

## User Interface - Bug #2772

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

### the ask-gui.p testcase doesn't show the message update dialog.

10/21/2015 10:41 AM - Sergey Ivanovskiy

|                        |                                    |                        |            |
|------------------------|------------------------------------|------------------------|------------|
| <b>Status:</b>         | Closed                             | <b>Start date:</b>     | 10/21/2015 |
| <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> | GUI Support for a Complex ADM2 App | <b>case_num:</b>       |            |
| <b>billable:</b>       | No                                 |                        |            |
| <b>vendor_id:</b>      | GCD                                |                        |            |
| <b>Description</b>     |                                    |                        |            |

### History

#### #1 - 10/21/2015 10:47 AM - Sergey Ivanovskiy

The JS web client doesn't display "Try another program?: yes/No" dialog and the browser console produces

```
17:43:56.911 Window ids must be non-positive integers.1 p2j.screen.js:1840:7
me.error() p2j.screen.js:1840
me.createWindow() p2j.screen.js:1524
me.createChildWindow() p2j.screen.js:1574
me.init/ws.onmessage() p2j.socket.js:440
```

#### #2 - 10/21/2015 10:49 AM - Greg Shah

Please note that on the Java side this seems to be related to the use of `Window.tinyInput()`. I think this is the same cause of the problem in `message-update6.p`.

#### #3 - 10/21/2015 11:28 AM - Sergey Ivanovskiy

Please note that on the Java side this seems to be related to the use of `Window.tinyInput()`. I think this is the same cause of the problem in `message-update6.p`.

The JS client expects the child window id is a positive number. But the `ModalWindow` can have a unique id from the negative range of the signed integer type.

#### #4 - 10/21/2015 11:51 AM - Sergey Ivanovskiy

- File MessageUpdate.png added

If we permit negative numbers for window ids, then the message update dialog appears and looks like in the attached screen with a broken layout. Constantin, do you have any objections to accept negative window ids? It needs only to change

```
function isValidWindowId(wid)
{
    return (typeof wid === 'number') /*&& (wid % 1 === 0) && wid > 0*/;
};
```

#### #5 - 10/21/2015 11:59 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

If we permit negative numbers for window ids, then the message update dialog appears and looks like in the attached screen with a broken layout.

Constantin, do you have any objections to accept negative window ids? It needs only to change

[...]

I'm OK with the change. Also, update the javadoc for the function, too.

#### #6 - 10/21/2015 04:55 PM - Sergey Ivanovskiy

Checked that the icon if it would be defined for ModalWindow

```
public ModalWindow(String title)
{
    super(title);

    id = WidgetId.nextID();
    gd = (GuiDriver<?,?>) OutputManager.getDriver();

    // the modal window is parented to the active top-level window, not the focus owner
    setOwner(WindowManager.getActiveTopLevelWindow());

    setLayout(new WindowLayout());

    titleBar = new WindowTitleBar(WidgetId.nextID(),
        true, // tested the icon coordinates
        new CaptionButtonType[] {CaptionButtonType.CLOSE},
        title);
    .....
}
```

were located correctly within ModalWindow dialog, but the dialog title and its background displayed lower. The web log probably had correct coordinates to draw the title. Thus WindowTitleBar of ModalWindow works properly for icon but not for the title.

```
PaintPrimitives.CLIP 3; x = 0; y = 0; width = 17; height = 17
PaintPrimitives.DRAW_IMAGE x = 0; y = 0; width = 16; height = 16
PaintPrimitives.NO_CLIP 2
PaintPrimitives.TRANSLATE_POP 1; x = -2; y = -1
PaintPrimitives.TRANSLATE_PUSH 2; x = 19; y = 1
PaintPrimitives.CLIP 3; x = 0; y = 0; width = 95; height = 19
PaintPrimitives.SET_COLOR r = 10; g = 36; b = 106
PaintPrimitives.FILL_RECT x = 0; y = 0; width = 94; height = 17
PaintPrimitives.SET_COLOR r = 255; g = 255; b = 255
PaintPrimitives.SET_FONT
PaintPrimitives.DRAW_STRING text = Message Update; x = 0; y = 18; centered = true
```

It is unclear for me what happens.

