

User Interface - Feature #3394

generic drawing improvements: repaint and other misc code

11/27/2017 03:18 PM - Constantin Asofiei

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to User Interface - Feature #3246: reduce the amount of data being se...			Closed

History

#1 - 11/27/2017 03:19 PM - Constantin Asofiei

In branch 3369a (trunk rev 11206), there are some fixes which remove some aggressive repaint() calls when i.e. frame's size, location is set and other cases. These repaints were removed or performed only if some actual frame state was changed, which required the UI to redraw.

#2 - 11/27/2017 03:20 PM - Constantin Asofiei

An issue to try to improve is described by this email chain:

I agree that Hynek's idea of two queues (or pools) is probably necessary to implement this idea. Today, the EWS.offer() is protected with the same lock(), so Hynek's idea is the natural way to break the dependency.

Greg

On 11/27/2017 09:28 AM, Hynek Cihlar wrote:

Constantin,

I think you can take the similar locking approach in web as in current swing client. That is, have two pools of paint commands - a "working" and a "ready" pool. Where the working pool is used for PaintStructures coming from widgets and the ready pool used for "drawing" into web socket. The PSs are moved from working to ready pool on batch == 0.

Thanks,
Hynek

On 27.11.2017 15:01, Constantin Asofiei wrote:

Guys,

At some point, if I recall right, the Swing client was using the drawOnScreenBitmap approach, too, right? I.e. push the primitives to a queue and use a different thread for drawing... Now, in Swing we are drawing directly.

I've tried switching back the Swing mode to use this approach, but it's a lot slower than the direct approach. I think a reason might be because we drawOnScreenBitmap locks the entire loop code, not just the lock waiting code. Do you see any reason why the lock couldn't be moved inside the loop for web clients?

Thanks,

Constantin

#3 - 11/27/2017 03:22 PM - Constantin Asofiei

Another issue to improve is described by this email chain:

Constantin,

according to win32 api docs, the win32 calls that we replaced with DISABLE-REDRAW may do more stuff than just disabling paint updates. For example the win32 native list control may postpone adding new list items. So for now please leave the DISABLE-REDRAW attribute as we may find use cases where this could be helpful. But I still hope your optimizations will mitigate the need for DISABLE-REDRAW.

Thanks,
Hynek

On 27.11.2017 18:33, Greg Shah wrote:

Constantin,

I think there will be a limitation for Window movement/resize generated by the user

It seems like Window move can be handled with a cached image of the window. The move doesn't change the contents, it just draws the same contents in a different place.

The resize is different, but it also happens rarely.

Neither of these are relevant in embedded mode either, which is what production users will be using. We can probably set a "drawing refresh compatibility" flag in the directory that can be turned off to enable this optimization. Then customers could explicitly choose the slight compatibility differences if they didn't cause any real issue for a user.

I don't recall if I read in our redmine or somewhere else, but is it possible to change the UI draw mode in 4GL to behave like our DISABLE-REDRAW attribute?

This is only possible by making WIN32 API calls. Our enhancement provided "4GL" syntax to do this same this without the WIN32 API calls.

Beside this, the only difference I see at this time between FWD and normal 4GL UI drawing is that in FWD there will be no more 'flickering' (and flickering can be seen in some AW screens in 4GL, too, when widgets are drawn 'from left to right, top-down' as Eric mentioned at some point).

As much as I love flickering UIs, I guess I can live without it.

Go ahead with this idea.

Greg

On 11/27/2017 12:25 PM, Constantin Asofiei wrote:

Greg/Hynek,

I think there will be a limitation for Window movement/resize generated by the user; but if there is a event processing loop active, then the repaints will be picked up and UI drawn. Only case I can think of resize if there is no event processing loop active.

I don't recall if I read in our redmine or somewhere else, but is it possible to change the UI draw mode in 4GL to behave like our DISABLE-REDRAW attribute?

Beside this, the only difference I see at this time between FWD and normal 4GL UI drawing is that in FWD there will be no more 'flickering' (and flickering can be seen in some AW screens in 4GL, too, when widgets are drawn 'from left to right, top-down' as Eric mentioned at some point).

