

## User Interface - Bug #3075

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

### The window icon popup menu isn't displayed if a user clicks on the window icon by the left mouse button

04/21/2016 01:43 AM - Sergey Ivanovskiy

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Sergey Ivanovskiy	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Cleanup and Stabilization for GUI	<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Related issues:</b>			
Related to User Interface - Bug #2988: The web click events on the caption bu...			<b>Closed</b>

### History

#### #1 - 04/21/2016 01:50 AM - Sergey Ivanovskiy

It is probably not a bug, but why the previous version implements a left mouse popup. The implementation is changed if we compare it with the previous 1811u implementation.

#### #2 - 05/06/2016 01:14 AM - Sergey Ivanovskiy

The issue is fixed in #3080 (task branch 3084a).

#### #3 - 05/11/2016 04:42 PM - Greg Shah

The changes are fine with me.

Constantin: any concerns?

#### #4 - 05/12/2016 05:31 AM - Constantin Asofiei

Sergey, about 3084a rev 11029:

1. for the title-bar popup: for embedded mode, no decorations are shown, so there will be no title-bar popup. WindowGuiImpl.init needs to check this. Also, you are not updating the windowTitlePopup field... so the next call of TopLevelWindow.initWindowTitlePopup will re-create it. Also, the same issue needs to be fixed for modal dialogs.
2. GuiWebSocket.graphicsCached is missing javadoc and it needs a new-line under it.
3. GuiWebSocket c'tor - please right-align the parameters.
4. p2j.screen.js - H014 text needs to be removed (is part of H013)
5. ScrollableSelectionListGuiImpl - the changes affect the SELECTION-LIST widget - please check if this is OK.

#### #5 - 05/13/2016 05:45 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey, about 3084a rev 11029:

1. for the title-bar popup: for embedded mode, no decorations are shown, so there will be no title-bar popup. WindowGuilImpl.init needs to check this. Also, you are not updating the windowTitlePopup field... so the next call of TopLevelWindow.initWindowTitlePopup will re-create it. Also, the same issue needs to be fixed for modal dialogs.

It seems that this code from init() in the embedded mode works properly. If the window doesn't have a title bar, then the window title popup isn't created

```
// can be missing if there are no decorations
if (hasTitlebar())
{
    contentPane.add(titleBar);
    windowTitlePopup = createWindowTitlePopup(this);
}
```

Please explain what is incorrect? Do you mean to use (EmbeddedClient (gd)).isEmbeddedClient().

#### #6 - 05/13/2016 07:29 AM - Sergey Ivanovskiy

- File sel\_list\_test0.png added
- File 3075\_4.txt added
- File 3075\_3.txt added

Committed revision 11032 fixes the creation of window popup for a modal window. I have checked SELECTION-LIST widget, there are no regressions here. It seems that P2J trunc draws the SELECTION-LIST border incorrectly and doesn't have a mouse selection for this test ./selection\_list/sel\_list\_test0.p. Please look at the attached screen of the Swing client for this example.

#### #7 - 05/13/2016 07:37 AM - Sergey Ivanovskiy

./simple\_alert\_box.p com.goldencode.testcases.SimpleAlertBox.execute The web client issue is if we select "Close" menu item from the window title popup menu for modal window dialog, then the owner window becomes hidden, but the modal window is still visible and task bar has no visible task icons. I think this behavior is incorrect. Also if we select "Close" menu item from the window title bar of the active non modal window, then this window disappears from the screen.

#### #8 - 06/01/2016 08:51 AM - Greg Shah

Committed revision 11032 fixes the creation of window popup for a modal window. I have checked SELECTION-LIST widget, there are no regressions here. It seems that P2J trunc draws the SELECTION-LIST border incorrectly and doesn't have a mouse selection for this test ./selection\_list/sel\_list\_test0.p. Please look at the attached screen of the Swing client for this example.