#### #7 - 10/21/2015 05:40 PM - Greg Shah

Have you compared the JS logs to the Swing logs? Since Swing is working, you should be able to see which operation gets the wrong coordinates.

#### #8 - 10/22/2015 03:12 AM - Sergey Ivanovskiy

Have you compared the JS logs to the Swing logs? Since Swing is working, you should be able to see which operation gets the wrong coordinates.

There is a definite issue with resize sequences in the web java log and the JS client log. They are different and for JS side there is interleaves. For example, this JS drawing circle don't match the web java log. It needs to investigate what happens.

```
START DRAWING CYCLE FOR WINDOW -44
PaintPrimitives.CLIP 1; x = 0; y = 0; width = 141; height = 48
PaintPrimitives.SET_COLOR r = 212; g = 208; b = 200
PaintPrimitives.FILL_RECT x = 0; y = 0; width = 141; height = 48
PaintPrimitives.TRANSLATE_PUSH 1; x = 4; y = 4
PaintPrimitives.CLIP 2; x = 0; y = 0; width = 134; height = 20
PaintPrimitives.SET_COLOR r = 128; g = 128; b = 128
PaintPrimitives.FILL_RECT x = 0; y = 0; width = 133; height = 18
PaintPrimitives.SET_ICON x = 0; y = 0; width = 16; height = 16
PaintPrimitives.TRANSLATE_PUSH 2; x = 2; y = 1
PaintPrimitives.CLIP 3; x = 0; y = 0; width = 17; height = 17
PaintPrimitives.DRAW_IMAGE x = 0; y = 0; width = 16; height = 16
PaintPrimitives.NO_CLIP 2
PaintPrimitives.TRANSLATE_POP 1; x = -2; y = -1
PaintPrimitives.SET_TITLE title = Message Update
PaintPrimitives.TRANSLATE_PUSH 2; x = 19; y = 1
PaintPrimitives.CLIP 3; x = 0; y = 0; width = 95; height = 19
PaintPrimitives.SET_COLOR r = 128; g = 128; b = 128
PaintPrimitives.FILL_RECT x = 0; y = 0; width = 94; height = 17
PaintPrimitives.SET_COLOR r = 255; g = 255; b = 255
PaintPrimitives.SET_FONT
PaintPrimitives.DRAW_STRING text = Message Update; x = 0; y = 18; centered = true
PaintPrimitives.NO_CLIP 2
PaintPrimitives.TRANSLATE_POP 1; x = -19; y = -1
PaintPrimitives.TRANSLATE_PUSH 2; x = 115; y = 2
PaintPrimitives.CLIP 3; x = 0; y = 0; width = 17; height = 15
```

