

User Interface - Bug #3136

Bug # 2677 (New): fix drawing and functional differences between P2J GUI and 4GL GUI

Unhandled exception after VIEW frame, due to frame widget not registered on the client

06/27/2016 12:37 PM - Hynek Cihlar

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Sergey Ivanovskiy	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Cleanup and Stabilization for GUI	case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#1 - 06/27/2016 12:40 PM - Hynek Cihlar

EDIT: Removed WITH from the sample code.

Running the following converted sample will cause an unexpected RuntimeException.

```
def frame f.  
view frame f.
```

The exception is thrown because the frame does not get registered in the widget registry on the client.

The exception call stack:

```
ThinClient.getFrame(int) line: 12917  
ThinClient.preprocessView(ScreenBuffer, Widget, int[]) line: 18278  
ThinClient.view_(int, ScreenBuffer, int[], boolean, boolean, int) line: 9061  
ThinClient.view(int, ScreenBuffer, int[], boolean, int) line: 8879  
NativeMethodAccessorImpl.invoke0(Method, Object, Object[]) line: not available [native method]  
NativeMethodAccessorImpl.invoke(Object, Object[]) line: 62  
DelegatingMethodAccessorImpl.invoke(Object, Object[]) line: 43  
Method.invoke(Object, Object...) line: 498  
MethodInvoker.invoke(Object[]) line: 76  
Dispatcher.processInbound(InboundRequest, boolean, NetResource) line: 705  
Conversation.block() line: 364  
Conversation.waitMessage(int) line: 300  
Queue.transactImpl(Message, int) line: 1141  
Queue.transact(Message, int) line: 598  
DirectSession(BaseSession).transact(Message, int) line: 223  
HighLevelObject.transact(RoutingKey, Object[]) line: 163  
RemoteObject$RemoteAccess.invokeCore(Object, Method, Object[]) line: 1425  
RemoteObject$RemoteAccess(InvocationStub).invoke(Object, Method, Object[]) line: 97  
$Proxy4.standardEntry() line: not available  
ClientCore.start(BootstrapConfig, boolean, boolean) line: 294  
ClientCore.start(BootstrapConfig, boolean) line: 104  
ClientDriver.start(BootstrapConfig) line: 202  
ClientDriver(CommonDriver).process(String[]) line: 396  
ClientDriver.process(String[]) line: 1  
ClientDriver.main(String[]) line: 265
```

#2 - 06/27/2016 01:07 PM - Hynek Cihlar

It seems that the cause of the issue is `openScope()` not being generated for a frame when it has no widgets. `openScope()` calls `LogicalTerminal.processDeferredPush()` that makes the frame registered on the client.

#3 - 06/29/2016 08:14 AM - Greg Shah

- Project changed from *Runtime Infrastructure* to *User Interface*

#4 - 06/29/2016 08:14 AM - Greg Shah

- Parent task set to #2677

#5 - 08/11/2016 08:26 AM - Greg Shah

- Target version set to *Milestone 16*

- Status changed from *New* to *WIP*

- Assignee set to *Sergey Ivanovskiy*

Hynek will provide guidance as needed.

#6 - 08/15/2016 01:19 AM - Sergey Ivanovskiy

If one of these commented lines is used, then the conversion driver generates `openScope()` for the target frame and AST file contains `FRAME_SCOPE` node,

```
def frame f.  
/*display "test" with frame f.*/  
/*enable all with frame f.*/  
view frame f.
```

and ast looks like

```
<ast col="0" id="12884901921" line="0" text="" type="FRAME_ALLOC">  
  <annotation datatype="java.lang.String" key="javaname" value="fFrame"/>  
  <annotation datatype="java.lang.String" key="name" value="f"/>  
  <annotation datatype="java.lang.String" key="classname" value="ViewFrame1F"/>  
  <annotation datatype="java.lang.String" key="pkgname" value="com.goldencode.testcases.ui.demo"/>  
  <annotation datatype="java.lang.Long" key="down" value="1"/>  
  <annotation datatype="java.lang.Long" key="expression" value="0"/>  
  <annotation datatype="java.lang.Boolean" key="unnamed" value="false"/>  
  <annotation datatype="java.lang.Boolean" key="displayed" value="true"/>  
  <annotation datatype="java.lang.Boolean" key="shared" value="false"/>  
  <annotation datatype="java.lang.Long" key="scope-id" value="12884901920"/>  
  <annotation datatype="java.lang.Long" key="scope_peer" value="146028888099"/>  
</ast>  
<ast col="0" id="12884901920" line="0" text="" type="FRAME_SCOPE">  
  <annotation datatype="java.lang.Long" key="frame-id" value="12884901921"/>  
  <annotation datatype="java.lang.Long" key="peerid" value="146028888099"/>  
</ast>  
<ast col="0" id="12884901922" line="0" text="" type="VALIDATION"/>  
<ast col="0" id="12884901890" line="0" text="statement" type="STATEMENT">  
  <ast col="1" hidden="true" id="12884901891" line="1" text="def" type="DEFINE_FRAME">  
    <annotation datatype="java.lang.Long" key="frame-id" value="12884901921"/>  
    <ast col="11" id="12884901893" line="1" text="f" type="SYMBOL"/>  
  </ast>  
</ast>  
<ast col="0" id="12884901895" line="0" text="statement" type="STATEMENT">  
  <ast col="1" id="12884901896" line="3" text="enable" type="KW_ENABLE">  
    <annotation datatype="java.lang.Long" key="scope-id" value="12884901920"/>  
    <annotation datatype="java.lang.Long" key="frame-id" value="12884901921"/>  
    <annotation datatype="java.lang.Long" key="block_par_id" value="146028888088"/>  
    <annotation datatype="java.lang.Long" key="peerid" value="146028888102"/>  
    <ast col="8" id="12884901901" line="3" text="all" type="KW_ALL"/>  
    <ast col="12" hidden="true" id="12884901904" line="3" text="with" type="FRAME_PHRASE">  
      <annotation datatype="java.lang.Long" key="frame-id" value="12884901921"/>  
      <ast col="17" hidden="true" id="12884901906" line="3" text="frame" type="KW_FRAME">
```

```

        <ast col="23" id="12884901909" line="3" text="f" type="WID_FRAME"/>
    </ast>
</ast>
</ast>
<ast col="0" id="12884901911" line="0" text="statement" type="STATEMENT">
    <ast col="1" id="12884901912" line="4" text="view" type="KW_VIEW">
        <annotation datatype="java.lang.Long" key="scope-id" value="12884901920"/>
        <annotation datatype="java.lang.Long" key="frame-id" value="12884901921"/>
        <annotation datatype="java.lang.Long" key="block_par_id" value="146028888088"/>
        <annotation datatype="java.lang.Long" key="peerid" value="146028888107"/>
        <ast col="6" hidden="true" id="12884901915" line="4" text="frame" type="KW_FRAME">
            <ast col="12" id="12884901918" line="4" text="f" type="WID_FRAME"/>
        </ast>
    </ast>
</ast>
</ast>