The border issue will be resolved in [#2741](#). The mouse issue will be resolved in [#2879](#). Don't work the selection list issues further in this task.

**#9 - 06/01/2016 03:22 PM - Constantin Asofiei**

Sergey Ivanovskiy wrote:

`./simple_alert_box.p com.goldencode.testcases.SimpleAlertBox.execute` The web client issue is if we select "Close" menu item from the window title popup menu for modal window dialog, then the owner window becomes hidden, but the modal window is still visible and task bar has no visible task icons. I think this behavior is incorrect. Also if we select "Close" menu item from the window title bar of the active non modal window, then this window disappears from the screen.

We need to check how 4GL behaves in this case, first. Please create a separate task for it.

**#10 - 06/10/2016 05:48 AM - Sergey Ivanovskiy**

After applying changes of 11042 trunc revision all changes for this task are vanished and mean nothing. Now LeftMousePopup and RightMousePopup don't work for the web client. Continue this task to fix the issue. The branch for this task is 3084a (11072 rev.)

**#11 - 06/10/2016 08:10 AM - Sergey Ivanovskiy**

To work around this issue I am planning to register new fake actions, SystemAction, for the window title bar and its window icon with MOUSE\_POPUPABLE\_RIGHT and MOUSE\_POPUPABLE\_LEFT respectively.

**#12 - 06/10/2016 08:36 AM - Greg Shah**

Please clear this issue in 3084a and I'd like to get that tested and merged to trunk ASAP.

**#13 - 06/10/2016 10:15 AM - Sergey Ivanovskiy**

Committed revision 11073 fixed mouse window title popup for the Web client. Please review the changes.

**#14 - 06/10/2016 02:45 PM - Sergey Ivanovskiy**

Greg, what are tests that you are planning to be performed before it can be merged to trunc? Manual GUI testing?

**#15 - 06/10/2016 03:19 PM - Greg Shah**

Code Review Task Branch 3084a Revision 11073

The changes are fine.

**#16 - 06/10/2016 03:21 PM - Greg Shah**

Sergey Ivanovskiy wrote:

Greg, what are tests that you are planning to be performed before it can be merged to trunc? Manual GUI testing?

Although the changes are mostly affecting the web-client, there are some changes to both the conversion and to some common code for the UI.

To be safe, do the following:

1. Run conversion testing on devsrv01. It should generate no changes.
2. Start the MAJIC server and login to MAJIC. Try some simple menu navigation and open a screen or two. As long as that works, the common code changes are OK.
3. Manual GUI testing.

#### #17 - 06/10/2016 05:16 PM - Sergey Ivanovskiy

Starting the conversion testing on devsrv01 according to handbook/timco.html#comparison\_version.

1. ~/projects/p2j (from trunc) and ~/projects/3084a (from 3084a) were updated to the their up-to-date versions
2. ~/testing/majic was updated to the current version (staging branch) ~/testing/majic/p2j is pointed to ~/projects/p2j
3. cd ~/testing and run\_regression.sh build save-generated -Dsave.candidate=y  
Next planning to change a p2j link to be pointed to ~/projects/3084a and run cd ~/testing and run\_regression.sh build save-generated -Dsave.candidate=y  
Finally, execute run\_regression.sh conv-regression -Dsave.candidate=y -Dsrc.stable=generated\_for\_p2j -Dsrc.candidate=generated\_for\_3084a, where  
generated\_for\_p2j is a path to the first conversion run located in ~/testing/generated/YYYYMMDDa and generated\_for\_3084a is a path to the second conversion run located in ~/testing/generated/YYYYMMDDb

#### #18 - 06/12/2016 03:13 AM - Sergey Ivanovskiy

There are no differences in the conversion sources for the Majic. Planning to rebase and do manual GUI testing later this evening.

#### #19 - 06/12/2016 04:11 PM - Sergey Ivanovskiy

3084a rev 11074 is rebased on the current trunc rev 11043.

