User Interface - Bug #4078

cannot login in Hotel demo in Swing mode

05/09/2019 07:21 AM - Alexei Kaigorodov

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Alexei Kaigorodov	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

Related issues:

Duplicated by User Interface - Bug #4072: NPE logging into Hotel GUI with Swi...	**Rejected**

History

#1 - 05/09/2019 07:31 AM - Alexei Kaigorodov

1. run deploy/server/server.sh
2. run deploy/client/client.sh
3. window appear; enter credentials hotel/hotel
4. the window disappear and recreates shortly whith empty credentials

the file deploy/server/server.log contains stacktrace

```
java.lang.NullPointerException
at com.goldencode.p2j.ui.client.gui.driver.swing.SwingEmulatedWindow.drawStringCenteredWithinBox(SwingEmulatedWindow.java:1748)
at com.goldencode.p2j.ui.client.gui.driver.swing.SwingEmulatedWindow.draw(SwingEmulatedWindow.java:1126)
at com.goldencode.p2j.ui.client.gui.driver.swing.SwingEmulatedWindow.offer(SwingEmulatedWindow.java:673)
...
at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
```

#2 - 05/09/2019 08:09 AM - Greg Shah

- Duplicated by Bug #4072: NPE logging into Hotel GUI with Swing client added

#3 - 05/09/2019 08:43 AM - Alexei Kaigorodov

- Status changed from New to WIP

- Assignee set to Alexei Kaigorodov

#4 - 05/10/2019 01:08 AM - Eric Faulhaber

- File 4078_workaround.diff added

FWIW, here is a simple workaround I used to get past the NPE so I could log in and test the widget browser with Hotel GUI. Please note that I have no idea if it is an appropriate fix. Since the original code here assumes ps.text will not be null, the problem may well be further back from this point, where ps.text is supposed to be set.

#5 - 05/10/2019 01:19 AM - Alexei Kaigorodov

Eric,
I made similar workaround - but conditional statement, not expression, to be able to set breakpoint when text==null. However, this breakpoint was never employed. Then I removed the workaround and indeed, NPE did not occur. I do not know what it was. I did not change p2j code in other way.

#6 - 05/10/2019 01:32 AM - Eric Faulhaber

I backed out the workaround and it still fails the same way for me. Did you run ant deploy.prepare in hotel_gui after rebuilding FWD without the change?

#7 - 05/10/2019 01:36 AM - Alexei Kaigorodov

No, I did not. Should I also run **ant clean** and **ant deploy.all**?

#8 - 05/10/2019 02:05 AM - Eric Faulhaber

You could, but that would wipe out the current conversion results and database and start the full conversion and database import from scratch. This is not needed if you just want to deploy a new build of FWD.

Please review the deploy.* targets in Hotel GUI's build.xml. We use ant deploy.prepare to push new FWD binaries into the deploy directory structure after rebuilding FWD. If you don't do this, any changes you made to FWD will not be picked up by the application when you restart the application server.

#9 - 05/10/2019 08:09 AM - Alexei Kaigorodov

What I found so far.
The main window of the Hotel Demo has several tabs. Each tab contains a scrollable list of items (rooms, guests, rates etc). Redrawing of each member item invokes method com.goldencode.p2j.ui.client.gui.theme.Windows10Theme::drawButton(ButtonGuiImpl button,...), where button has null fields label, mnemonic and textLabel. The drawButton method tries to get a string to display and encounters NPE. I added a workaround to that method which sets label = button.toString(), which is something like "ButtonGuiImpl{38, "null"}". Actually, no button with such label can be seen on the image of the scroll list.

So I propose:
1. make drawButton method more robust and do not issue NPE for any malformed buttons.
2. find out where that bad buttons go from and either remove them, or give them proper attributes.

#10 - 05/10/2019 09:31 AM - Greg Shah

> 1. make drawButton method more robust and do not issue NPE for any malformed buttons.

Agreed.

> 2. find out where that bad buttons go from and either remove them, or give them proper attributes.

This never happened before recent changes. This regression was possibly introduced revision 11302 (branch 3750b which was a very large set of changes for a customer proof of concept).

Perhaps some change exposed a race condition which is more likely to be seen in Swing than the web client. If you allow the application to continue running, do that button's attributes end up being set properly? If so, the question is why are we drawing before all the frame/UI initialization is complete?

#11 - 05/13/2019 08:41 AM - Alexei Kaigorodov

No, it does not look like a race condition. I clicked on different list items in different tabs and always got an NPE cased by drawing a button with many null fields.

#12 - 05/14/2019 05:38 AM - Alexei Kaigorodov

made a fix to avoid NPE in file ui/client/gui/driver/swing/SwingEmulatedWindow.java:1747:

```
if (text == null)       {
text = "NULL";
}
```

Committed revision 11311.

#13 - 05/14/2019 06:39 AM - Greg Shah

This is OK for temporarily making things work, but this isn't a fix to the root cause of the problem. The issue must relate to how the UI is being initialized. Perhaps there is something that fails earlier such that the frame/widgets never get initialized. ThinClient.pushScreenDefinition() must be called for each frame being initialized. Please trace through that call for the frame that is missing the button labels.

#14 - 05/14/2019 07:57 AM - Alexei Kaigorodov

- File data.txt added

- File stack.txt added

attached are the files with what I see when breakpoint in ThinClient.pushScreenDefinition is hit.
The ButtonConfig with id=38 is the source to create the button with null fields.
The corresponding frame of type com.goldencode.p2j.ui.DynamicFrameTemplate$DynFrameDef was never shown in widget browser.
I have no idea where to move next.

#15 - 05/15/2019 10:29 AM - Alexei Kaigorodov

Please code review task branch secure/code/p2j_repo/p2j/active/4078a revision 11311.

#16 - 05/15/2019 11:10 AM - Greg Shah

Hynek, please do a code review.

#17 - 05/15/2019 11:55 AM - Hynek Cihlar

Code review 4078a revision 11311.

Alexei, the check for ps.text null and assigning "NULL" doesn't solve the issue. Instead of an NPE we will get unexpected behavior ("NULL" displayed where it should not).

To get to the cause, you have to check where the invalid paint structure originates. Put a breakpoint in AbstractGuiDriver.drawStringWithinBox() with the condition text null. When you hit the break point, examine the call stack and find why the null text is passed to the method. From there you should be able to fix the issue.

Also please revert your change in SwingEmulatedWindow and instead check for text == null in drawStringWithinBox(). If null, throw an IllegalArgumentException. This will allow us to quickly address similar issue in the future when somebody passes null in the method.