```

otherwise the AST has only FRAME_ALLOC, VALIDATION and STATEMENT nodes.

```

<ast col="0" id="154618822675" line="0" text="" type="FRAME_ALLOC">
    <annotation datatype="java.lang.String" key="javaname" value="fFrame"/>
    <annotation datatype="java.lang.String" key="name" value="f"/>
    <annotation datatype="java.lang.String" key="classname" value="ViewFrame2F"/>
    <annotation datatype="java.lang.String" key="pkgname" value="com.goldencode.testcases.ui.demo"/>
    <annotation datatype="java.lang.Long" key="down" value="1"/>
    <annotation datatype="java.lang.Long" key="expression" value="0"/>
    <annotation datatype="java.lang.Boolean" key="unnamed" value="false"/>
    <annotation datatype="java.lang.Boolean" key="displayed" value="false"/>
    <annotation datatype="java.lang.Boolean" key="shared" value="false"/>
</ast>
<ast col="0" id="154618822676" line="0" text="" type="VALIDATION"/>
<ast col="0" id="154618822658" line="0" text="statement" type="STATEMENT">
    <ast col="1" hidden="true" id="154618822659" line="2" text="def" type="DEFINE_FRAME">
        <annotation datatype="java.lang.Long" key="frame-id" value="154618822675"/>
        <ast col="11" id="154618822663" line="2" text="f" type="SYMBOL"/>
    </ast>
</ast>
<ast col="0" id="154618822665" line="0" text="statement" type="STATEMENT">
    <ast col="1" id="154618822666" line="3" text="view" type="KW_VIEW">
        <annotation datatype="java.lang.Long" key="frame-id" value="154618822675"/>
        <annotation datatype="java.lang.Long" key="block_par_id" value="158913789976"/>
        <annotation datatype="java.lang.Long" key="peerid" value="158913789987"/>
        <ast col="6" hidden="true" id="154618822669" line="3" text="frame" type="KW_FRAME">
            <ast col="12" id="154618822672" line="3" text="f" type="WID_FRAME"/>
        </ast>
    </ast>
</ast>

```

#7 - 08/15/2016 01:38 AM - Sergey Ivanovskiy

But these AST's are generated finally by CB phase, these nodes have been added

```
<ast col="0" id="12884901920" line="0" text="" type="FRAME_SCOPE">
  <annotation datatype="java.lang.Long" key="frame-id" value="12884901921"/>
  <annotation datatype="java.lang.Long" key="peerid" value="150323855395"/>
</ast>
<ast col="0" id="12884901922" line="0" text="" type="VALIDATION"/>
```

and in the the target case FRAME_SCOPE has been missed.

#8 - 08/15/2016 09:49 AM - Greg Shah

The problem is that the VIEW frame is not being treated properly in annotations/frame_scoping.rules where we create the FRAME_SCOPE node. Search on prog.frame_scope in that file and then "back track" from there.

#9 - 08/15/2016 09:53 AM - Sergey Ivanovskiy

Greg Shah wrote:

The problem is that the VIEW frame is not being treated properly in annotations/frame_scoping.rules where we create the FRAME_SCOPE node. Search on prog.frame_scope in that file and then "back track" from there.

Thank you, now I am using simply <action>println("DEBUG: get_named_frame %s, cnt = %s, scope= %s", name, cnt, scope)</action> to investigate AST transformations. Is there another way to track this process?

#10 - 08/15/2016 09:59 AM - Sergey Ivanovskiy

It would be helpful to log all applied and processed rules for this small program like it was described by [com/goldencode/p2j/pattern/package.html](http://com.goldencode/p2j/pattern/package.html).

#11 - 08/15/2016 10:05 AM - Greg Shah

It would be helpful to log all applied and processed rules for this small program like it was described by [com/goldencode/p2j/pattern/package.html](http://com.goldencode/p2j/pattern/package.html).

What are you referencing here?

#12 - 08/15/2016 10:08 AM - Greg Shah

now I am using simply `<action>println("DEBUG: get_named_frame %s, cnt = %s, scope= %s", name, cnt, scope)</action>` to investigate AST transformations. Is there another way to track this process?

There is the `-d3` (debug level 3) switch to the `ConversionDriver` that enables very verbose output.

I personally don't ever use it. I just read the rule-set code and compare the rules (which encode matching patterns) with the structure of the tree that is being processed. If needed, I do put some logging in like you have done.

#13 - 08/15/2016 10:12 AM - Sergey Ivanovskiy

Thank you, now I can proceed independently. This part of `com/goldencode/p2j/pattern/package.html`

The following diagram shows a trivial example. The legend lists each event in the order that the events are triggered. The event number matches the numbers on the tree nodes and the arrows. The arrows designate events that are triggered in between `<walk-rules>`, while moving between the visits of two related nodes. The numbers in a node specify the case where there is a specific visit to a node (`<init-rules>`, `<walk-rules>` and `<post-rules>`).

gives an example how rules are applied by their execution order.

#14 - 08/15/2016 10:24 AM - Greg Shah

I manually wrote that as documentation. There is no tool or setting to generate that.

#15 - 08/15/2016 10:36 AM - Hynek Cihlar

Greg Shah wrote:

I manually wrote that as documentation. There is no tool or setting to generate that.

XML Rules --XSLT--> Java/Kotlin/Scala ---> Debug. Sorry for spoiling :-).

#16 - 08/15/2016 11:35 AM - Greg Shah

XML Rules --XSLT--> Java/Kotlin/Scala ---> Debug.

:)

Our plan (to be implemented sometime next year) is to implement TRPL 2.0 which most likely be based on Scala. I've looked at Kotlin. Kotlin does not provide enough of the features needed to implement some of the TRPL 2.0 syntax that we are wanting to explore. For example, the operator overloading in Kotlin is limited to some common cases, but does not allow defining your own operators and does not allow overloading things like assignment or the logical operators (which have flow control/short circuiting behavior). All of these cases are of great interest in TRPL 2.0. Another example, is how the Scala pattern matching syntax might be extended and used in conjunction with TRPL. Kotlin does not provide a rich approach to this.