#### #20 - 06/13/2016 03:33 AM - Sergey Ivanovskiy

Committed revision 11075 fixed the modal window case if the window icon wouldn't present.

#### #21 - 06/13/2016 09:54 AM - Sergey Ivanovskiy

The window minimize action doesn't work correctly from the system menu for the web client. The menu overlay window is hidden after the minimize action has been executed on the java side (set the window minimized state and post WindowEvent(window, Keyboard.SE\_WINDOW\_MINIMIZED)), then it causes its owner to become the top visible window sending MSG\_MOVE\_TO\_TOP message. Finally, the posted event WindowEvent(window, Keyboard.SE\_WINDOW\_MINIMIZED) is processed and the target window is set hidden by sending MSG\_WINDOW\_VISIBILITY. This sequence causes a problem to handle a minimize action correctly. Committed revision 11076 fixed this issue. For the trunc it can not be reproduced since the system menu is not accessible for the web client.

P2J has StringIndexOutOfBoundsException for the "Select All" editor action.

```
Caused by: java.lang.StringIndexOutOfBoundsException: String index out of range: 1
    at java.lang.String.substring(String.java:1963)
    at com.goldencode.p2j.ui.client.gui.FillInGuiImpl.drawSelectionInt(FillInGuiImpl.java:1048)
    at com.goldencode.p2j.ui.client.gui.FillInGuiImpl.drawSelection(FillInGuiImpl.java:1025)
    at com.goldencode.p2j.ui.client.gui.FillInGuiImpl$2.run(FillInGuiImpl.java:895)
    at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
    at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
    at com.goldencode.p2j.ui.client.gui.FillInGuiImpl$2.run(FillInGuiImpl.java:883)
    at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
    at com.goldencode.p2j.ui.client.gui.FillInGuiImpl.draw(FillInGuiImpl.java:830)
    at com.goldencode.p2j.ui.client.FillIn.draw(FillIn.java:1077)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.draw(AbstractContainer.java:427)
    at com.goldencode.p2j.ui.client.widget.Viewport.draw(Viewport.java:64)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.draw(AbstractContainer.java:427)
    at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl.access$501(BorderedPanelGuiImpl.java:44)
    at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl$1$1.run(BorderedPanelGuiImpl.java:246)
    at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
```

```

at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl$1.run(BorderedPanelGuiImpl.java:235)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl.draw(BorderedPanelGuiImpl.java:182)
at com.goldencode.p2j.ui.client.gui.ScrollPaneGuiImpl$2.run(ScrollPaneGuiImpl.java:202)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.ScrollPaneGuiImpl.draw(ScrollPaneGuiImpl.java:193)
at com.goldencode.p2j.ui.client.gui.FrameGuiImpl$2.run(FrameGuiImpl.java:483)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.FrameGuiImpl.draw(FrameGuiImpl.java:469)
at com.goldencode.p2j.ui.client.Frame.draw(Frame.java:1998)
at com.goldencode.p2j.ui.client.widget.AbstractContainer.draw(AbstractContainer.java:427)
at com.goldencode.p2j.ui.client.widget.Viewport.draw(Viewport.java:64)
at com.goldencode.p2j.ui.client.widget.AbstractContainer.draw(AbstractContainer.java:427)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl.access$501(BorderedPanelGuiImpl.java:44)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl$1$1.run(BorderedPanelGuiImpl.java:246)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl$1.run(BorderedPanelGuiImpl.java:235)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl.draw(BorderedPanelGuiImpl.java:182)
at com.goldencode.p2j.ui.client.gui.ScrollPaneGuiImpl$2.run(ScrollPaneGuiImpl.java:202)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.ScrollPaneGuiImpl.draw(ScrollPaneGuiImpl.java:193)
at com.goldencode.p2j.ui.client.gui.WindowWorkSpace.draw(WindowWorkSpace.java:87)
at com.goldencode.p2j.ui.client.widget.AbstractContainer.draw(AbstractContainer.java:427)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl.access$501(BorderedPanelGuiImpl.java:44)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl$1$1.run(BorderedPanelGuiImpl.java:246)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl$1.run(BorderedPanelGuiImpl.java:235)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2160)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.draw(AbstractGuiDriver.java:2037)
at com.goldencode.p2j.ui.client.gui.BorderedPanelGuiImpl.draw(BorderedPanelGuiImpl.java:182)
at com.goldencode.p2j.ui.client.Window.draw(Window.java:1375)
at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.draw(WindowGuiImpl.java:526)
at com.goldencode.p2j.ui.client.OutputManager.setInvalidate(OutputManager.java:1275)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:14284)
at com.goldencode.p2j.ui.chui.ThinClient.applyWorker(ThinClient.java:14033)
at com.goldencode.p2j.ui.chui.ThinClient.waitForEvent(ThinClient.java:13360)
at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:11246)
at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:10825)
at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:10773)
at com.goldencode.p2j.ui.chui.ThinClient.promptFor(ThinClient.java:9506)