#18 - 05/16/2019 01:59 AM - Alexei Kaigorodov

- File serverstack.txt added

I examined the call stack and found that the null text is passed to the method when the client tries to draw a button which configuration was sent by the server. On the server it was created at the start, when the client asked the server to invoke method com.goldencode.p2j.main.StandardServer#standardEntry(). The method has no parameters, but does a lot of work, including creation of the bad button configuration (see attached stacktrace).
Since you treat null button label as IllegalArgumentException, then the best way to avoid it is to prohibit the creation of such buttons. Since it looks like buttons on client side are created after button configurations sent by the server, we need to prohibit creation of bad configurations on server side.
But currently configuration is created as follows:

```
public ButtonWidget(boolean dynamic)    {
super(dynamic, new ButtonConfig());
}
```

That is, an empty ButtonConfig is created in hope it will be fixed later, but this does not happen.
Either we should prohibit such constructor call, or have to validate that proper attributes are set, some time later. If first, I need an explanation of how DynamicWidgetFactory works. If second, I need advice when it is the right moment to validate correctness of ButtonConfig.

#19 - 05/16/2019 04:04 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> I examined the call stack and found that the null text is passed to the method when the client tries to draw a button which configuration was sent by the server. On the server it was created at the start, when the client asked the server to invoke method com.goldencode.p2j.main.StandardServer#standardEntry(). The method has no parameters, but does a lot of work, including creation of the bad button configuration (see attached stacktrace).
> Since you treat null button label as IllegalArgumentException, then the best way to avoid it is to prohibit the creation of such buttons. Since it looks like buttons on client side are created after button configurations sent by the server, we need to prohibit creation of bad configurations on server side.
> But currently configuration is created as follows:
>
> ```
> public ButtonWidget(boolean dynamic) {
> super(dynamic, new ButtonConfig());
> }
> ```
>
> That is, an empty ButtonConfig is created in hope it will be fixed later, but this does not happen.
> Either we should prohibit such constructor call, or have to validate that proper attributes are set, some time later. If first, I need an explanation of how DynamicWidgetFactory works. If second, I need advice when it is the right moment to validate correctness of ButtonConfig.

Alexei, can you also post the call stack from the FWD client process? The one that includes the call to drawStringWithinBox method when text param value is null.

#20 - 05/16/2019 04:40 AM - Alexei Kaigorodov

- File clientstack.txt added

client stack attached

#21 - 05/16/2019 05:38 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> I examined the call stack and found that the null text is passed to the method when the client tries to draw a button which configuration was sent by the server. On the server it was created at the start, when the client asked the server to invoke method com.goldencode.p2j.main.StandardServer#standardEntry(). The method has no parameters, but does a lot of work, including creation of the bad button configuration (see attached stacktrace).
> Since you treat null button label as IllegalArgumentException, then the best way to avoid it is to prohibit the creation of such buttons. Since it looks like buttons on client side are created after button configurations sent by the server, we need to prohibit creation of bad configurations on server side.

Yes as you say, the problem seems to be in the bad state of the created button. It should have a label set, even if no explicit label is assigned in the legacy code (buttonHandle:LABEL = "label"). Btw. the IllegalStateException help us fail early (or earlier :-)), which will help with diagnosing similar issue in the future.

I suggest you isolate the issue if possible. The following sample should do it.

```
define var h as handle.
create button h.
define frame f with size 20 by 5.
enable all with frame f.
h:frame = frame f:handle.
wait-for close of this-procedure.
```

I assume this is a recent regression in trunk. Try to find a trunk revision that works and find where the label in the button is set. Then compare this to the broken revision.

#22 - 05/16/2019 06:57 AM - Alexei Kaigorodov

Hynek,

the proposed abl test works fine both on openEdge and FWD/Swing. The problem I guess is that the malformed button is not defined by user. When running with widget browser, that button was not seen among swing elements. When I set label="NULL" there were no "NULL" string on the screen. This button is not defined by user, is not included in the swing widget hierarchy and is not shown on the screen.

#23 - 05/16/2019 09:37 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> Hynek,
> the proposed abl test works fine both on openEdge and FWD/Swing. The problem I guess is that the malformed button is not defined by user. When running with widget browser, that button was not seen among swing elements. When I set label="NULL" there were no "NULL" string on the screen. This button is not defined by user, is not included in the swing widget hierarchy and is not shown on the screen.

When I run the test with 4078a revision 11311 I see NULL text on the button. With revision 11310 I get the NPE you posted above. You probably have some other changes in your workspace?

#24 - 05/17/2019 12:28 AM - Alexei Kaigorodov

Hynek,
yes there are another changes but they should not affect the execution. Can you please tell me how you prepare and run the test? I inserted it into testcases/uast.

#25 - 05/17/2019 01:16 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> Hynek,
> yes there are another changes but they should not affect the execution. Can you please tell me how you prepare and run the test? I inserted it into testcases/uast.

1. In testcases/uast dir create symbolic link to p2j dir (ln -s <your_p2j_4078a_location> p2j).
2. Convert the external procedure file with java -classpath ./p2j/build/lib/p2j.jar com.goldencode.p2j.convert.ConversionDriver F2+M0+CB ./test.p
3. Rebuild p2j with ant jar (the converted classes are added in the p2j source tree).
4. Run the server with cd testcases/simple/server && ./server.sh
5. Run the client with cd testcases/simple/client && ./client-swing.sh

I run the converted classes directly from my IDE (Idea), but that requires some extra configuration steps. Notably the AspectJ load-time weaving as I didn't find a good way to integrate the AspectJ compiler in the IDE's build process.

#26 - 05/17/2019 01:17 AM - Hynek Cihlar

Hynek Cihlar wrote:

> Alexei Kaigorodov wrote:
>
> > Hynek,
> > yes there are another changes but they should not affect the execution. Can you please tell me how you prepare and run the test? I inserted it into testcases/uast.
>
> 1. In testcases/uast dir create symbolic link to p2j dir (ln -s <your_p2j_4078a_location> p2j).
> 2. Convert the external procedure file with java -classpath ./p2j/build/lib/p2j.jar com.goldencode.p2j.convert.ConversionDriver F2+M0+CB ./test.p
> 3. Rebuild p2j with ant jar (the converted classes are added in the p2j source tree).
> 4. Run the server with cd testcases/simple/server && ./server.sh
> 5. Run the client with cd testcases/simple/client && ./client-swing.sh
>
> I run the converted classes directly from my IDE (Idea), but that requires some extra configuration steps. Notably the AspectJ load-time weaving as I didn't find a good way to integrate the AspectJ compiler in the IDE's build process.

Also to debug the server and client processes, add -d param to the startup scripts.