Anyway, one of the improvements that we get "for free" by moving to Scala is debugging. We will all be very happy when that is available. Until then, we will have to go with what we've got now. Sorry.

#17 - 08/15/2016 01:19 PM - Sergey Ivanovskiy

Greg, Hynek, do you know why the scope node is removed if the referred frame has no widgets. The source rule is annotations/frame_scoping.rules, line 2145, defined by this function "need_process"

```
<!-- empty frame, no widgets -->
<rule>tmp.size() == 0
  <action>rlist = #(java.util.Set) frame.get(frameRef)</action>

  <action>iter = rlist.iterator()</action>

  <while>iter.hasNext()
    <action>
      ref = #(com.goldencode.ast.Aast) iter.next()
    </action>

    <rule>ref.type == prog.frame_scope
      <action>ref.remove()</action>
      <action>iter.remove()</action>
    </rule>
```

#18 - 08/15/2016 01:37 PM - Greg Shah

I don't know for sure. I suspect that it may have been some "safety" code that was put in there because at some point we had some incorrect frame scoping processing and this cleaned up.

I don't know if this is or ever was needed. Disabling it and seeing the differences generated in MAJIC might tell you a great deal. If there are many new openScope() calls added and those calls should not really exist, then we either need to tighten up our scoping logic or filter out this VIEW frame case from this removal.

#19 - 08/15/2016 03:02 PM - Sergey Ivanovskiy

Created a task branch, 3136a. I am trying to find some hints from the original commit.

```
bzr diff -c4417 rules/annotations/frame_scoping.rules
=== modified file 'rules/annotations/frame_scoping.rules'
--- rules/annotations/frame_scoping.rules      2006-06-26 13:15:15 +0000
+++ rules/annotations/frame_scoping.rules      2006-06-27 08:18:08 +0000
@@ -174,6 +174,7 @@
**
**                                no explicit frame defined.
** 086 SIY 20060626 CHG   @27564 Improved handling of the NEXT-PROMPT with
**                                references to current default frame.
+** 087 SIY 20060627 CHG   @27571 More careful cleanup of empty frames.
*/
-->

@@ -1676,6 +1677,24 @@
        </action>
        </rule>
    </while>
+
+        <!-- empty frame, no widgets -->
+        <rule>tmp.size() == 0
+            <action>rlist = #(java.util.List) frame.get(frameRef)</action>
+
+            <action>iter = rlist.iterator()</action>
+
+            <while>iter.hasNext()
+                <action>
+                    ref = #(com.goldencode.p2j.uast.Aast) iter.next()
+                </action>
+
+                <rule>ref.type == prog.frame_scope
+                    <action>ref.remove()</action>
+                    <action>iter.remove()</action>
+                </rule>
+            </while>
+        </rule>
    </rule>

    <rule>tmp.size() != 0
```

#20 - 08/15/2016 04:33 PM - Sergey Ivanovskiy

Planning to test MAJIC conversion against the committed revision 11080(3136a).

```
=== modified file 'rules/annotations/frame_scoping.rules'
--- rules/annotations/frame_scoping.rules      2016-03-14 14:17:22 +0000
+++ rules/annotations/frame_scoping.rules      2016-08-15 20:23:54 +0000
@@ -2067,6 +2067,7 @@
     <variable name="iter"      type="java.util.Iterator" />
     <variable name="ref"       type="com.goldencode.ast.Aast" />
     <variable name="others"    init="false" />
+    <variable name="vref"      init="false" />
     <return name="gen"        init="false" />

    <rule>true
@@ -2153,7 +2154,7 @@
        ref = #(com.goldencode.ast.Aast) iter.next()
    </action>

-    <rule>ref.type == prog.frame_scope
+    <rule>ref.type == prog.frame_scope and !vref
        <action>ref.remove()</action>
        <action>iter.remove()</action>
    </rule>
@@ -2181,6 +2182,7 @@
    -->
    <rule>ref.type == prog.kw_view
        <action>others = true</action>
+        <action>vref = true</action>
    </rule>
  </while>
</rule>
```

#21 - 08/16/2016 06:08 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

Planning to test MAJIC conversion against the committed revision 11080(3136a).

Regression testing by run_regression.sh conv-regression -Dsave.candidate=y -Dsrc.stable=20160816a -Dsrc.candidate=20160816b reports differences in the converted source in one file in comparison with the Majic converted source based on the P2J trunc version

```
diff -r generated/20160816a/src/aero/timco/majic/util/Labadj.java generated/20160816b/src/aero/timco/majic/uti
l/Labadj.java
245a246
> fTotalFrame.openScope();
```

Don't know the changes are bad or good (correct)?

#22 - 08/16/2016 09:22 AM - Greg Shah

Don't know the changes are bad or good (correct)?

Did you look at util/labadj.p.cache?

It has these references to f-total:

```
def new shared frame f-total.
...
view frame f-total.
```

This seems to be the exact case you were trying to resolve.

In Progress, when this is executed, nothing is displayed. Normally, I would be tempted to just remove the frame completely as a dead resource. The only problem is if the frame is shared and some downstream code then uses it in a way that adds widgets (dynamic ones) or otherwise can cause some visible behavior or result.

For the original case:

```
def frame f.
view frame f.
```

If these are the only references to that frame, since this is not a shared resource I'm not sure why we can't just remove these completely.

#23 - 08/16/2016 09:40 AM - Sergey Ivanovskiy

No, I only looked at the java converted file. These code changes don't distinguish these cases. Is there some attribute of a frame node that helps to recognize that this frame is shared?

#24 - 08/16/2016 09:42 AM - Sergey Ivanovskiy

Found this one `<variable name="frameShr" init="Shared" />`.

#25 - 08/16/2016 09:53 AM - Greg Shah

Found this one `<variable name="frameShr" init="Shared" />`.

This is something that can be used in `frame_scoping.rules` but it is not useful outside of that.

I am suggesting that for non-shared empty frames we might want to write new TRPL code to remove the define frame and the view frame. But to be useful, we would have to do this very early on (well before `frame_scoping.rules`) so that all the `frame_scope` nodes would never be created.

On the other hand, your current approach should be OK so long as it operates properly at runtime. It may not clean up these "dead" frames, but as long as it works we will probably leave it as is.

#26 - 08/16/2016 01:49 PM - Sergey Ivanovskiy

I tried to come through these tests `run_regression.sh runtime-regression -Dmajic-built=true` for 3136a branch, but they were failed. CTRL-C tests were failed first and then the next series of tests were failed.