PaintPrimitives.SET\_COLOR r = 212; g = 208; b = 200  
PaintPrimitives.FILL\_RECT x = 0; y = 0; width = 16; height = 14  
PaintPrimitives.SET\_COLOR r = 255; g = 255; b = 255  
PaintPrimitives.DRAW\_LINE x1 = 0; y1 = 0; x2 = 0; y2 = 12  
PaintPrimitives.DRAW\_LINE x1 = 1; y1 = 0; x2 = 14; y2 = 0  
PaintPrimitives.SET\_COLOR r = 64; g = 64; b = 64  
PaintPrimitives.DRAW\_LINE x1 = 15; y1 = 0; x2 = 15; y2 = 13  
PaintPrimitives.DRAW\_LINE x1 = 0; y1 = 13; x2 = 15; y2 = 13  
PaintPrimitives.SET\_COLOR r = 128; g = 128; b = 128  
PaintPrimitives.DRAW\_LINE x1 = 14; y1 = 1; x2 = 14; y2 = 12  
PaintPrimitives.DRAW\_LINE x1 = 1; y1 = 12; x2 = 13; y2 = 12  
PaintPrimitives.DRAW\_IMAGE x = 0; y = 0; width = 16; height = 14  
PaintPrimitives.NO\_CLIP 2  
PaintPrimitives.TRANSLATE\_POP 1; x = -115; y = -2  
PaintPrimitives.NO\_CLIP 1  
PaintPrimitives.TRANSLATE\_POP 0; x = -4; y = -4  
PaintPrimitives.TRANSLATE\_PUSH 1; x = 4; y = 23  
PaintPrimitives.CLIP 2; x = 0; y = 0; width = 133; height = 21  
PaintPrimitives.SET\_COLOR r = 212; g = 208; b = 200  
PaintPrimitives.FILL\_RECT x = 0; y = 0; width = 133; height = 21  
**PaintPrimitives.RESIZE\_WINDOW id = -44; width = 8; height = 27**  
**PaintPrimitives.RESIZE\_WINDOW id = -44; width = 141; height = 48**  
PaintPrimitives.NO\_CLIP 1  
PaintPrimitives.TRANSLATE\_POP 0; x = -4; y = -23  
PaintPrimitives.NO\_CLIP 0  
PaintPrimitives.SET\_COLOR r = 212; g = 208; b = 200  
PaintPrimitives.FILL\_RECT x = 137; y = 0; width = 4; height = 48  
PaintPrimitives.FILL\_RECT x = 0; y = 44; width = 141; height = 4  
PaintPrimitives.FILL\_RECT x = 0; y = 0; width = 4; height = 48  
PaintPrimitives.FILL\_RECT x = 0; y = 0; width = 141; height = 4  
PaintPrimitives.SET\_COLOR r = 255; g = 255; b = 255  
PaintPrimitives.DRAW\_LINE x1 = 1; y1 = 1; x2 = 1; y2 = 45  
PaintPrimitives.DRAW\_LINE x1 = 2; y1 = 1; x2 = 138; y2 = 1  
PaintPrimitives.SET\_COLOR r = 128; g = 128; b = 128  
PaintPrimitives.DRAW\_LINE x1 = 1; y1 = 46; x2 = 139; y2 = 46  
PaintPrimitives.DRAW\_LINE x1 = 139; y1 = 1; x2 = 139; y2 = 46  
PaintPrimitives.SET\_COLOR r = 64; g = 64; b = 64  
PaintPrimitives.DRAW\_LINE x1 = 0; y1 = 47; x2 = 140; y2 = 47  
PaintPrimitives.DRAW\_LINE x1 = 140; y1 = 0; x2 = 140; y2 = 46  
END DRAWING CYCLE FOR WINDOW -44

#### #9 - 10/22/2015 05:20 AM - Sergey Ivanovskiy

Committed revision 11012 only changes to accept negative window ids and to log drawing circles.

Please, help to explain in general words the implementation of draw function of FrameWindowGuiImpl? There are contentPane.doLayout(); and doLayout(); invocations

```
/**
 * Draw widget.
 */
@Override
public void draw()
{
    contentPane.doLayout ();
    doLayout ();

    super.draw ();
}
```

#### #10 - 10/22/2015 05:23 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Please, help to explain in general words the implementation of draw function of FrameWindowGuiImpl? There are contentPane.doLayout(); and doLayout(); invocations

doLayout() layouts the window itself - window title, etc. contentPane.doLayout() layouts the content.

#### #11 - 10/22/2015 06:55 AM - Sergey Ivanovskiy

- File dialog\_title.png added

I have some troubles, the JS client can't handle correctly resizes requests in the middle of the drawings. It leads to position previously drawings incorrectly. But all resizes requests are from WindowLayout in our case of ModalWindow (see usages of gd.resizeWindow) I blocked the JS resize for the particular unfitted values in order to produce this screen.

#### #12 - 10/22/2015 06:59 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

I have some troubles, the JS client can't handle correctly resizes requests in the middle of the drawings. It leads to position previously drawings incorrectly. But all resizes requests are from WindowLayout in our case of ModalWindow (see usages of gd.resizeWindow) I blocked the JS resize for the particular unfitted values in order to produce this screen.

How come the title is drawn? Do we have any logic that assumes window title is always visible?

