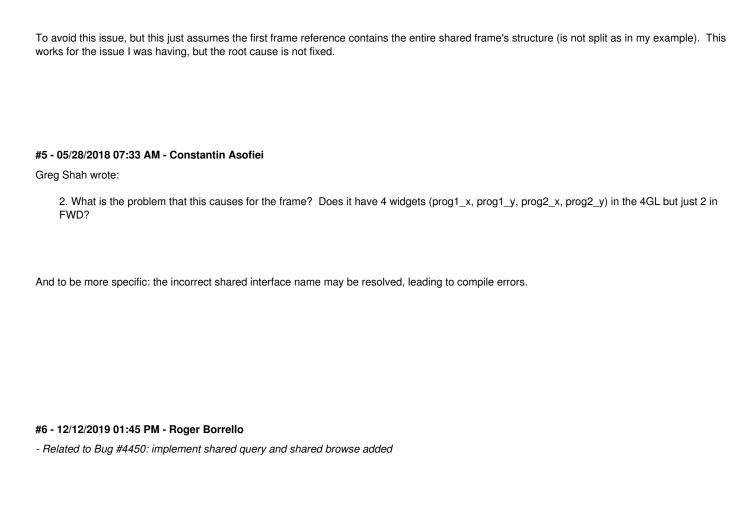
2. What is the problem that this causes for the frame? Does it have 4 widgets (prog1_x, prog1_y, prog2_x, prog2_y) in the 4GL but just 2 in FWD?

The problem is that frame_generator.xml processes each individual i.e FORM statement as having an unique frame structure, from which a shared frame interface must be built - this is incorrect, as the frame structure must be determined from all these i.e. FORM statements.

I've used this patch:

```
### Eclipse Workspace Patch 1.0
#P p2i
Index: rules/annotations/frame_generator_pre.xml
_____
--- rules/annotations/frame_generator_pre.xml
                                             (revision 1592)
+++ rules/annotations/frame_generator_pre.xml (working copy)
@@ -3,11 +3,17 @@
** Module : frame_generator_pre.xml
** Abstract : Walks procedures and looks for common interfaces for each FRAME.
-** Copyright (c) 2016-2017, Golden Code Development Corporation.
+** Copyright (c) 2016-2018, Golden Code Development Corporation.
** _#_ _I_ __Date_
                                                   Description
^{\star\star} 001 OM \, 20161107 Walks procedures and looks for common interfaces for each SHARED FRAME.
** 002 OM 20170118 Avoid interface frame name collision on Windows filesystem.
+** 003 CA 20180528 This code should capture the frame definition from all frame-structure related
+**
                    statements (FORM, UPDATE, DISPLAY, etc), same as frame generator.xml does;
+**
                    this is because a frame definition may be split in multiple statements.
+**
                    To bypass incorrect frame interface resolution, the code now uses only the
+**
                    first frameAlloc call - once "fr_interface" is set, then that is used, but
+**
                    this is not entirely correct.
-->
<!--
** This program is free software: you can redistribute it and/or modify
@@ -119,7 +125,11 @@
         <parameter name="fAlloc" type="com.goldencode.ast.Aast" />
         <return name="res" init="false" />
         <rule>true
         <!-- this is a hack to avoid frame interface computation more than once.
              TODO: 4GL allows the shared frame definition/structure to be split among as many
              UPDATE, FORM, DISPLAY, etc statements as it wants - we need to compute the frame
              structure from all of these, and not only from the first one -->
         <rule>not fAlloc.isAnnotation("fr_interface")
            <variable name="formKey"</pre>
                                            type="com.goldencode.p2j.uast.FrameAstKey" />
            <variable name="nextFrameSuffix" type="java.lang.Integer" />
            <variable name="iName" type="java.lang.String" />
```

05/21/2024 2/3



05/21/2024 3/3