Thanks,

Constantin

On 11/27/2017 07:08 PM, Hynek Cihlar wrote:

I believe top-level windows are moved and resized directly and not postponed until the event processing resumes.

Thanks,
Hynek

On 27.11.2017 18:05, Greg Shah wrote:

Constantin,

I am in favor of this. My only concern: what are the potential negative consequences from a 4GL compatibility standpoint?

Are there scenarios where the 4GL would draw some useful content on the screen and then there is a delay before a blocking statement would make it visible in FWD?

Greg

On 11/27/2017 11:41 AM, Constantin Asofiei wrote:

Guys,

Cutting down on the trips from server to client side is more problematic than I thought; the SENSITIVE and VISIBLE attributes are used a lot by the business logic in the same code where other attributes (like location, size, are being set), and I don't know yet how I can 'ignore' this trip to the client-side safely, as the logic is more complex. And when this is done in a loop, the performance impact is pretty obvious just from SENSITIVE/VISIBLE.

But I have an alternative: Hynek added a DISABLE-REDRAW FWD extension... thinking on this, why not just ignore all PaintEvent's, accumulate them in a map (per each window), and before the client blocks/checks for an event, process these PaintEvent's and allow the UI to draw (just before typeAhead.getKeystroke) ?

The changes are minimal (in code), but the impact in both Swing and Web is pretty obvious.

OTOH, these changes would make the DISABLE-REDRAW attribute obsolete...

Thanks,

Constantin

Another reply from Hynek:

On 27.11.2017 17:41, Constantin Asofiei wrote:

Guys,

Cutting down on the trips from server to client side is more problematic than I thought; the SENSITIVE and VISIBLE attributes are used a lot by the business logic in the same code where other attributes (like location, size, are being set), and I don't know yet how I can 'ignore' this trip to the client-side safely, as the logic is more complex. And when this is done in a loop, the performance impact is pretty obvious just from SENSITIVE/VISIBLE.

But I have an alternative: Hynek added a DISABLE-REDRAW FWD extension... thinking on this, why not just ignore all PaintEvent's, accumulate them in a map (per each window), and before the client blocks/checks for an event, process these PaintEvent's and allow the UI to draw (just before typeAhead.getKeystroke) ?

Sounds good to me. In similar way it would also help to postpone (accumulate) widget layout requests.

Thanks,
Hynek

#4 - 11/27/2017 04:40 PM - Constantin Asofiei

For [#3394-3](#), the changes are in 3394a rev 11207.

#5 - 11/27/2017 05:38 PM - Constantin Asofiei

I've been looking into the JS code and there are some findings:

1. we can reduce the MSG_REGISTER_HOVERABLE_WIDGET calls by batching them together... now these are sent immediately via the websocket, I don't see a reason a distinct call is needed for these, as the user can't do anything with the widget until it is being drawn and the app is in user-interactive state. The use-case is when a screen has tens of editable widgets/buttons which need to react on mouse hover - there will be a websocket call for each of them.
2. the MSG_DRAW message - for complex windows, I've encountered even ~500KB for a single window page, with an average of ~300KB a message. Currently, I'm interested in checking the 3394a branch with the remote VM, maybe the response has improved

#6 - 11/27/2017 06:49 PM - Constantin Asofiei

3394a rev 11208 adds this:

- Batch the MSG_REGISTER_HOVERABLE_WIDGET messages.
- Avoid gd.setCurrentSelection(null) if the selection was invalid already. This call is expensive on the web driver.
- Moved code used only by the Web driver to the Web specific classes.

I think the widget register/deregister calls can be batched, too.

#7 - 11/28/2017 10:57 AM - Greg Shah

Do you want to make 3394a the next "accumulate the fixes" branch? Or would you prefer to keep it separate?

#8 - 11/28/2017 11:43 AM - Constantin Asofiei

Greg Shah wrote:

Do you want to make 3394a the next "accumulate the fixes" branch? Or would you prefer to keep it separate?

Yes, we can use 3394a.