**#13 - 10/22/2015 07:14 AM - Sergey Ivanovskiy**

How come the title is drawn? Do we have any logic that assumes window title is always visible?

WindowTitleBar is responsible for drawing WindowTitle and it seems that there are no implementations of visibility logic by WindowTitle and WindowTitleBar.

**#14 - 10/22/2015 07:46 AM - Hynek Cihlar**

Please ignore note 12.

**#15 - 10/22/2015 09:36 AM - Sergey Ivanovskiy**

Please ignore note 12.

Debugging this method

```
public MessageReturnValue tinyInput(String text,
                                     int row,
                                     BaseDataType var,
                                     String format,
                                     boolean auto,
                                     boolean set)
```

of Window leads ModalWindow.doLayout and contentPane.doLayout(); are called several times from line 1569  
Frame frame = (Frame) screen.getRegistry().getComponent(id0);, from line 1598 of Window.java : frame.doLayout(); and third time from line 1621 of  
Window.java : tc.enable(id0, null, sb, true, wndld);. The last one invokes contentPane.doLayout(); and here it is set the old size for our modal window.  
It looks private void fillSpace(Container<O> container, Widget<O> innerPane) of BorderLayout calls WindowLayout but the caller class has no  
knowledge about the size is already calculated. Thus these invocations lead these resize events  
16:21:47.136 !!!! w = 8 h = 27 p2j.screen.js:591:10  
16:21:47.139 !!!! w = 141 h = 48 p2j.screen.js:591:10  
16:21:47.244 !!!! w = 8 h = 27 p2j.screen.js:591:10  
16:21:47.248 !!!! w = 141 h = 48  
on the JS side. It only explains an overview due to the complexity of the invoked code.

**#16 - 10/22/2015 09:47 AM - Sergey Ivanovskiy**

If we call doLayout inside draw it leads to resize events are inside drawing circles!

**#17 - 10/22/2015 09:51 AM - Constantin Asofiei**

Sergey Ivanovskiy wrote:

If we call doLayout inside draw it leads to resize events are inside drawing circles!

Are you referring to doLayout called from the BorderPanel.draw code?

**#18 - 10/22/2015 10:01 AM - Sergey Ivanovskiy**

If we call doLayout inside draw it leads to resize events are inside drawing circles!

Are you referring to doLayout called from the BorderPanel.draw code?

Yes, draw method of FrameWindowGuiImpl #note-9 invokes doLayout of BorderedPanel<GuiOutputManager> contentPane, and the resize events are inside the drawing circle due to BorderedPanel.doLayout returns different size for ModalWindow that contains it.

**#19 - 10/22/2015 10:10 AM - Constantin Asofiei**

Sergey Ivanovskiy wrote:

If we call doLayout inside draw it leads to resize events are inside drawing circles!

Are you referring to doLayout called from the BorderPanel.draw code?

Yes, draw method of FrameWindowGuiImpl #note-9 invokes doLayout of BorderedPanel<GuiOutputManager> contentPane, and the resize events are inside the drawing circle due to BorderedPanel.doLayout returns different size for ModalWindow that contains it.

OK, I don't think this is correct. At the time draw is called, all PaintEvent's are posted with the widget's coordinates/dimensions as at the time of the repaint. So, all clipping rectangles will be forced to draw using that dimensions... and calling doLayout() again in draw() might modify the widget's

dimension/location, which in turn will no longer match its associated PaintEvent (and bitmap rectangles added to the ScreenBitmap).

Hynek, do you think is OK to move the doLayout() calls from FrameWindowGuiImpl.draw() to its show() method? Also, shouldn't the layout code from BorderedPanel.draw() be in its own doLayout() method?

**#20 - 10/22/2015 10:26 AM - Hynek Cihlar**

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

If we call doLayout inside draw it leads to resize events are inside drawing circles!

Are you referring to doLayout called from the BorderPanel.draw code?

Yes, draw method of FrameWindowGuiImpl #note-9 invokes doLayout of BorderedPanel<GuiOutputManager> contentPane, and the resize events are inside the drawing circle due to BorderedPanel.doLayout returns different size for ModalWindow that contains it.