Greg, Hynek could you help to pass regression testing for 3136a. What are the directories that must be checked first and zipped? May be I must try to repeat tests but the conversion tests help to determine only one difference in the converted source. The Majic documentation lists these important directories

<code>~/testing/majic_baseline/reports/</code>	Captured report files for automated comparison.
<code>~/testing/majic_baseline/results/</code>	Summary and details of testing results.
<code>~/testing/majic_baseline/screens/</code>	Captured screenshots for automated comparison.

#27 - 08/16/2016 02:04 PM - Greg Shah

Did you read the "Interpreting the Results" section of that document?

#28 - 08/16/2016 02:32 PM - Sergey Ivanovskiy

Thank you, I don't notice it at the first run.

The first part, "Ctrl-C" test suite, is that

Test Set	NOT_RUN	PASSED	FAILED	FAILED_DEPENDENCY
1 - gso_ctrlc_tests	0	74	0	0
2 - gso_ctrlc_3way_tests	0	3	3	0
Total	0	77	3	0

and gso_ctrlc_3way_tests is failed due to these reasons

- 1. failure in step 18: 'Timeout while waiting for event semaphore to be posted.'
- 2. failure in step 15: 'timeout occurred before all output could be read [null] (1 bytes read)'
- 3. failure in step 1: 'Timeout while waiting for event semaphore to be posted.'

The detailed report "gso_ctrlc_3way_tests" is that

Test	NOT_RUN	PASSED	FAILED	FAILED_DEPENDENCY
1 - ctrlc_10_session1	0	32	0	0
2 - ctrlc_10_session3	0	25	0	0
3 - ctrlc_10_session4	0	28	0	0
4 - ctrlc_11_session1	10	17	1	0
5 - ctrlc_11_session3	8	14	1	0
6 - ctrlc_11_session4	25	0	1	0
Total	43	116	3	0

and

- 1 to 3 tests passed
- 4 ctrlc_11_session1 CTRL-C 11 (Session 1) - CL - GSO - RFQ. FAILED
CONCURRENT BACKOUT Driver 0 348.251
failure in step 18: 'Timeout while waiting for event semaphore to be posted.'
- 5 ctrlc_11_session3 CTRL-C 11 (Session 3) - CL - GSO - RFQ. FAILED
CONCURRENT BACKOUT Driver 1 111.739
failure in step 15: 'timeout occurred before all output could be read [null] (1 bytes read)'
- 6 ctrlc_11_session4 CTRL-C 11 (Session 4) - CL - GSO - RFQ. FAILED
CONCURRENT BACKOUT Driver 2 300.001
failure in step 1: 'Timeout while waiting for event semaphore to be posted.'

#30 - 08/16/2016 03:07 PM - Greg Shah

These CTRL-C 3-way failures commonly happen on devsrv01. Ignore them.

#31 - 08/16/2016 03:08 PM - Greg Shah

Also, please don't post MAJIC-specific details in this task which is visible to all customers.

#32 - 08/16/2016 04:32 PM - Greg Shah

After some emails to discuss the test results, runtime testing is deemed to have passed.

The code in 3136a needs a history entry.

Does the P2J runtime for this testcase work the same way as in the 4GL?

#33 - 08/16/2016 05:22 PM - Sergey Ivanovskiy

- *File empty-frame.png added*

This testcase

```
def frame f.  
view frame f.
```

works similarly, but their look and feels are different. The native 4GL window displays a small white rectangle (square) with a 1-pixel black border to represent an empty frame.



#34 - 08/16/2016 05:25 PM - Sergey Ivanovskiy

Committed revision 11081 added comments and a history entry.

#35 - 08/16/2016 06:45 PM - Greg Shah

but their look and feels are different.

OK, please fix this.

Also, please post what the P2J result is for both GUI and ChUI.

By the way, in ChUI, the 4GL displays nothing. Make sure P2J acts the same.

#36 - 08/17/2016 10:32 AM - Sergey Ivanovskiy

Is it correct that frames are pushed to the client in a sequence that is in a reverse order to their appearances in the 4GL source. For an example, this source

```
def var i as integer.  
def frame a.  
view frame a.  
def frame b.  
view frame b.  
def frame c i.  
view frame c.  
def frame d.  
view frame d.
```

is converted to

```
**  
 * Business logic (converted to Java from the 4GL source code  
 * in demo/view-frame-2.p).  
 */  
public class ViewFrame2  
{  
    ViewFrame2D dFrame = GenericFrame.createFrame(ViewFrame2D.class, "d");  
  
    ViewFrame2C cFrame = GenericFrame.createFrame(ViewFrame2C.class, "c");  
  
    ViewFrame2B bFrame = GenericFrame.createFrame(ViewFrame2B.class, "b");  
  
    ViewFrame2A aFrame = GenericFrame.createFrame(ViewFrame2A.class, "a");  
  
    /**  
     * External procedure (converted to Java from the 4GL source code  
     * in demo/view-frame-2.p).  
     */  
    public void execute()  
    {  
        externalProcedure(new Block()  
        {  
            integer i = UndoableFactory.integer();
```

```

    public void body()
    {
        aFrame.openScope();
        bFrame.openScope();
        cFrame.openScope();
        dFrame.openScope();

        aFrame.view();

        bFrame.view();

        cFrame.view();

        dFrame.view();
    }
}

```

Thus these frames definitions are pushed in this order [d, c, b, a] to the client.