```

It can be reproduced for the Swing client with ./ask-gui.p.

**#22 - 06/13/2016 10:53 AM - Greg Shah**

Code Review Task Branch 3084a Revision 11076

1. In WindowTitlePopupGuiImpl.create(), why do we only call registerWidgetAction() for minimize and not for the other menu items?
2. In p2j.mouse.js, why don't we need code for maximize in processOSSystemAction()?

**#23 - 06/13/2016 10:54 AM - Greg Shah**

P2J has StringIndexOutOfBoundsException for the "Select All" editor action.

If you see a quick solution, go ahead and fix it now.

Otherwise, please create a new task.

**#24 - 06/13/2016 12:13 PM - Sergey Ivanovskiy**

Greg Shah wrote:

P2J has StringIndexOutOfBoundsException for the "Select All" editor action.

If you see a quick solution, go ahead and fix it now.

Otherwise, please create a new task.

Ok, I found that the paste menu action is broken now for browsers. Planning to fix it.

**#25 - 06/13/2016 12:49 PM - Sergey Ivanovskiy**

The following case doesn't work for the web client

- 1) Open the Editor's popup and click on "Select All" menu item
- 2) Open the Editor's popup again and click on the "Copy" or "Cut" menu item.
- 3) The selected text must be in the system clipboard.

But if the selection is done with help of the keyboard, then the scenario described above is passed.

**#26 - 06/13/2016 01:10 PM - Sergey Ivanovskiy**

The issue is related to the selection issue. The copy/cut/paste functionalities can work correctly only if the user's selected text is send to the web client. If we execute "Select All" action, the current selection is unknown to the Web client.

**#27 - 06/13/2016 01:15 PM - Sergey Ivanovskiy**

Code Review Task Branch 3084a Revision 11076

1. In WindowTitlePopupGuiImpl.create(), why do we only call registerWidgetAction() for minimize and not for the other menu items?
2. In p2j.mouse.js, why don't we need code for maximize in processOSSystemAction()?

Because this action can be done on the java side and then the web client updates its screen receiving MSG\_DRAW. Otherwise minimize action requires to work with task bar.

**#28 - 06/13/2016 01:37 PM - Sergey Ivanovskiy**

Committed revision 11077 has minor fixes: StringIndexOutOfBoundsException (note 23) and select all and copy scenario (note 25).

**#29 - 06/13/2016 05:19 PM - Greg Shah**

Code Review Task Branch 3084a Revision 11077

The changes look fine.

Constantin: is the EditorPopupGuiImpl change always safe?

**#30 - 06/13/2016 05:20 PM - Greg Shah**

Sergey: please rebase and restart testing.