And on topic: I think the improvements I made to postpone the drawing should be done only in GUI; ChUI has the PUT_SCREEN stuff which requires specific order of the UI drawing operations.

#9 - 11/28/2017 01:15 PM - Greg Shah

I think the improvements I made to postpone the drawing should be done only in GUI; ChUI has the PUT_SCREEN stuff which requires specific order of the UI drawing operations.

Agreed.

#10 - 11/28/2017 04:08 PM - Greg Shah

Code Review Task Branch 3394a Revision 11211

1. In `Button.setText()` and `ToggleBox.setText()` why is it safe to only repaint in `autoResize` mode? Doesn't any change of text cause a repaint?
2. In `ToggleBoxGuiImpl.setText()` , it overrides and doesn't call the `ToggleBox` version. Thus the `ToggleBox` changes are only for `ChUI`?
3. In `ToggleBoxGuiImpl.setText()`, changing the text is not immediately visible?

#11 - 11/28/2017 04:11 PM - Constantin Asofiei

Greg Shah wrote:

1. In `Button.setText()` and `ToggleBox.setText()` why is it safe to only repaint in `autoResize` mode? Doesn't any change of text cause a repaint?

The `AbstractButton.setTextLabel` takes care of the repaint, if needed.

2. In `ToggleBoxGuiImpl.setText()` , it overrides and doesn't call the `ToggleBox` version. Thus the `ToggleBox` changes are only for `ChUI`?

Same as issue 1. Also, `autoResize` in `GUI` is different as I recall, so nothing to do here explicitly.

3. In `ToggleBoxGuiImpl.setText()`, changing the text is not immediately visible?

Same as issue 1.

#12 - 11/29/2017 05:58 AM - Constantin Asofiei

3394a 11213 passed main runtime testing.

#13 - 12/01/2017 09:40 AM - Constantin Asofiei

I'm rebasing 3394a rev 11217 in 15 minutes if nobody objects.

#14 - 12/01/2017 09:55 AM - Constantin Asofiei

Constantin Asofiei wrote:

I'm rebasing 3394a rev 11217 in 15 minutes if nobody objects.

#15 - 12/01/2017 10:27 AM - Constantin Asofiei

3394a was rebased from trunk rev 11208 - new rev 11219.

#16 - 12/04/2017 01:35 PM - Sergey Ivanovskiy

Interesting note if setup conditional breakpoints for the tree control widgets in `AbstractWidget.afterConfigUpdateBase`, then the tree control is repainted correctly. Thus it can be that repaint events have been processed at incorrect moments.

#17 - 12/04/2017 01:38 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Interesting note if setup conditional breakpoints for the tree control widgets in `AbstractWidget.afterConfigUpdateBase`, then the tree control is repainted correctly. Thus it can be that repaint events have been processed at incorrect moments.

You might encounter window-related paint events (which force the entire window to redraw) when switching windows; so I would not rely on what you found to be true. Instead, can you log the repaint events in `PaintEvent` - source widget, location, size?

#18 - 12/04/2017 03:08 PM - Sergey Ivanovskiy

Yes, I tested Hotel GUI Add Room Dialog and opened the first node and then opened the second node. Finally, the last two nodes were not repainted.