#37 - 08/17/2016 10:57 AM - Sergey Ivanovskiy

- *File sequence_of_empty_frames.png added*

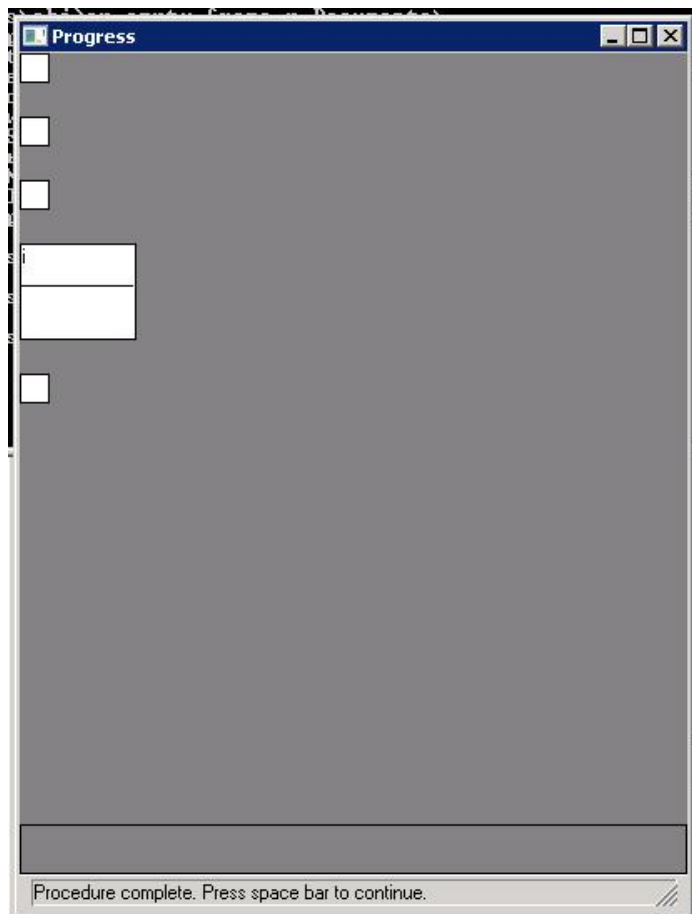
This question appears due to the layout issues if my understanding is correct. For an example this program

```

def var i as integer.
def frame a.
view frame a.
def frame b.
view frame b.
def frame c.
view frame c.
def frame d i.
view frame d.
def frame e.
view frame e.

```

must have this look and feel



And `frame.postprocess(newFrame, screenDefinition.isSuppressRedraw());` must layout frames correctly.

#38 - 08/17/2016 11:03 AM - Sergey Ivanovskiy

I think that the frames layout is sensitive to the frames sequence definition received by the client. It seems that the definitions must preserve the order of definitions. Is it correct?

#39 - 08/17/2016 11:22 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

I think that the frames layout is sensitive to the frames sequence definition received by the client. It seems that the definitions must preserve the order of definitions. Is it correct?

No, it is not correct.

The note 35 issue can be resolved if we set the default width and height for the empty frames.

#40 - 08/18/2016 01:14 AM - Sergey Ivanovskiy

To test if frames would be placed correctly the default pixel width and height were set by changing the corresponding frame definition. For an example if a frame, a, is an empty frame, then the frame definition is changed to

```
public interface ViewFrame2A
extends CommonFrame
{
    public static final Class configClass = ViewFrame2ADef.class;

    public static class ViewFrame2ADef
    extends WidgetList
    {
        public void setup(CommonFrame frame)
        {
            frame.setDown(1);
            frame.setWidthPixels(16);
            frame.setHeightPixels(16);
        }
    }
}
```

and the produced result is 4GL GUI look and feel. But I encountered some problems to move this code to the client side FrameGuiImpl. Hope that it will be resolved sooner.

#41 - 08/18/2016 01:46 AM - Sergey Ivanovskiy

- File P2J-CHUI-empty-frame.png added

- File P2J-empty-frame.png added

Greg Shah wrote:

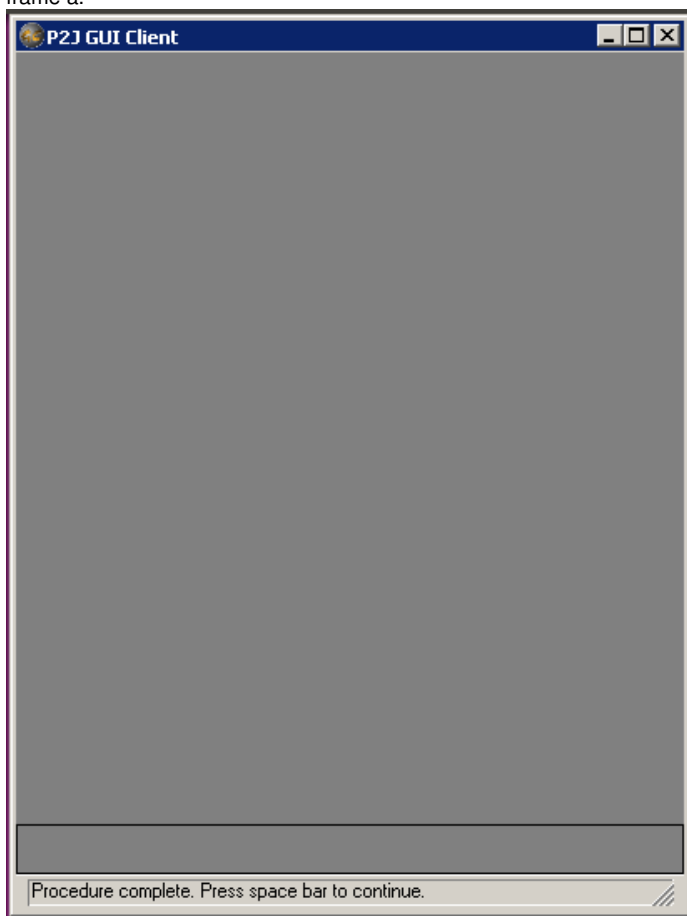
but their look and feels are different.

OK, please fix this.

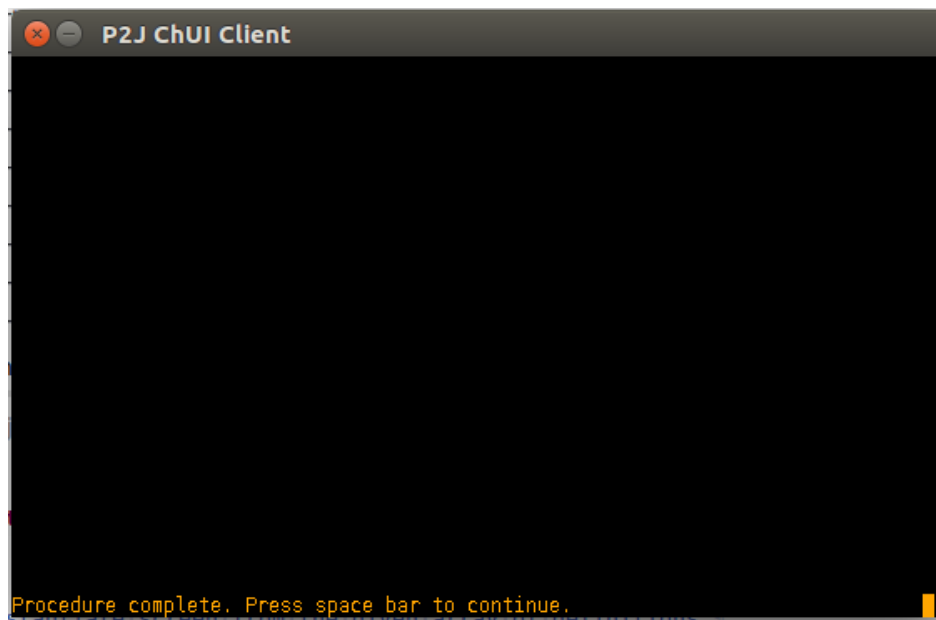
Also, please post what the P2J result is for both GUI and ChUI.