#27 - 05/17/2019 02:21 AM - Alexei Kaigorodov

Hynek,

command
4. Run the server with cd testcases/simple/server && ./server.sh

caused an error saved in server/stdout.log:

```
Error occurred during initialization of VM
java.lang.Error: java.lang.ClassNotFoundException: com.goldencode.p2j.classloader.MultiClassLoader
at java.lang.ClassLoader.initSystemClassLoader(ClassLoader.java:1469)
at java.lang.ClassLoader.getSystemClassLoader(ClassLoader.java:1436)
Caused by: java.lang.ClassNotFoundException: com.goldencode.p2j.classloader.MultiClassLoader
at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)
at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:348)
at java.lang.SystemClassLoaderAction.run(ClassLoader.java:2209)
at java.lang.SystemClassLoaderAction.run(ClassLoader.java:2195)
at java.security.AccessController.doPrivileged(Native Method)
at java.lang.ClassLoader.initSystemClassLoader(ClassLoader.java:1456)
at java.lang.ClassLoader.getSystemClassLoader(ClassLoader.java:1436)
```

And, what is the problem with AspectJ? Is it required if the test is already converted by ConversionDriver?

#28 - 05/17/2019 02:37 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> Hynek,
>
> command
> 4. Run the server with cd testcases/simple/server && ./server.sh
>
> caused an error saved in server/stdout.log:
>
> Error occurred during initialization of VM
> java.lang.Error: java.lang.ClassNotFoundException: com.goldencode.p2j.classloader.MultiClassLoader
> at java.lang.ClassLoader.initSystemClassLoader(ClassLoader.java:1469)
> at java.lang.ClassLoader.getSystemClassLoader(ClassLoader.java:1436)

Sorry I gave you wrong instructions. I modified the official way of running uast test cases in order to allow for branch switching. Instead of the symbolic link in step 1, set/export an environment variable P2J_HOME that will point to your P2J directory before running the startup scripts:

export P2J_HOME="<your p2j 4078a dir>"

> And, what is the problem with AspectJ? Is it required if the test is already converted by ConversionDriver?

AspectJ is required for FWD itself (p2j.jar). We use several aspects to support multiple of features. During the standard build (build.xml) we compile the java classes with ajc (AspectJ compiler). I was having some issues with ajc in Idea, so I configured my Idea p2j module to run with load-time AspectJ weaving, which does the same as ajc but on runtime when the Java process starts.

#29 - 05/17/2019 04:20 AM - Alexei Kaigorodov

Hynek,

testcases/simple/client/client-swing.sh is your own script? can you please share it with me?

#30 - 05/17/2019 04:26 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> Hynek,
>
> testcases/simple/client/client-swing.sh is your own script? can you please share it with me?

Yes, I created this before client.sh got the ability to run GUI clients directly. Sorry, I didn't realize that. You should be able to start the client with client.sh -g.

#31 - 05/20/2019 06:43 AM - Alexei Kaigorodov

Hynek Cihlar wrote:

> Alexei, the check for ps.text == null and assigning "NULL" doesn't solve the issue. Instead of an NPE we will get unexpected behavior ("NULL" displayed where it should not).

instead of text="NULL" there should be text="". Then your **test.p** would run the same way both on 4gl and FWD.

> To get to the cause, you have to check where the invalid paint structure originates. Put a breakpoint in AbstractGuiDriver.drawStringWithinBox() with the condition text null. When you hit the break point, examine the call stack and find why the null text is passed to the method. From there you should be able to fix the issue.

No, I was not able to find out why the null text is passed to the method, because it was passed by network.

> Also please revert your change in SwingEmulatedWindow and instead check for text null in drawStringWithinBox(). If null, throw an IllegalArgumentException. This will allow us to quickly address similar issue in the future when somebody passes null in the method.

If we do so, then your **test.p** would throw IllegalArgumentException when run on 4FD, while it runs without errors on 4gl. If we really want compatibility, then the only fix must be if (text==null) text = ""; as I stated above.

The question remains, where in hotel_gui application buttons with not initialized labels are created. But these buttons are not visible anyway, so this bug is not critical. And if we decide to fix it anyway, then another bug report must be created, with the area *hotel_gui application* and not *user interface* as the current bug.

#32 - 05/20/2019 08:21 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> Hynek Cihlar wrote:
>
> > Alexei, the check for ps.text == null and assigning "NULL" doesn't solve the issue. Instead of an NPE we will get unexpected behavior ("NULL" displayed where it should not).
>
> instead of text="NULL" there should be text="". Then your **test.p** would run the same way both on 4gl and FWD.
>
> > To get to the cause, you have to check where the invalid paint structure originates. Put a breakpoint in AbstractGuiDriver.drawStringWithinBox() with the condition text null. When you hit the break point, examine the call stack and find why the null text is passed to the method. From there you should be able to fix the issue.
>
> No, I was not able to find out why the null text is passed to the method, because it was passed by network.

Actually it is not difficult to debug both the client and the server processes. The legacy application is single-threaded and the remote calls are blocking. So to get the related server call stack of a paused client process, just attach your debugger to the server process, pause it and switch to the thread named "Conversation". This is the thread, that executes the legacy code and is blocking waiting for the client remote call to finish.

> > Also please revert your change in SwingEmulatedWindow and instead check for text null in drawStringWithinBox(). If null, throw an IllegalArgumentException. This will allow us to quickly address similar issue in the future when somebody passes null in the method.
>
> If we do so, then your **test.p** would throw IllegalArgumentException when run on 4FD, while it runs without errors on 4gl. If we really want compatibility, then

the only fix must be if (text==null) text = ""; as I stated above.

The purpose of throwing the IllegalArgumentException is to detect the unexpected application state sooner than later.

At the driver level you cannot enforce any particular legacy application behavior. In other words, there is not enough context information to decide, what the uninitialized value should be. Is it "", ? (a special unknown literal in 4gl often used in cases where a value has not been set) or any arbitrary value like "uninitialized"? For your particular case it may be an empty string, but in other cases it could be ?, or anything else.

> The question remains, where in hotel_gui application buttons with not initialized labels are created.

We know this from the test case I attached in #4078-21. The question is why the default label is not set for the button, what the label value should be and where it should be set.

> But these buttons are not visible anyway, so this bug is not critical. And if we decide to fix it anyway, then another bug report must be created, with the area *hotel_gui application* and not *user interface* as the current bug.

While the buttons are not visible, the test case #4078-21 confirms, that the invalid label value is the cause of the exception.

#33 - 05/20/2019 12:10 PM - Greg Shah

> The question remains, where in hotel_gui application buttons with not initialized labels are created. But these buttons are not visible anyway, so this bug is not critical. And if we decide to fix it anyway, then another bug report must be created, with the area hotel_gui application and not user interface as the current bug.

I disagree with this conclusion.