**#31 - 06/14/2016 03:27 AM - Sergey Ivanovskiy**

Committed revision is 11078 (rebase). SystemAction registers a widget action that is known to the web client too. It is used to intercept the required copy, cut, paste and minimize actions to be preprocessed or executed by the web client. Ideally, it can be used to split logic processing required for the Swing client and the Web client. For an example, we can process an action on the java side for the Swing client only and can process an action on the Web client only. We can add SystemAction for the maximize and close actions, but the web client doesn't process them now, because they are executed by the java side.

**#32 - 06/14/2016 07:44 AM - Greg Shah**

Committed revision is 11078 (rebase). SystemAction registers a widget action that is known to the web client too. It is used to intercept the required copy, cut, paste and minimize actions to be preprocessed or executed by the web client. Ideally, it can be used to split logic processing required for the Swing client and the Web client. For an example, we can process an action on the java side for the Swing client only and can process an action on the Web client only. We can add SystemAction for the maximize and close actions, but the web client doesn't process them

now, because they are executed by the java side.

This is a very useful explanation. It should be placed in javadoc for SystemAction to help future readers.

**#33 - 06/14/2016 08:02 AM - Sergey Ivanovskiy**

Committed revision 11079 updated the java doc for SystemAction.

**#34 - 06/14/2016 08:31 AM - Constantin Asofiei**

Sergey, you are pushing the selected text from the widget to the driver. Is this always correct? What if the user selects the text in widget W1, switches to widget W2 (while the text in W1 remains selected) and presses CTRL-C to copy (and W2 doesn't have any selected text, or maybe even it doesn't support copy)? Wouldn't the driver see the text from W1, instead of seeing "no selection" text, as W2 doesn't have any selected text? Another issue is that we are pushing the selected text to the driver always, even if the selection hasn't yet finished or even if the user doesn't plan to copy it (maybe just wants to delete it). Do you think is possible to let the driver interrogate the P2J client, to find the focused widget and get the selected text from that, only when COPY/CUT action is performed?

Also, in EditorGuilImpl, in the current approach you are missing the invalidateSelection() part, where the config.selectionText is set to null.

**#35 - 06/14/2016 08:54 AM - Sergey Ivanovskiy**

Constantin Asofiei wrote:

Sergey, you are pushing the selected text from the widget to the driver. Is this always correct? What if the user selects the text in widget W1, switches to widget W2 (while the text in W1 remains selected) and presses CTRL-C to copy (and W2 doesn't have any selected text, or maybe even it doesn't support copy)? Wouldn't the driver see the text from W1, instead of seeing "no selection" text, as W2 doesn't have any selected text?

I think that CTRL+C is used to copy the selection to the system clipboard. If a user selects the text in W1 and presses CTRL+C, then this text must be in the system clipboard. It is useful from the user's perspective. Please explain what is the correct behavior in this case.

Another issue is that we are pushing the selected text to the driver always, even if the selection hasn't yet finished or even if the user doesn't plan to copy it (maybe just wants to delete it). Do you think is possible to let the driver interrogate the P2J client, to find the focused widget and get the selected text from that, only when COPY/CUT action is performed?

Since I didn't find the way to get the selected text at the moment of the COPY/CUT action (may be it is not possible from the JS client), I used the approach with the current selection. Please explain the desired behavior.

Also, in EditorGuilImpl, in the current approach you are missing the invalidateSelection() part, where the config.selectionText is set to null.

Ok, thanks, I will try to fix it.

**#36 - 06/14/2016 09:15 AM - Sergey Ivanovskiy**

Constantin Asofiei wrote:

Also, in EditorGuilImpl, in the current approach you are missing the invalidateSelection() part, where the config.selectionText is set to null.

At first it seems clear for me but looking at the code confuses me. What do you mean here? Please explain it more thoroughly.

**#37 - 06/14/2016 11:19 AM - Constantin Asofiei**

Sergey Ivanovskiy wrote:

Constantin Asofiei wrote:

Also, in EditorGuilImpl, in the current approach you are missing the invalidateSelection() part, where the config.selectionText is set to null.