By the way, in ChUI, the 4GL displays nothing. Make sure P2J acts the same.

Terminal P2J client doesn't display empty frames. The Swing GUI and the Swing CHUI clients produce respectively these result for def frame a. view frame a.



and



#42 - 08/18/2016 07:38 AM - Sergey Ivanovskiy

- File 3136a_2.txt added

Please review the rebased revision 11084 that fixes an empty frame look and feel according to Progress 4GL GUI look and feel. The terminal client doesn't display empty frames with this revision too.

#43 - 08/18/2016 07:56 AM - Sergey Ivanovskiy

Committed revision 11085 adds a safety fix to set the default size only if the certain condition holds

```
=== modified file 'src/com/goldencode/p2j/ui/client/gui/FrameGuiImpl.java'
--- src/com/goldencode/p2j/ui/client/gui/FrameGuiImpl.java      2016-08-18 11:26:47 +0000
+++ src/com/goldencode/p2j/ui/client/gui/FrameGuiImpl.java      2016-08-18 11:51:25 +0000
@@ -498,8 +498,10 @@
     AbstractContainer<GuiOutputManager> container = getContentPane();
     if (container != null)
     {
-        // if this frame has no widgets, then set its default dimension.
-        if (container.widgets().length == 0)
+        // if this frame has no widgets and its dimension is not set,
+        // then set its default dimension.
+        if (newFrame && container.widgets().length == 0 && !config().fixedWidth &&
+            !config().fixedHeight)
         {
```

```
setWidth(defaultWidth);  
setHeight(defaultHeight);
```

#44 - 08/18/2016 08:35 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Committed revision 11085 adds a safety fix to set the default size only if the certain condition holds
[...]

I am wondering whether the default size should be set when the frame is actually realized. Consider the case when an empty frame is created and a widget is added to the frame later.

#45 - 08/18/2016 08:45 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

Committed revision 11085 adds a safety fix to set the default size only if the certain condition holds
[...]

I am wondering whether the default size should be set when the frame is actually realized. Consider the case when an empty frame is created and a widget is added to the frame later.

Please explain it more thoroughly. What is the event that happens when the frame has been realized?. Do you have a simple example?

#46 - 08/18/2016 08:53 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

What is the event that happens when the frame has been realized?. Do you have a simple example?

Please see `Frame.realizeFrame()` and check when the method is called.

#47 - 08/18/2016 10:30 AM - Sergey Ivanovskiy

Ok, thank you.

#48 - 08/18/2016 10:35 AM - Greg Shah

- File *empty_frame_display_in_openedge_chui.png* added

Terminal P2J client doesn't display empty frames.

Please explain what you mean by this. Is there some exception,abend or error?

The Swing ChUI client also doesn't display the empty frames (there is no output on the terminal), which is shown in your image from note 41.

And the 4GL (at least in my testing on lindev01) looks the same as our Swing ChUI result:



#49 - 08/18/2016 10:42 AM - Sergey Ivanovskiy

I mean that my changes don't influence the terminal client and CHUI client that these clients work already properly in the considered testcase. Thus, I need only to fix the GUI client's compliance with Progress 4GL.

#50 - 08/18/2016 10:46 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

Committed revision 11085 adds a safety fix to set the default size only if the certain condition holds
[...]

I am wondering whether the default size should be set when the frame is actually realized. Consider the case when an empty frame is created and a widget is added to the frame later.

I think now if we want to fix the static case, then we can use !config().dynamic, can't?

#51 - 08/18/2016 10:50 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

Committed revision 11085 adds a safety fix to set the default size only if the certain condition holds
[...]

I am wondering whether the default size should be set when the frame is actually realized. Consider the case when an empty frame is created and a widget is added to the frame later.

I think now if we want to fix the static case, then we can use !config().dynamic, can't?

Yes, but wouldn't it be nice to have the dynamic case fixed as well (assuming it is not a significantly more work)?

#52 - 08/18/2016 11:24 AM - Sergey Ivanovskiy

Yes, agree, please review the committed revision 11086. I moved this code to Frame.realizeFrame overridden method. Is it ENABLE statement can add new widget to the frame?

```
=== modified file 'src/com/goldencode/p2j/ui/client/gui/FrameGuiImpl.java'
--- src/com/goldencode/p2j/ui/client/gui/FrameGuiImpl.java      2016-08-18 11:54:30 +0000
+++ src/com/goldencode/p2j/ui/client/gui/FrameGuiImpl.java      2016-08-18 15:06:58 +0000
@@ -483,32 +483,30 @@
     }

    /**
     * This method is invoked when frame is constructed and configuration of all widgets is set,
     * but no other actions are performed. It has been overridden in order to set the default size
     * for an empty frame according to the Progress 4GL GUI look and feel.
     * Realize the frame, only if it's not yet realized. It has been overridden in order to set
     * the default size for an empty frame according to the Progress 4GL GUI look and feel.
     *
     * @param newFrame
     *         true value indicates that doLayout must be processed for this frame.
     * @param suppressRedraw
     *         true means suppress redrawing of frame.
     * @param realize
     *         Flag indicating if the frame should be realized.
     */
    @Override
    public void postprocess(boolean newFrame, boolean suppressRedraw)
    public void realizeFrame(boolean realize)
    {
        AbstractContainer<GuiOutputManager> container = getContentPane();
        if (container != null)
        super.realizeFrame(realize);
        if (realize)
        {
            // if this frame has no widgets and its dimension is not set,
            // then set its default dimension.
            if (newFrame && container.widgets().length == 0 && !config().fixedWidth &&
                !config().fixedHeight)
            AbstractContainer<GuiOutputManager> container = getContentPane();
            if (container != null)
            {
                setWidth(defaultWidth);
                setHeight(defaultHeight);
                // if this frame has no widgets and its dimension is not set,
                // then set its default dimension.
                if (container.widgets().length == 0 && !config().fixedWidth && !config().fixedHeight)
                {
                    setWidth(defaultWidth);
                    setHeight(defaultHeight);
                }
            }
        }
        super.postprocess(newFrame, suppressRedraw);
    }

    /**
```

#53 - 08/18/2016 12:06 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Yes, agree, please review the committed revision 11086. I moved this code to Frame.realizeFrame overridden method.