FWD is meant to be a reliable and compatible replacement for OpenEdge. All 4GL programs should be safe to convert and execute, so long as the same program is able to compile and execute in OpenEdge. If the syntax is invalid such that it cannot be compiled in OpenEdge, then we would not expect it to convert in FWD. As long as OpenEdge can compile the code, then it can be executed in OpenEdge. As soon as this is the case, any deviation in behavior is **assumed** to be a bug in FWD. Of course, it is possible that there could be issues with the local installation (for example, dependence on some utilities, a library, file system permissions or other platform dependency that is installed on the OpenEdge system and not on the FWD system). But here, there is no such external cause of the problem. We start by assuming the issue is not with the 4GL code.

Then we look at the behavior. In this case it is an NPE. I can tell you that OpenEdge does not fail with an NPE (or any other kind of abend) when Hotel GUI is run. This means that the bug is definitely in FWD because we should not abend. Of course, if in OpenEdge there is some kind of error or failure, then we must also generate the equivalent error or failure in FWD, but we know we would not allow an abend.

To say this differently, if the 4GL code runs in OpenEdge, we must also run it even if the code is behaves badly or is somehow logically flawed. We must faithfully reproduce these application level bugs, because we simply can't know from the code alone whether something is or is not correct about the 4GL. Our job is to privide the same resulting behavior as OpenEdge would.

Just protecting from the NPE is not a correct solution. Before recent changes to trunk, the same Hotel GUI 4GL code ran without this NPE. And we can still run this code in the web client without an NPE. There has been some kind of regression caused by the recent changes. This is the root cause of the problem and we need you to track this down.

As I noted in #4078-10, I think the regression may have been introduced in 11302, but I'm not sure. One simple thing you can do is to try trunk 11301 (convert and run Hotel GUI) to see if the problem can be reproduced. You can test back or forward from there until you find the revision that causes the issue. You can review the changes in that revision to see if something related can be seen. Regardless, once you know which revision causes the issue, you certainly will reduce the possible causes to the changes in that revision. I think that is a useful starting point for investigations.

#34 - 05/20/2019 12:16 PM - Greg Shah

> > the null text is passed to the method when the client tries to draw a button which configuration was sent by the server. On the server it was

> > created at the start, when the client asked the server to invoke method com.goldencode.p2j.main.StandardServer#standardEntry(). The method has no parameters, but does a lot of work, including creation of the bad button configuration (see attached stacktrace).
>
> that is I want to say I have no idea were to go on. The malformed button has no mapping on swing components (I checked it with widget browser). I need a clue on what happens.

For a user's session on the server, StandardServer#standardEntry() is just the equivalent of main() but on the server side. It doesn't directly do anything related to the UI. It just sets up the session and then started the converted application entry point. Starting here is not a great idea.

Any UI processing is driven by the converted application code. So there must be some code that sets up the button. Find that 4GL code, map it to the converted code and you can step through that to see how the setup works.

Eventually, there will be a call to pushScreenDefinition() to move all of that frame's state to the client. It seems that at that point the setup is already wrong.

Hynek's suggestion to work backward from the failure is a good one. The root cause is most likely much closer to the failure than to the start of the session on the server side.

#35 - 05/21/2019 08:43 AM - Alexei Kaigorodov

One simple thing you can do is to try trunk 11301 (convert and run Hotel GUI) to see if the problem can be reproduced. You can test back or forward from there until you find the revision that causes the issue. You can review the changes in that revision to see if something related can be seen.

The first revision where the NPE appears is 11300. The file with changes of that revision is attached.

The problematic file change is in src/com/goldencode/p2j/ui/client/gui/theme/MaterialTheme.java. When this file is replaced from previous revision 11299, then the hotel_gui application runs without errors.

#36 - 05/21/2019 10:44 PM - Alexei Kaigorodov

- File c11300.txt added

The file with changes from rev 11299 to 11300 attached.

#37 - 05/22/2019 06:58 AM - Alexei Kaigorodov

the change in MaterialTheme.java which causes NPE is at line 529, call to gd.drawStringWithMnemonic() changed to call to gd.drawStringWithinBoxWithMnemonic. As a result, method drawStringCenteredWithinBox is called which tries to get textLength = text.length(), but argument text equals to null.

Actually buttons with null labels were created for long time, but method gd.drawStringWithMnemonic() accepted null argument. It turned out that malformed button was not defined by user, but belongs to the class com.goldencode.hotel.ui.common.Afoldfr0FolderFrm.Afoldfr0FolderFrmDef generated from the file abl/common/Afoldfr0FolderFrm.jast. There is no file Afoldfr0FolderFrm.p or Afoldfr0FolderFrm.w.

I need a recommendation where to look to understand when and why the file Afoldfr0FolderFrm.jast was generated. The wrong button can be seen in widget browser but is not visible at the hotel_gui application screen. Probably we can turn off generation of that button without visible changes to the graphic user interface.

#38 - 05/22/2019 07:11 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> the change in MaterialTheme.java which causes NPE is at line 529, call to gd.drawStringWithMnemonic() changed to call to gd.drawStringWithinBoxWithMnemonic. As a result, method drawStringCenteredWithinBox is called which tries to get textLength = text.length(), but argument text is equals to null.
>
> Actually buttons with null labels were created for long time, but method gd.drawStringWithMnemonic() accepted null argument. It turned out that malformed button was not defined by user, but belongs to the class com.goldencode.hotel.ui.common.Afoldfr0FolderFrm.Afoldfr0FolderFrmDef generated from the file abl/common/Afoldfr0FolderFrm.jast. There is no file Afoldfr0FolderFrm.p or Afoldfr0FolderFrm.w.

The source code is in abl/common/afoldrfr0.w. You want to open the file afoldrfr0.w.cached which is the sources as processed by the preprocessor.

#39 - 05/22/2019 07:35 AM - Alexei Kaigorodov

that is, afoldrfr0.w.cached is converted in two .jast files? Where can I read how FWD chooses to convert one source file in one or two jast files? How the name of the additional jast file is generated?

#40 - 05/22/2019 08:43 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> that is, afoldrfr0.w.cached is converted in two .jast files? Where can I read how FWD chooses to convert one source file in one or two jast files? How the name of the additional jast file is generated?

Every procedure file (usually *.w, *.p) is converted in one or multiple set of files (*.ast, *.jast, *.java). One set of file is for the external procedure itself and one set for every statically defined frame. This is why you see two set of files for afoldrfr0.w. For more details see Conversion Technology Architecture.

#41 - 05/22/2019 08:46 AM - Greg Shah

The main business logic will be emitted as a class which has an execute() method (for the external procedure) and optional methods for internal procedures and functions. Triggers can be inner classes or lambdas inside that business logic class.

Statically defined frames and menus are refactored into frame definition classes and menu definition classes respectively.

Temp-tables and permanent tables are refactored into data model objects (DMOs) which are POJOs used by Hibernate.

Afoldfr0FolderFrmDef is a frame definition.

> Probably we can turn off generation of that button without visible changes to the graphic user interface.