1) opened this dialog

```
widget= ButtonGuiImpl id=113 bounds=Rectangle[top=0.48, left=44.2, bottom=0.48, right=44.2] updateRect=Rectangle[top=2.01, left=44.6, bottom=2.01, right=44.6]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=3.38, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=5.39, right=65.4]
widget= ImageGuiImpl id=114 graphics/treeview/dottedVerticalLine.png bounds=Rectangle[top=0.29, left=0.8, bottom=3.0, right=0.8] updateRect=Rectangle[top=2.3, left=46.2, bottom=5.01, right=46.2]
widget= TextGuiImpl id=115 Single bounds=Rectangle[top=0.05, left=6.8, bottom=0.62, right=13.6] updateRect=Rectangle[top=2.06, left=52.2, bottom=2.63, right=59.0]
widget= ImageGuiImpl id=116 graphics/treeview/level.png bounds=Rectangle[top=0.1, left=3.4, bottom=0.81, right=6.4] updateRect=Rectangle[top=2.11, left=48.8, bottom=2.82, right=51.8]
widget= ImageGuiImpl id=117 graphics/treeview/plus.png bounds=Rectangle[top=0.24, left=0.0, bottom=0.62, right=2.8] updateRect=Rectangle[top=2.25, left=45.4, bottom=2.63, right=48.2]
widget= TextGuiImpl id=118 Double bounds=Rectangle[top=0.91, left=6.8, bottom=1.48, right=14.4] updateRect=Rectangle[top=2.92, left=52.2, bottom=3.49, right=59.8]
widget= ImageGuiImpl id=119 graphics/treeview/level.png bounds=Rectangle[top=0.95, left=3.4, bottom=1.66, right=6.4] updateRect=Rectangle[top=2.96, left=48.8, bottom=3.67, right=51.8]
widget= ImageGuiImpl id=120 graphics/treeview/plus.png bounds=Rectangle[top=1.1, left=0.0, bottom=1.48, right=2.8] updateRect=Rectangle[top=3.11, left=45.4, bottom=3.49, right=48.2]
```

```
widget= TextGuiImpl id=121 Twin bounds=Rectangle[top=1.76, left=6.8, bottom=2.33, right=12.6] updateRect=Rectangle[top=3.77, left=52.2, bottom=4.34, right=58.0]
widget= ImageGuiImpl id=122 graphics/treeview/level.png bounds=Rectangle[top=1.81, left=3.4, bottom=2.52, right=6.4] updateRect=Rectangle[top=3.82, left=48.8, bottom=4.53, right=51.8]
widget= ImageGuiImpl id=123 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=2.19, left=0.6, bottom=2.19, right=2.8] updateRect=Rectangle[top=4.2, left=46.0, bottom=4.2, right=48.2]
widget= TextGuiImpl id=124 Luxury Suite bounds=Rectangle[top=2.62, left=6.8, bottom=3.19, right=20.0] updateRect=Rectangle[top=4.63, left=52.2, bottom=5.2, right=65.4]
widget= ImageGuiImpl id=125 graphics/treeview/level.png bounds=Rectangle[top=2.67, left=3.4, bottom=3.38, right=6.4] updateRect=Rectangle[top=4.68, left=48.8, bottom=5.39, right=51.8]
widget= ImageGuiImpl id=126 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=3.05, left=0.6, bottom=3.05, right=2.8] updateRect=Rectangle[top=5.06, left=46.0, bottom=5.06, right=48.2]
```