The change looks OK. One minor thing instead of setHeight() and setWidth() call setOuterSize(), this will save one call to setOuterSize().

Is it ENABLE statement can add new widget to the frame?
[...]

See an example:

```
DEFINE FRAME f.  
VIEW FRAME f.  
  
DEF VAR h AS HANDLE.  
CREATE RECTANGLE h ASSIGN  
    WIDTH-PIXELS = 2  
    HEIGHT-PIXELS = 2.  
  
h:FRAME = FRAME f:HANDLE.  
h:VISIBLE = TRUE.  
  
WAIT-FOR GO OF THIS-PROCEDURE.
```

#54 - 08/18/2016 12:11 PM - Hynek Cihlar

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

Yes, agree, please review the committed revision 11086. I moved this code to Frame.realizeFrame overridden method.

The change looks OK. One minor thing instead of `setHeight()` and `setWidth()` call `setOuterSize()`, this will save one call to `setOuterSize()`.

I should have mentioned that I didn't test the runtime behavior. You may find out that setting the default size sooner (for example the posprocess phase) was actually correct.

See an example:

[...]

Or another case when the realization is moved after the widget is added.

```
DEFINE FRAME f.  
  
DEF VAR h AS HANDLE.  
CREATE RECTANGLE h ASSIGN  
    WIDTH-PIXELS = 2  
    HEIGHT-PIXELS = 2.  
  
h:FRAME = FRAME f:HANDLE.  
  
VIEW FRAME f.  
  
WAIT-FOR GO OF THIS-PROCEDURE.
```

#55 - 08/18/2016 04:02 PM - Sergey Ivanovskiy

Thank you for these examples. Committed revision 11087 uses `setOuterSize` instead of `setWidth` and `setHeight`. It seems that `setOuterSize` has a more advanced logic than the replacing one.

#56 - 08/18/2016 04:16 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Thank you for these examples. Committed revision 11087 uses `setOuterSize` instead of `setWidth` and `setHeight`. It seems that `setOuterSize` has a more advanced logic than the replacing one.

setWidth() and setHeight() just forwards to setOuterSize().

Also, did you check the following case?

```
def frame f.  
/* what should happen when setting size smaller than the default before view? */  
frame f:width-pixels = 10.  
frame f:height-pixels = 10.  
view frame f.  
/* what should happen when setting size smaller than the default after view? */  
frame f:width-pixels = 10.  
frame f:height-pixels = 10.  
/* also consider a frame with widgets in it */
```

#57 - 08/18/2016 04:24 PM - Sergey Ivanovskiy

3136a is rebased to the current revision 11088. No, I have checked only how Progress 4GL displays empty frames.

#58 - 08/18/2016 04:31 PM - Sergey Ivanovskiy

I will check and compare these examples and their variations on windev01 with P2J later.

#59 - 08/18/2016 04:34 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

No, I have checked only how Progress 4GL displays empty frames.

IMHO size assignment belongs to the scope of "how 4GL displays empty frames". The customer's GUI tabbed interface uses empty frames so I think we should make sure we get this right.

#60 - 08/19/2016 07:27 AM - Sergey Ivanovskiy

I have checked this program on winddev01

```
DEFINE FRAME f.  
/* what should happen when setting size smaller than the default before view? */  
frame f:width-pixels = 28.  
frame f:height-pixels = 5.  
VIEW FRAME f.  
PAUSE.  
/* what should happen when setting size smaller than the default after view? */  
frame f:width-pixels = 10.  
frame f:height-pixels = 10.  
PAUSE.
```

```
/* also consider a frame with widgets in it */
```

```
DEFINE FRAME g.  
VIEW FRAME g.  
PAUSE.
```

```
DEF VAR h AS HANDLE.  
CREATE RECTANGLE h ASSIGN  
    WIDTH-PIXELS = 2  
    HEIGHT-PIXELS = 2.
```

```
h:FRAME = FRAME g:HANDLE.  
h:VISIBLE = TRUE.  
PAUSE.  
frame g:width-pixels = 28.  
frame g:height-pixels = 5.
```

```
PAUSE.
```

```
WAIT-FOR GO OF THIS-PROCEDURE.
```

and observed that the minimal frame size is preserved and equals to 16 pixels in my environment. Progress 4GL doesn't make frames to be less than this minimal size. Don't know it is possible to change this minimal size. Is the minimal size related to the current font settings?

#61 - 08/19/2016 08:58 AM - Sergey Ivanovskiy

It seems that P2J works with the following example incorrectly. It squeezes the frame size to the size of the contained rectangle widget.

```
DEFINE FRAME g.  
VIEW FRAME g.  
PAUSE.
```

```
DEF VAR h AS HANDLE.  
CREATE RECTANGLE h ASSIGN  
    WIDTH-PIXELS = 2  
    HEIGHT-PIXELS = 2.
```

```
h:FRAME = FRAME g:HANDLE.  
h:VISIBLE = TRUE.  
PAUSE.  
frame g:width-pixels = 28.  
frame g:height-pixels = 5.
```

```
PAUSE.
```

```
WAIT-FOR GO OF THIS-PROCEDURE.
```

Otherwise Progress 4GL preserves the frame's minimal size.

#62 - 08/19/2016 09:05 AM - Sergey Ivanovskiy

Planning to rebase.

#63 - 08/19/2016 09:19 AM - Sergey Ivanovskiy

Pushed up to revision 11089.

#64 - 08/19/2016 02:05 PM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

It seems that P2J works with the following example incorrectly. It squeezes the frame size to the size of the contained rectangle widget.
[...]
Otherwise Progress 4GL preserves the frame's minimal size.

It seems that these code `Frame.adjustDown()` is not correct or is used incorrectly to produce the desired look and feel

```
public void adjustDown()
{
.....
    calculateHeight(d, delta);

    getContentPane().setHeight(d.height);
.....
}
```

The function that must calculate the frame's height is used to set its inner size. To work around I am planning to fix `FrameGuiImpl.calculateHeight(d, delta)` based on this usage and the last finding about the minimal frame size.

#65 - 08/19/2016 02:19 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Sergey Ivanovskiy wrote:

It seems that P2J works with the following example incorrectly. It squeezes the frame size to the size of the contained rectangle widget.
[...]
Otherwise Progress 4GL preserves the frame's minimal size.