We must assume the button maps back to specific UI code that we must honor because it was placed there by a developer at some point. And it seems that the button is being drawn otherwise why would the drawStringCenteredWithinBox() be called? Of course, perhaps it is being drawn at a z-order that makes it invisible or at a location off screen. Still, I'm not sure we can know that we should not be processing this button at this time. And removing that processing can have negative consequences on other 4GL code that expects that button to be initialized and realized. It likely would cause other parts of the runtime to fail.

As an alternative, please review what happens in other themes for this same case. That may help us understand the correct solution.

A related question: if you change the theme (search for it in the directory to see how), does the failure still occur?

#42 - 05/23/2019 06:59 AM - Alexei Kaigorodov

I found that malformed button is created in file hotel_gui/src/com/goldencode/hotel/common/Afoldfr0.java:
67:   handle lhBackgroundButton = TypeFactory.handle();
203:               DynamicWidgetFactory.createButton(lhBackgroundButton);

its prototipe is in file abl/common/afoldfr0.w.cache:
206: DEFINE VARIABLE LhBackgroundButton AS HANDLE NO-UNDO.

But file abl/common/afoldfr0.w does not contain such declaration. That is, it was inserted by preprocessor.
If I have to investigate why preprocessor did so, I need to know where the documentation on preprocessor reside.

#43 - 05/23/2019 08:14 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> I found that malformed button is created in file hotel_gui/src/com/goldencode/hotel/common/Afoldfr0.java:
> 67:   handle lhBackgroundButton = TypeFactory.handle();

> 203:   DynamicWidgetFactory.createButton(lhBackgroundButton);
>
> its prototipe is in file abl/common/afoldfr0.w.cache:
> 206: DEFINE VARIABLE LhBackgroundButton AS HANDLE NO-UNDO.
>
> But file abl/common/afoldfr0.w does not contain such declaration. That is, it was inserted by preprocessor.
> If I have to investigate why preprocessor did so, I need to know where the documentation on preprocessor reside.

The button creation statement resides in afoldml0.i. afoldfr0.w includes this file.

#44 - 05/23/2019 08:46 AM - Greg Shah

> If I have to investigate why preprocessor did so, I need to know where the documentation on preprocessor reside.

It is not likely that this is a preprocessor issue.

afoldml0.i and afoldfr0.w are used to create a tab control implemented in 4GL code. In the Hotel GUI the main window uses tabs. There may be many widgets created during that process, including buttons, rectangles and so forth. Each tab would be comprised of multiple 4GL widgets.

Since these tabs are created at runtime, the backing widgets are created using "dynamic" UI statements such as CREATE <widget> (in this case it would be CREATE BUTTON).

#45 - 05/23/2019 09:05 AM - Alexei Kaigorodov

Greg Shah wrote:

> A related question: if you change the theme (search for it in the directory to see how), does the failure still occur?

the call to gd.drawStringWithMnemonic() was changed to call to gd.drawStringWithinBoxWithMnemonic in all themes. So the failure should occur in any themes.

#46 - 05/23/2019 09:18 AM - Greg Shah

Did you test this? A change in the directory and server restart is all that is needed to test.

#47 - 05/23/2019 09:33 AM - Alexei Kaigorodov

the bug occur on MaterialTheme, Windows10Theme, Windows8Theme and ClassicTheme.

#48 - 05/23/2019 09:33 AM - Alexei Kaigorodov

I tested, the bug occurs on MaterialTheme, Windows10Theme, Windows8Theme and ClassicTheme.

#49 - 05/23/2019 09:44 AM - Greg Shah

If you have one, please post the proposed diff for your solution.

#50 - 05/23/2019 12:23 PM - Alexei Kaigorodov

- File patch11311.txt added

The diff attached as patch11311.txt.

#51 - 05/23/2019 12:46 PM - Hynek Cihlar

Alexei, the button label must be initialized to empty string. Look at the following test case.

```
def var h as handle.
create button h.
message h:label.
```

In OpenEdge the label is "", while in FWD the label is ? (the unknown literal).

It should be safe to set the label in ButtonConfig.init() for both dynamically and statically created cases. In drawStringCenteredWithinBox() keep the check for null, but instead of setting the empty string to text, just return.

#52 - 05/23/2019 11:57 PM - Alexei Kaigorodov

Hynek Cihlar wrote:

> Alexei, the button label must be initialized to empty string.

label is declared in ControlConfig, the superclass of the ButtonConfig:
public String label = null;
Why not to initialize to empty string there?

> Look at the following test case.
>
> [...]
>
> In OpenEdge the label is "", while in FWD the label is ? (the unknown literal).

Why ? and not NPE?

I tried to run this test and it works with blank label (see attached file test.png).

#53 - 05/23/2019 11:59 PM - Alexei Kaigorodov

- File test.png added

#54 - 05/24/2019 01:52 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> Hynek Cihlar wrote:
>
>> Alexei, the button label must be initialized to empty string.
>
> label is declared in ControlConfig, the superclass of the ButtonConfig:
> public String label = null;
> Why not to initialize to empty string there?

Because the empty string initial value is a specific behavior for the BUTTON widget. The other widgets should init the label to an unknown value, hence null is expected.

>> Look at the following test case.
>>
>> [...]
>>
>> In OpenEdge the label is "", while in FWD the label is ? (the unknown literal).
>
> Why ? and not NPE?

The test doesn't attach the widget to a frame and so doesn't execute drawStringCenteredWithinBox. It only prints the label value.

#55 - 05/24/2019 12:35 PM - Alexei Kaigorodov

Hynek Cihlar wrote:

> It should be safe to set the label in ButtonConfig.init() for both dynamically and statically created cases. In drawStringCenteredWithinBox() keep the check for null, but instead of setting the empty string to text, just return.

I made a branch with such fixes. It is at secure/code/p2j_repo/p2j/active/4078b. Committed revision 11311.
Please review.

#56 - 05/25/2019 02:11 PM - Hynek Cihlar

Code review 4078b revision 11311. The changes are OK. Just add the file history entries at the top of each file.

#57 - 05/27/2019 06:04 AM - Alexei Kaigorodov

Branch 4078c, revision 11311. Added the file history entries at the top of each file.

#58 - 05/27/2019 08:09 AM - Alexei Kaigorodov

Branch 4078c, revision 11311. Added the file history entries at the top of each file.

#59 - 05/29/2019 08:44 AM - Alexei Kaigorodov

I consider this task fixed. Please do code review and integrate (branch 4078c).

#60 - 05/29/2019 11:02 AM - Hynek Cihlar

Code review 4078c revision 11311.

The history entry description is misaligned in SwingEmulatedWindow.java, please fix this. Otherwise the changes are OK. The change doesn't require ChUI regression changes. Please run GUI regression tests and if they are OK the task branch can be merged to trunk.