At first it seems clear for me but looking at the code confuses me. What do you mean here? Please explain it more thoroughly.

Well, the problem is this: lets say you have two editors, E1 and E2. E1 has content "abcde" , you select it (so "abcde" is pushed to the web driver) and move to E2 (which doesn't have any content). Now, you press CTRL-C while in E2: this shouldn't copy anything, as E2 doesn't have content - but your changes do copy the previously selected content "abcde" from E1, as this is what was pushed to the web client, and never invalidated. Once the selection is gone, you need to let the web client know that there is no longer a selection. You need to do this for fill-in and editor. The idea is: the selected text known by the web client's "prepareCopy" must always match the selected text in the currently focused widget, on the screen.

Now, the second part: there is this comment in p2j.clipboard.js prepareCopy:

```
// TODO: this is broken for GUI CUT/COPY, this will send a request for the selected text
//       to the server, but the text will be sent back asynchronously and we have no way
//       to block this code and wait for the results, thus this text will always be
//       wrong until we find a way around this issue
```

I think now I understand why you choose the "push" instead of the "pull" approach, to prepare the copy (i.e. get the selected text). My idea was to try to avoid unneeded round trips to the client, as they are expensive. If there is no good way to use a "pull" approach (i.e. the web driver gets the selected text as needed), then that's OK.

**#38 - 06/14/2016 11:47 AM - Sergey Ivanovskiy**

Is it ok if we send the shrunken MSG\_CURRENT\_SELECTION message without its content to indicate that the current selection is cleared and the web client doesn't copy or cut the empty string and simply does nothing without changing the system clipboard?

**#39 - 06/14/2016 11:49 AM - Constantin Asofiei**

Sergey Ivanovskiy wrote:

Is it ok if we send the shrunken MSG\_CURRENT\_SELECTION message without its content to indicate that the current selection is cleared and the web client doesn't copy or cut the empty string and simply does nothing without changing the system clipboard?

Yes, I think is OK to use "empty string" to mean no selection, on the web-client JS side.

**#40 - 06/14/2016 12:47 PM - Sergey Ivanovskiy**

- File 3075\_7.txt added

**#41 - 06/14/2016 01:21 PM - Sergey Ivanovskiy**

Thanks, please review the committed revision 11080. It should fix the issue.

**#42 - 06/14/2016 07:31 PM - Greg Shah**

Code Review Task Branch 3084a Revision 11080

My only concern is that in p2j.clipboard.js, the prepareCopy() will not call setClipboardValue() if text is the empty string. Doesn't this mean that the current selection should be reset? But in fact, the inputText.value will retain the current content. Can't we then accidentally copy this later when the selection really should be empty?

**#43 - 06/14/2016 10:15 PM - Constantin Asofiei**

Sergey, 11080 fixes the issue, but I don't understand the change in prepareCopy mentioned by Greg:

```
if (text !== "")
{
    setClipboardValue(text);
}
```

Why is this needed? For the case when CTRL-C is pressed with no selection, the copyHandler already consumes the event:

```
if (getSelection() === "")
{
    p2j.consumeEvent(e);
    return;
}
```

Also, the Copy option in the right-click popup-menu will be disabled if there is no selection.

#### #44 - 06/15/2016 01:09 AM - Sergey Ivanovskiy

Greg Shah wrote:

Code Review Task Branch 3084a Revision 11080

My only concern is that in p2j.clipboard.js, the prepareCopy() will not call setClipboardValue() if text is the empty string. Doesn't this mean that the current selection should be reset? But in fact, the inputText.value will retain the current content. Can't we then accidentally copy this later when the selection really should be empty?

The current selection is a changeable value that is hold by var userSelection; p2j.clipboard-module's field, otherwise methods setClipboardValue and getClipboardValue sets and returns the system clipboard value. Since in this case the clipboard value is not changed, we don't set the clipboard value to the empty string.