OK, I don't think this is correct. At the time draw is called, all PaintEvent's are posted with the widget's coordinates/dimensions as at the time of the repaint. So, all clipping rectangles will be forced to draw using that dimensions... and calling doLayout() again in draw() might modify the widget's dimension/location, which in turn will no longer match its associated PaintEvent (and bitmap rectangles added to the ScreenBitmap).

Hynek, do you think is OK to move the doLayout() calls from FrameWindowGuiImpl.draw() to its show() method?

I think I put doLayout() into draw() to cover the window resize case. But since, window resize events moved from Window to TopLevelWindow (and cover FrameWindowGuiImpl), so it should be ok to move to show().

Also, shouldn't the layout code from BorderedPanel.draw() be in its own doLayout() method?

Probably should. I never investigated this code in more detail. Also some caution should be taken, this is common to both ChUI and GUI



**#21 - 10/22/2015 10:29 AM - Constantin Asofiei**

Hynek Cihlar wrote:

Hynek, do you think is OK to move the doLayout() calls from FrameWindowGuiImpl.draw() to its show() method?

I think I put doLayout() into draw() to cover the window resize case. But since window resize events moved from Window to TopLevelWindow (and cover FrameWindowGuiImpl), so it should be ok to move to show().

Also, shouldn't the layout code from BorderedPanel.draw() be in its own doLayout() method?

Probably should. I never investigated this code in more detail. Also some caution should be taken, this is common to both ChUI and GUI

Sergey, please test the FrameWindowGuiImpl change as described above and leave the BorderedPanel.draw() as is for now (BorderedPanel change has a more chance at regressions, so if it works with changing only FrameWindowGuiImpl, we will check BorderedPanel after 1811s).

**#22 - 10/22/2015 10:31 AM - Hynek Cihlar**

Constantin Asofiei wrote:

Hynek Cihlar wrote:

Hynek, do you think is OK to move the doLayout() calls from FrameWindowGuiImpl.draw() to its show() method?

I think I put doLayout() into draw() to cover the window resize case. But since window resize events moved from Window to TopLevelWindow (and cover FrameWindowGuiImpl), so it should be ok to move to show().

Also, shouldn't the layout code from BorderedPanel.draw() be in its own doLayout() method?

Probably should. I never investigated this code in more detail. Also some caution should be taken, this is common to both ChUI and GUI

Sergey, please test the FrameWindowGuiImpl change as described above

Create a simple frame in DIALOG-BOX and then change its size through WIDTH/HEIGHT attributes.

### #23 - 10/22/2015 10:59 AM - Sergey Ivanovskiy

It doesn't help because of this stack trace

```
BorderLayout<O>.fillSpace(Container<O>, Widget<O>) line: 135
BorderLayout<O>.doLayout(Container<O>) line: 93
BorderedPanelGuiImpl(BorderedPanel<O>).draw() line: 136
BorderedPanelGuiImpl.access$501(BorderedPanelGuiImpl) line: 37
BorderedPanelGuiImpl$1$1.run() line: 200
GuiWebDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1923
BorderedPanelGuiImpl$1.run() line: 189
GuiWebDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1923
BorderedPanelGuiImpl.draw() line: 136
BorderedPanelGuiImpl(AbstractContainer<O>).draw() line: 326
BorderedPanelGuiImpl(BorderedPanel<O>).draw() line: 139
BorderedPanelGuiImpl.access$501(BorderedPanelGuiImpl) line: 37
BorderedPanelGuiImpl$1$1.run() line: 200
GuiWebDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1923
BorderedPanelGuiImpl$1.run() line: 189
GuiWebDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1923
BorderedPanelGuiImpl.draw() line: 136
FrameWindowGuiImpl(AbstractContainer<O>).draw() line: 326
FrameWindowGuiImpl(TitledWindow<O>).draw() line: 100
FrameWindowGuiImpl(OuterFrame<O>).drawInt() line: 98
FrameWindowGuiImpl(OuterFrame<O>).draw() line: 90
FrameWindowGuiImpl(ModalWindow).draw() line: 159
FrameWindowGuiImpl.draw() line: 159
GuiOutputManager(OutputManager<P>).setInvalidate(Widget<?>, boolean, boolean) line: 1275
ThinClient.eventDrawingBracket(Widget<?>, boolean, boolean, Runnable) line: 13653
ThinClient.independentEventDrawingBracket(Widget, Runnable) line: 13519
ThinClient.viewWorker(Widget, ScreenBuffer, int[], boolean, int) line: 9285
ThinClient.enableWorker(int[], ScreenBuffer, boolean, Widget, Frame, boolean, int) line: 4346
ThinClient.enable(int, int[], ScreenBuffer, boolean, int, boolean) line: 4213
ThinClient.enable(int, int[], ScreenBuffer, boolean, int) line: 4119
WindowGuiImpl(Window<O>).tinyInput(String, int, BaseType, String, boolean, boolean) line: 1621
```