For GUI regression tests I think it should be enough to go through all the screens in the Hotel GUI test app and check for any GUI glitches, unexpected exceptions and errors.

#61 - 05/29/2019 11:14 AM - Greg Shah

Code Review Task Branch 4078c Revision 11311

1. What is the reasoning for moving the explicit ButtonConfig.init() into an initializer block?

2. Please do NOT create a new branch for every edit of the task. Just make the edits in one branch and commit there. Let's stick with 4078c for now. You will need to "dead archive" 4078a and 4078b so that they don't "live forever".

3. The branch needs to be rebased. Make sure to follow the instructions carefully.

4. As per our coding standards, when a file is edited for the first time in a new year, the copyright date range needs to be updated. For example, ButtonConfig was last edited in 2017, so it needs an update in the copyright date range.

#62 - 05/30/2019 12:03 AM - Alexei Kaigorodov

Hynek Cihlar wrote:

> For GUI regression tests I think it should be enough to go through all the screens in the Hotel GUI test app and check for any GUI glitches, unexpected exceptions and errors.

how can I run GUI regression tests?

#63 - 05/30/2019 03:34 AM - Hynek Cihlar

Alexei Kaigorodov wrote:

> Hynek Cihlar wrote:
>
> > For GUI regression tests I think it should be enough to go through all the screens in the Hotel GUI test app and check for any GUI glitches, unexpected exceptions and errors.
>
> how can I run GUI regression tests?

We don't have an automated suite of GUI regression tests ATM. Instead we run a subset of GUI test cases (~/secure/code/p2j_repo/testcases/uast) and applications (like Hotel GUI) manually and search for deviations. For large customer applications we may also run the functional GUI test cases.

#64 - 05/30/2019 04:44 AM - Alexei Kaigorodov

Greg Shah wrote:

> Code Review Task Branch 4078c Revision 11311
>
> 1. What is the reasoning for moving the explicit ButtonConfig.init() into an initializer block?

initializer block is guaranteed to be executed when a constructor is called. in previous version, not all constructors called ButtonConfig.init().

> 2. Please do NOT create a new branch for every edit of the task. Just make the edits in one branch and commit there. Let's stick with 4078c for now. You will need to "dead archive" 4078a and 4078b so that they don't "live forever".

what is "dead archive"? I was going just to remove them.

> 3. The branch needs to be rebased. Make sure to follow the instructions carefully.

Done.

> 4. As per our coding standards, when a file is edited for the first time in a new year, the copyright date range needs to be updated. For example, ButtonConfig was last edited in 2017, so it needs an update in the copyright date range.

Fixed.

Branch ~/secure/code/p2j_repo/p2j/active/4078c is updated, please review.

#65 - 05/30/2019 02:59 PM - Greg Shah

> > What is the reasoning for moving the explicit ButtonConfig.init() into an initializer block?
>
> initializer block is guaranteed to be executed when a constructor is called. in previous version, not all constructors called ButtonConfig.init().

Fair enough. Before we accept this change, we must answer this question:

Was it intentional that ButtonConfig.init() was only used in some constructors but not all of them (the ButtonConfig(WidgetId) variant did not call it)?

Constantin/Hynek: what are your thoughts?

> what is "dead archive"? I was going just to remove them.

This is addressed here: Archive or Abandon the Branch

#66 - 05/30/2019 03:14 PM - Hynek Cihlar

Greg Shah wrote:

> > > What is the reasoning for moving the explicit ButtonConfig.init() into an initializer block?
> >
> > initializer block is guaranteed to be executed when a constructor is called. in previous version, not all constructors called ButtonConfig.init().
>
> Fair enough. Before we accept this change, we must answer this question:
>
> Was it intentional that ButtonConfig.init() was only used in some constructors but not all of them (the ButtonConfig(WidgetId) variant did not call it)?
>
> Constantin/Hynek: what are your thoughts?

I don't think it was intentional. But before the changes in this branch, the only field initialized in init() was only used in ChUI (for button widget) while the ctor not calling init() was only used in GUI. In other words we should be OK with the changes.

#67 - 05/30/2019 03:49 PM - Constantin Asofiei

Review for 4078c rev 11312:

- You also set label = ""; in ButtonConfig; this looks OK, as I can't find a way to make the buttons report the button:label attribute as unknown value. But for other widgets, the label must be considered that it can be in three states (this is just a FYI):
    - implicit, computed by the frame during layout
    - explicit, assigned by the user
    - not set to anything - and in this case the LABEL attribute is reported as ? (unknown value)
- Greg, correct me if I'm wrong, but in SwingEmulatedWindow.java I'd like to have a more technical description of what is fixed (i.e. Fixed a NPE in drawStringCenteredWithinBox, which was called with a null text, for a button without an explicit label set.). The cannot login in Hotel demo in Swing mode requires going through this task to find the reasoning behind the fix.

#68 - 05/30/2019 04:10 PM - Greg Shah

> correct me if I'm wrong, but in SwingEmulatedWindow.java I'd like to have a more technical description of what is fixed (i.e. Fixed a NPE in drawStringCenteredWithinBox, which was called with a null text, for a button without an explicit label set.). The cannot login in Hotel demo in Swing mode requires going through this task to find the reasoning behind the fix.

Agreed. Alexei: Please use this description.

#69 - 05/31/2019 06:56 AM - Alexei Kaigorodov

Description updated.

#70 - 05/31/2019 07:40 AM - Greg Shah

Code Review Task Branch 4078c Revision 11314

1. You merged the changes from trunk instead of rebasing. Merging results in the latest trunk changes being inserted in the wrong order in 4078c. Rebasing makes 4078c appear as if all changes in this task were made **on top of** the latest trunk. It is about the order of the changes in the branch history. Please read the 100% of the Source Code Management document and pay careful attention to the Rebase section. Once you have rebased to trunk 11311, then bzr log -r 11311 in 4078c should give you the same log message as bzr log -r 11311 in a trunk checkout. And a bzr diff -r 11311 in 4078c should give you only changes which are part of 4078c. After rebase, 4078c should have the revisions up to 11311 be the same as the trunk checkout, and from there on 4078c specific revisions.

To fix this:

```
bzr branch <path_to_the_trunk_rev_11311_checkout> 4078c_fixed
cd 4078c_fixed
bzr merge -c 11311 <path_to_broken_4078c>
bzr commit -m "Bug 4078 fixed according to the directions made by Hynek Cihlar."
bzr merge -c 11312 <path_to_broken_4078c>
bzr commit -m "cosmetic changes in comments"
bzr merge -c 11314 <path_to_broken_4078c>
bzr commit -m "change log description adjusted"
```

At this point the 4078c_fixed branch should look the same way that it would have if you had rebased. Check the bzr log to confirm this (it should meet the test noted above).