#### #45 - 06/15/2016 01:14 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey, 11080 fixes the issue, but I don't understand the change in prepareCopy mentioned by Greg:  
[...]

Why is this needed? For the case when CTRL-C is pressed with no selection, the copyHandler already consumes the event:  
[...]

Also, the Copy option in the right-click popup-menu will be disabled if there is no selection.

This statement is required because a user can press CTRL+C in the case if the current selection is not set. Please look at the p2j.keyboard module, var sendKey = function(key, evt). The keyboard logic permits CTRL+C/CTRL+X, but the generated event "copy/cut" doesn't change the system clipboard value, since the current selection is invalid.

#### #46 - 06/15/2016 01:44 AM - Sergey Ivanovskiy

I guess the next question can be that why we invoke prepareCopy() here

```
if (p2j.isGui)
{
  // consume copy/cut events if the current selection is invalid
  if (getSelection() === "")
  {
```

```
        p2j.consumeEvent(e);
        return;
    }
    prepareCopy();
}
```

It is the Safari's use case. The CTRL+C is processed on keyup, but in the Safari use case the event copy happens first followed by keyup event.

**#47 - 06/15/2016 02:36 AM - Sergey Ivanovskiy**

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

Is it ok if we send the shrunken MSG\_CURRENT\_SELECTION message without its content to indicate that the current selection is cleared and the web client doesn't copy or cut the empty string and simply does nothing without changing the system clipboard?

Yes, I think is OK to use "empty string" to mean no selection, on the web-client JS side.

We have a problem here because now we can't write an empty string to the system clipboard. Since an invalid selection is defined by "" empty string. I would like to change to the null value and to send the shrunken MSG\_CURRENT\_SELECTION message without its content to indicate that the current selection is cleared (invalid) or to send new MSG\_SELECTION\_INVALID.

**#48 - 06/15/2016 02:43 AM - Sergey Ivanovskiy**

Greg, Constantin please advise if it is required to be able to set an empty string to the system clipboard by CLIPBOARD:VALUE="".

**#49 - 06/15/2016 03:07 AM - Sergey Ivanovskiy**

- File 3075\_8.txt added

Please review the committed revision 11081 (3084a). The code is changed due to notes 47 and 48. Added MSG\_INVALIDATE\_SELECTION.

**#50 - 06/15/2016 04:09 AM - Sergey Ivanovskiy**

Committed revision 11082 (cleanup) utilizes that (undefined == null) is true in JS.

**#51 - 06/15/2016 07:57 AM - Greg Shah**

Greg, Constantin please advise if it is required to be able to set an empty string to the system clipboard by CLIPBOARD:VALUE="".

In the 4GL, this works, right? So it must have the same result for P2J too.

**#52 - 06/15/2016 08:05 AM - Greg Shah**

Code Review Task Branch 3084a Revision 11082

The changes are good.

**#53 - 06/15/2016 04:21 PM - Greg Shah**

- Status changed from New to Closed
- Start date deleted (04/21/2016)
- % Done changed from 0 to 100
- Assignee set to Sergey Ivanovskiy
- Target version set to Milestone 16

**#54 - 06/15/2016 04:24 PM - Greg Shah**

Task branch 3084a was merged to trunk as revision 11045 and that revision contains the changes that resolve this task.

**#55 - 11/16/2016 12:22 PM - Greg Shah**

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

**Files**

---

3075_3.txt	2.81 KB	05/13/2016	Sergey Ivanovskiy
3075_4.txt	979 Bytes	05/13/2016	Sergey Ivanovskiy
sel_list_test0.png	4.87 KB	05/13/2016	Sergey Ivanovskiy
3075_7.txt	4.2 KB	06/14/2016	Sergey Ivanovskiy
3075_8.txt	6.19 KB	06/15/2016	Sergey Ivanovskiy