It seems that these code `Frame.adjustDown()` is not correct or is used incorrectly to produce the desired look and feel
[...]
The function that must calculate the frame's height is used to set its inner size. To work around I am planning to fix

FrameGuilImpl.calculateHeight(d, delta) based on this usage and the last finding about the minimal frame size.

Also please beware of the following case:

```
DEF FRAME f.  
MESSAGE FRAME f:WIDTH-PIXELS. /* outputs 2 with boxed frame and 0 with no-box frame in Progress*/  
VIEW FRAME f.  
MESSAGE FRAME f:WIDTH-PIXELS. /* outputs 18 in Progress */  
FRAME f:WIDTH-PIXELS = 3. /* the frame width doesn't change after this assignment */  
MESSAGE FRAME f:WIDTH-PIXELS. /* outputs 3 in Progress */
```

#66 - 08/19/2016 02:23 PM - Sergey Ivanovskiy

Please review the committed revision committed revision 11090.

#67 - 08/19/2016 02:23 PM - Sergey Ivanovskiy

- File 3136a_3.txt added

Thank you, I will check it.

#68 - 08/19/2016 02:50 PM - Sergey Ivanovskiy

Fixed the minimal size to be 18 pixels, the committed revision 11091. It seems that

```
DEFINE FRAME g.  
VIEW FRAME g.  
PAUSE.  
  
FRAME g:width-pixels = 26.  
FRAME g:height-pixels = 26.  
PAUSE.  
FRAME g:width-pixels = 30.  
FRAME g:height-pixels = 30.  
PAUSE.  
FRAME g:width-pixels = 34.  
FRAME g:height-pixels = 34.  
PAUSE.  
  
WAIT-FOR GO OF THIS-PROCEDURE.
```

It seems that P2J doesn't produce the correct squares, it looks like the frame height is larger than its width, but I am not sure.

#69 - 08/20/2016 09:13 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Fixed the minimal size to be 18 pixels, the committed revision 11091. It seems that [...]

It seems that P2J doesn't produce the correct squares, it looks like the frame height is larger than its width, but I am not sure.

In this case Screen Magnifier is your best friend ;-).

#70 - 08/21/2016 06:31 AM - Sergey Ivanovskiy

The corresponding screen shots proved that it was a visual effect that one side was looking smaller than the other one. Thus it seems that this task is complete.

#71 - 08/22/2016 10:26 AM - Greg Shah

Code Review Task Branch 3136a Revision 11091

The changes are fine.

I do suspect that the minimum size is tied to the font metrics in some way. Sergey: Please change the default font used in your testcase on windev01 to see if the 18 pixels are still used or if instead there is a different number. You can see details on how to do this in [#2546-92](#).

#72 - 08/22/2016 11:14 AM - Sergey Ivanovskiy

I copied progress.ini file on windev01

```
C:\Users\sbi>cp D:\oe102b\bin\progress.ini .\Documents\my_progress.ini
```

and changed the default font to be 18 (the old value was 8 pixels) in my_progress.ini file

```
DefaultFont=MS Sans Serif, size=18
DefaultFixedFont=Courier New, size=18
```

Then run this program Documents\frame-dim.p from command line C:\Users\sbi>prowin32 -iname Documents\my_progress.ini -basekey INI -p Documents\frame-dim.p

```
DEF FRAME f.
MESSAGE FRAME f:WIDTH-PIXELS. /* outputs 2 with boxed frame and 0 with no-box frame in Progress*/
VIEW FRAME f.
MESSAGE FRAME f:WIDTH-PIXELS. /* outputs 18 in Progress */
FRAME f:WIDTH-PIXELS = 3. /* the frame width doesn't change after this assignment */
MESSAGE FRAME f:WIDTH-PIXELS. /* outputs 3 in Progress */
```

It seems that the default window's size is changed but the minimal frame's size is still 18 pixels.

#73 - 08/22/2016 11:19 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

I copied progress.ini file on windev01

[...]

and changed the default font to be 18 (the old value was 8 pixels) in my_progress.ini file

[...]

Then run this program Documents\frame-dim.p from command line C:\Users\sbi>prowin32 -ininame Documents\my_progress.ini -basekey INI -p Documents\frame-dim.p

[...]

It seems that the default window's size is changed but the minimal frame's size is still 18 pixels.

Please do message session:pixels-per-column session:pixels-per-row in your test program to make sure the font values are applied correctly. The default pixels-per-column is 5, pixels-per-row 21.

#74 - 08/22/2016 11:27 AM - Sergey Ivanovskiy

Checked it now. I think that my_progress.ini settings has been applied correctly, for the default size=10 the program prints pixels-per-column is 7, pixels-per-row is 26. If the size=8, then pixels-per-column is 5, pixels-per-row 21.

#75 - 08/22/2016 12:14 PM - Greg Shah

OK, good.

Please add this comment to the javadoc for DEFAULT_MINIMAL_SIZE_IN_PIXELS:

"This value seems to be hard coded into the 4GL. Based on testing, this value does not change when the default font size is changed."

After adding this comment, please do some basic testing of the standalone GUI testcases. Just test enough cases to confirm that no negative effects exist.

#76 - 08/22/2016 02:55 PM - Sergey Ivanovskiy

For the committed revision 11092 I did the standalone GUI testing and didn't find visible bugs.

#77 - 08/22/2016 03:27 PM - Greg Shah

OK, please merge to trunk.

In your email notification, remember to mention that there is a single expected change that will occur for MAJIC conversion.

#78 - 08/22/2016 03:59 PM - Sergey Ivanovskiy

The task branch 3136a is merged into the trunk as bzt revision 11084.

#79 - 08/22/2016 04:37 PM - Greg Shah

- Status changed from WIP to Closed
- % Done changed from 0 to 100

#80 - 11/16/2016 12:22 PM - Greg Shah

- Target version changed from Milestone 16 to Cleanup and Stabilization for GUI

Files

empty-frame.png	3.36 KB	08/16/2016	Sergey Ivanovskiy
sequence_of_empty_frames.png	3.98 KB	08/17/2016	Sergey Ivanovskiy
P2J-empty-frame.png	4.68 KB	08/18/2016	Sergey Ivanovskiy
P2J-CHUI-empty-frame.png	8.36 KB	08/18/2016	Sergey Ivanovskiy
3136a_2.txt	2.88 KB	08/18/2016	Sergey Ivanovskiy
empty_frame_display_in_openedge_chui.png	13.6 KB	08/18/2016	Greg Shah
3136a_3.txt	3.59 KB	08/19/2016	Sergey Ivanovskiy