If so, you can use bzr push --overwrite ~/secure/code/p2j_repo/p2j/active/4078c to replace the 4078c repo with the fixed version. Then you should use bzr bind ~/secure/code/p2j_repo/p2j/active/4078c to turn it into a checkout again.

2. Please note that our coding standards have a 98 character limit on line length. The history entry goes past that. You don't need to leave the title of the task in the entry. Writing "(bug #4078)." is enough. But if the text was important, you would need to wrap it.

#71 - 06/03/2019 12:26 AM - Alexei Kaigorodov

Greg Shah wrote:

> To fix this:

Fixed. All changes made in single commit #11312.

#72 - 06/03/2019 03:07 PM - Greg Shah

Alexei Kaigorodov wrote:

> Greg Shah wrote:
>
> > To fix this:
>
> Fixed. All changes made in single commit #11312.

There is no rev 11312 in branch 4078c.

#73 - 06/04/2019 06:05 AM - Alexei Kaigorodov

Greg Shah wrote:

> There is no rev 11312 in branch 4078c.

Fixed.

#74 - 06/04/2019 01:23 PM - Greg Shah

Code Review Task Branch 4078c Revision 11312

1. You will need to rebase this branch.

2. I've committed rev 11313 which fixes:

- a formatting issue with a history entry
- 98 char line limit issues
- changed a 4 char indent into a 3 char indent

Please do a bzr diff -r11312 --using meld to review my changes. In the future, please follow these coding standards.

3. Otherwise I'm OK with this. If common Hotel GUI usage works in both web (virtual desktop) and swing clients, then I think this is enough testing. Please do this testing after rebase and report the results here.

#75 - 06/05/2019 01:00 AM - Alexei Kaigorodov

Rebased. See rev 11314 in secure/code/p2j_repo/p2j/active/4078c.

#76 - 06/05/2019 04:17 AM - Greg Shah

You are not following the rebasing process as documented. In the resulting branch, you have lost all the revision history. It seems like you are not rebasing at all. It looks like you are branching trunk and then merging all changes into it as a single revision. This is not correct.

Please follow the exact process as documented. Using the bzr rewrite plugin and the rebase process allows the full revision history to be maintained. This is important to maintain the work in a branch over time. Rebasing happens often. Merging is not the equivalent.

#77 - 06/05/2019 04:24 AM - Greg Shah

As noted in #4078-70:

> Please read the 100% of the Source Code Management document and pay careful attention to the Rebase section.

#78 - 06/05/2019 10:24 AM - Alexei Kaigorodov

Greg Shah wrote:

> You are not following the rebasing process as documented. In the resulting branch, you have lost all the revision history.

I carefully followed Source_Code_Management document, chapter Process for Rebasing.
The history in the branch secure/code/p2j_repo/p2j/active/4078c is exactly as the history of the trunk, plus one revision 11314. Merging all partial commits into single commit is prescribed by the document:
11. the resulting change will go back into the trunk as a single revision, even if you had 50 branch-specific commits

When I prepared branch 3631a, I also added single commit, which contained the content of all the commits made while working on the issue, and you approved it.

#79 - 06/05/2019 10:49 AM - Greg Shah

The rebase process is described in Rebasing and it is an 8 step process.

> 10. eventually, when your changes are ready, you would use "bzr merge" to get the changes from your branch back into the source repo (with all the same conflict resolution and before/after commit requirements)
>
> 11. the resulting change will go back into the trunk as a single revision, even if you had 50 branch-specific commits

The parts you reference are not related to the specific rebasing process. This 11 step description is explaining how we use branching and checkouts in our development process.

Steps 10 and 11 are describing what will be the result when the branch is merged into the trunk. The bzr merge process results in having only a single revision in the trunk that represents all changes from that branch. If you then use bzr log -v -n0 you will actually see all of the branch history as sub-revisions. This is an important artifact that can help us understand how specific changes got into the trunk. It is a complete record of our development process in that branch. We don't want to lose it.

Nothing in steps 10 or 11 are actually things you do during rebase. Likewise steps 1 through 3 are not things you do during a rebase, but instead it just describes the normal development process for a branch. Steps 4 through 9 describe the how we use rebase to change the normal branching process to obtain a specific result. Nothing in this 11 steps is an specific set of instructions so there is no reason to use it as the definition of rebasing.

If you follow the specific 8 step rebasing instructions, you will get the correct result.

> When I prepared branch 3631a, I also added single commit, which contained the content of all the commits made while working on the issue, and you approved it.

I assumed that you had not properly read the Source Management document and that further reading would clarify things. Hynek pointed you to the process for merging/committing back to trunk and you were able to get the changes merged. Since you have repeated the incorrect process here, we have having the discussion to clarify this.

Reply with questions and we will get you any answers you need.

#80 - 06/05/2019 11:07 AM - Alexei Kaigorodov

actuall, I did not do merge. I did rebase, then resolved conflicts, and then made commit. Commit failed because the states of the trunk and 4078c were inconsistent, I do not know why. So I did local commit and then push overwrite.

The command bzr log -v -n0 shows that the history is preserved as subrevisions.

The questions are:
- why do you consider the branch as unappropriate?
- how to fix it?

#81 - 06/05/2019 11:26 AM - Constantin Asofiei

Alexei Kaigorodov wrote:

> actuall, I did not do merge. I did rebase, then resolved conflicts, and then made commit. Commit failed because the states of the trunk and 4078c were inconsistent, I do not know why. So I did local commit and then push overwrite.

You don't need to commit; the rebase commands are these, while in the task branch folder (bzr status must report no uncommitted changes or files):

1. bzr unbind to detach your checkout
2. bzr rebase /path/to/p2j/trunk/local/on/your/machine - make sure your checkout is the latest and no other changes are there, via bzr status
3. fix any conflics and resolve them via bzr resolve <file>, continue via bzr rebase-continue
4. when no more revisions to rebase, you need to do bzr push --overwrite ~/secure/code/p2j_repo/p2j/active/<branchid>/
5. bzr bind to attach the checkout

#82 - 06/05/2019 11:27 AM - Greg Shah

> I did rebase, then resolved conflicts, and then made commit. Commit failed because the states of the trunk and 4078c were inconsistent, I do not know why. So I did local commit and then push overwrite.

When you rebase, please notice that there is no commit step at the end. The commit in step 2 is only if you have pending changes in your checkout of the branch.

The rebasing process is designed to take a checkout (4078c) that was branched at an earlier version of trunk (rev 11311) and make it look like all development in the branch (e.g. multiple commits, in this latest case it was revisions 11312 and 11313) had been done on top of the later trunk revision (rev 11313). The new rebased 4078c would have a rev 11314 (same as old rev 11312) and rev 11315 (same as old rev 11313). In this way, all history is preserved.