It needs somehow to work around BorderedPanelGuiImpl(BorderedPanel<O>).draw()

#### #24 - 10/22/2015 11:04 AM - Sergey Ivanovskiy

But if we comment this code

```
/**
 * Draw widget.
 */
@Override
public void draw()
{
//     layoutManager<O> layout = getLayout();
//     if (layout != null)
//     {
//         layout.doLayout(this);
//     }

    super.draw();
}
```

then we fix the issue.

#### #25 - 10/22/2015 11:06 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

```
But if we comment this code
[...]
then we fix the issue.
```

Try creating a BorderedPanel.doLayout() method and move the code there.

#### #26 - 10/22/2015 11:14 AM - Sergey Ivanovskiy

Constantin, look at this, BorderLayout extends AbstractContainer and it already has doLayout.

```
/**
 * Lay out container.
 */
@Override
public void doLayout()
```

```
{
    if (layoutManager != null)
        layoutManager.doLayout(this);
}
```

How to check inside BorderLayout that chui mode is on?

**#27 - 10/22/2015 11:38 AM - Constantin Asofiei**

Sergey Ivanovskiy wrote:

Constantin, look at this, BorderLayout extends AbstractContainer and it already has doLayout.

[...]

How to check inside BorderLayout that chui mode is on?

There are ChUI and GUI specific sub-classes - BorderPanelImpl and BorderedPanelGuiImpl, so you don't need to explicitly check.

**#28 - 10/22/2015 11:40 AM - Sergey Ivanovskiy**

- File *resize\_issue\_2.txt* added

Thanks Constantin and Hynek, committed revision 11016 (branch 1811) should fix the issue. Committed revision 11017 adds comments and removes draw method of BorderLayout.

**#29 - 10/22/2015 12:04 PM - Hynek Cihlar**

Sergey Ivanovskiy wrote:

Thanks Constantin and Hynek, committed revision 11016 (branch 1811) should fix the issue. Committed revision 11017 adds comments and removes draw method of BorderLayout.

Sergey, just a minor thing. In `FrameWindowGuiImpl.show()`, `super.doLayout()` should be `doLayout()`.

**#30 - 10/22/2015 12:18 PM - Sergey Ivanovskiy**

Sergey, just a minor thing. In `FrameWindowGuiImpl.show()`, `super.doLayout()` should be `doLayout()`.

Committed revision 11019 fixes this issue.

**#31 - 10/22/2015 02:17 PM - Greg Shah**

- Status changed from New to Closed

- Target version set to Milestone 12

**#32 - 11/10/2015 05:09 AM - Greg Shah**

- % Done changed from 0 to 100

**#33 - 11/16/2016 12:12 PM - Greg Shah**

- Target version changed from Milestone 12 to GUI Support for a Complex ADM2 App

**Files**

---

|                    |         |            |                   |
|--------------------|---------|------------|-------------------|
| MessageUpdate.png  | 25.4 KB | 10/21/2015 | Sergey Ivanovskiy |
| dialog_title.png   | 55.6 KB | 10/22/2015 | Sergey Ivanovskiy |
| resize_issue_2.txt | 1.58 KB | 10/22/2015 | Sergey Ivanovskiy |