After resolving conflicts, you must use bzr rebase-continue, you do not commit. The bzr rewrite tool handles applying the commits for you.

Perhaps this is where went "off track". Re-read the 8 step rebase process. Steps 3 through 7 are the parts that are the iterative steps for rebasing. Steps 1 and 2 are just to make sure you have no uncommitted changes. Step 8 describes how other people that have the old 4078c can update their branch to your rebased version.

> - why do you consider the branch as unappropriate?

Because the revision history is corrupted. For example, there is no evidence of my commit that was 11313 (done yesterday with a message "Fixed 98 char line limit issues. Made 4 char indent into a 3 char indent.") in the original branch. Look carefully at the output of bzr log -v -n0 and you will see it is gone.

Following the process exactly will preserve this history. There is no merging during the rebase process. During rebase the rewrite tool removes all the revisions and applies them on top of the newer revision of trunk. The result is very different from a merge.

> - how to fix it?

For this task it isn't that important. But in future tasks I want it done properly from the beginning, so it is important that you have a correct understanding.

#83 - 06/05/2019 11:43 AM - Alexei Kaigorodov

Greg Shah wrote:

> Following the process exactly will preserve this history. There is no merging during the rebase process. During rebase the rewrite tool removes all the revisions and applies them on top of the newer revision of trunk.

the revisions moved on top: should they be in form of single revision, or seen as distinct revisions as before?

#84 - 06/05/2019 11:44 AM - Greg Shah

> the revisions moved on top: should they be in form of single revision, or seen as distinct revisions as before?

They are distinct revisions as before.

#85 - 06/05/2019 01:04 PM - Alexei Kaigorodov

Greg Shah wrote:

> > the revisions moved on top: should they be in form of single revision, or seen as distinct revisions as before?
>
> They are distinct revisions as before.

I asked whether the should be seen as single complex revision, or seen as distinct top-level revisions. Now there are 2 top-level revisions:
revno: 11315

author: Greg Shah <ges@goldencode.com>
committer: Alexei Kaigorodov <avk@goldencode.com>
branch nick: 4078c
timestamp: Tue 2019-06-04 13:18:57 ~~0400~~
~~message:~~
~~Fixed 98 char line limit issues. Made 4 char indent into a 3 char indent.~~

```
revno: 11314 [merge]
committer: Alexei Kaigorodov <avk@goldencode.com>
branch nick: p2j
timestamp: Wed 2019-06-05 11:50:19 +0700
message:
Bug 4078 fixed according to the directions made by Hynek Cihlar.
------------------------------------------------------------
```

the 11314 is complex revision. The command bzr log -v -n0 -r 11314 shows:

```
------------------------------------------------------------
revno: 11314 [merge]
committer: Alexei Kaigorodov <avk@goldencode.com>
branch nick: p2j
timestamp: Wed 2019-06-05 11:50:19 +0700
message:
Bug 4078 fixed according to the directions made by Hynek Cihlar.
modified:
src/com/goldencode/p2j/ui/ButtonConfig.java*
src/com/goldencode/p2j/ui/client/gui/driver/swing/SwingEmulatedWindow.java
------------------------------------------------------------
revno: 11311.2.1 [merge]
committer: Alexei Kaigorodov <avk@goldencode.com>
branch nick: p2j
timestamp: Mon 2019-06-03 11:22:30 +0700
message:
Bug 4078 fixed according to the directions made by Hynek Cihlar.
modified:
src/com/goldencode/p2j/ui/ButtonConfig.java
src/com/goldencode/p2j/ui/client/gui/driver/swing/SwingEmulatedWindow.java
------------------------------------------------------------
revno: 11310.2.1
committer: Alexei Kaigorodov <avk@goldencode.com>
branch nick: 4078c
timestamp: Mon 2019-05-27 16:50:55 +0700
message:
Bug 4078 fixed according to the directions made by Hynek Cihlar.
modified:
src/com/goldencode/p2j/ui/ButtonConfig.java
src/com/goldencode/p2j/ui/client/gui/driver/swing/SwingEmulatedWindow.java
------------------------------------------------------------
```

the question is: should 11315 be included in 11314; or should 11314 be unpacked so that its internal subversions transformed to distinct top level revisions?

#86 - 06/05/2019 02:14 PM - Greg Shah

> the question is: should 11315 be included in 11314; or should 11314 be unpacked so that its internal subversions transformed to distinct top level revisions?

11315 should not be included in 11314.

If it had originally been rebased instead of merged, then 11314 would have been separate revisions. In that case, each one would still be separate in the final rebased result. But since the rebase happened on top of a single 11312 revision, then we would expect a single 11314 revision that matches the 11312 in the pre-rebase branch.

#87 - 06/05/2019 02:55 PM - Greg Shah

> If common Hotel GUI usage works in both web (virtual desktop) and swing clients, then I think this is enough testing. Please do this testing after rebase and report the results here.

Please post the results of your testing here.

#88 - 06/07/2019 03:35 AM - Alexei Kaigorodov

- File screen.png added

it runs successfully in swing client.
I could not run it in web mode. Server.log has no errors, but the screen shows "Cannot start the embedded server (see attached file screen.png).

#89 - 06/07/2019 06:22 AM - Greg Shah

Alexei Kaigorodov wrote:

> it runs successfully in swing client.
> I could not run it in web mode. Server.log has no errors, but the screen shows "Cannot start the embedded server (see attached file screen.png).

It is important that you resolve this web client issue so that we know the change is safe. It is probably something broken in your environment/cfg but until you resolve it we won't know.

#90 - 06/09/2019 11:27 PM - Alexei Kaigorodov

Now it runs successfully both in swing and virtual desktop clients.

#91 - 06/10/2019 01:34 PM - Greg Shah

Please merge 4078c to trunk.

#92 - 06/11/2019 10:46 AM - Alexei Kaigorodov

4078c was merged to trunk rev 11314 and was archived.

#93 - 06/11/2019 11:18 AM - Greg Shah

- Status changed from WIP to Closed

- Start date deleted (05/09/2019)

- % Done changed from 0 to 100

Files

4078_workaround.diff	515 Bytes	05/10/2019	Eric Faulhaber
stack.txt	1001 Bytes	05/14/2019	Alexei Kaigorodov
data.txt	7.11 KB	05/14/2019	Alexei Kaigorodov
serverstack.txt	7.17 KB	05/16/2019	Alexei Kaigorodov
clientstack.txt	10.9 KB	05/16/2019	Alexei Kaigorodov
c11300.txt	54.2 KB	05/22/2019	Alexei Kaigorodov
patch11311.txt	811 Bytes	05/23/2019	Alexei Kaigorodov
test.png	132 KB	05/24/2019	Alexei Kaigorodov
screen.png	171 KB	06/07/2019	Alexei Kaigorodov