1) opened the first node "Single"

```
widget= ButtonGuiImpl id=112 bounds=Rectangle[top=0.0, left=0.0, bottom=0.0, right=0.0] updateRect=Rectangle[top=1.96, left=44.6, bottom=2.21, right=44.65]
widget= ButtonGuiImpl id=112 bounds=Rectangle[top=0.0, left=0.0, bottom=0.0, right=0.0] updateRect=Rectangle[top=2.01, left=44.6, bottom=2.01, right=44.6]
widget= FrameGuiImpl id=127 bounds=Rectangle[top=0.0, left=0.0, bottom=0.81, right=0.2] updateRect=Rectangle[top=2.01, left=45.4, bottom=2.41, right=46.26]
widget= FrameGuiImpl id=127 bounds=Rectangle[top=0.0, left=0.0, bottom=0.81, right=23.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=2.82, right=45.6]
widget= FrameGuiImpl id=127 bounds=Rectangle[top=0.0, left=0.0, bottom=0.81, right=23.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=2.41, right=46.26]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=3.38, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=5.39, right=65.4]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=3.38, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=5.39, right=65.4]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=3.38, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=5.39, right=65.4]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=5.39, right=65.4]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=8.58, right=48.03]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=65.4]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=65.4]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=20.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=65.4]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=65.4]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=8.58, right=48.89]
widget= TextGuiImpl id=118 Double bounds=Rectangle[top=0.91, left=6.8, bottom=1.48, right=14.4] updateRect=Rectangle[top=2.92, left=52.2, bottom=3.49, right=59.8]
widget= ImageGuiImpl id=114 graphics/treeview/dottedVerticalLine.png bounds=Rectangle[top=0.29, left=0.8, bottom=3.86, right=0.8] updateRect=Rectangle[top=2.3, left=46.2, bottom=5.87, right=46.2]
widget= ImageGuiImpl id=114 graphics/treeview/dottedVerticalLine.png bounds=Rectangle[top=0.29, left=0.8, bottom=3.86, right=0.8] updateRect=Rectangle[top=2.3, left=46.2, bottom=5.87, right=46.2]
widget= ImageGuiImpl id=119 graphics/treeview/level.png bounds=Rectangle[top=0.95, left=3.4, bottom=1.66, right=6.4] updateRect=Rectangle[top=2.96, left=48.8, bottom=3.67, right=51.8]
widget= ImageGuiImpl id=120 graphics/treeview/plus.png bounds=Rectangle[top=1.1, left=0.0, bottom=1.48, right=2.8] updateRect=Rectangle[top=2.26, left=45.4, bottom=5.01, right=45.83]
widget= ImageGuiImpl id=120 graphics/treeview/plus.png bounds=Rectangle[top=1.1, left=0.0, bottom=1.48, right=2.8] updateRect=Rectangle[top=3.11, left=45.4, bottom=3.49, right=48.2]
widget= TextGuiImpl id=121 Twin bounds=Rectangle[top=1.76, left=6.8, bottom=2.33, right=12.6] updateRect=Rectangle[top=3.77, left=52.2, bottom=4.34, right=58.0]
widget= ImageGuiImpl id=122 graphics/treeview/level.png bounds=Rectangle[top=1.81, left=3.4, bottom=2.52, right=6.4] updateRect=Rectangle[top=3.82, left=48.8, bottom=4.53, right=51.8]
widget= ImageGuiImpl id=123 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=2.19, left=0.6, bottom=2.19, right=2.8] updateRect=Rectangle[top=4.2, left=46.0, bottom=4.2, right=48.2]
widget= TextGuiImpl id=124 Luxury Suite bounds=Rectangle[top=2.62, left=6.8, bottom=3.19, right=20.0] updateRect=Rectangle[top=4.63, left=52.2, bottom=5.2, right=65.4]
widget= ImageGuiImpl id=125 graphics/treeview/level.png bounds=Rectangle[top=2.67, left=3.4, bottom=3.38, right=6.4] updateRect=Rectangle[top=4.68, left=48.8, bottom=5.39, right=51.8]
widget= ImageGuiImpl id=126 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=3.05, left=0.6, bottom=3.05, right=2.8] updateRect=Rectangle[top=5.06, left=46.0, bottom=5.06, right=48.2]
widget= FrameGuiImpl id=127 bounds=Rectangle[top=0.0, left=4.0, bottom=0.81, right=27.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=46.26]
widget= FrameGuiImpl id=127 bounds=Rectangle[top=0.86, left=4.0, bottom=1.67, right=27.8] updateRect=Rectangle[top=2.87, left=49.4, bottom=3.68, right=73.2]
```

```
widget= ImageGuiImpl id=130 graphics/treeview/dottedVerticalLine.png bounds=Rectangle[top=0.0, left=0.8, bottom=0.43, right=0.8] updateRect=Rectangle[top=2.87, left=50.2, bottom=3.3, right=50.2]
widget= TextGuiImpl id=131 Single - Superior bounds=Rectangle[top=0.05, left=6.8, bottom=0.62, right=23.8] updateRect=Rectangle[top=2.92, left=56.2, bottom=3.49, right=73.2]
widget= ImageGuiImpl id=132 graphics/treeview/level.png bounds=Rectangle[top=0.1, left=3.4, bottom=0.81, right=6.4] updateRect=Rectangle[top=2.97, left=52.8, bottom=3.68, right=55.8]
widget= ImageGuiImpl id=133 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=0.48, left=0.6, bottom=0.48, right=2.8] updateRect=Rectangle[top=3.35, left=50.0, bottom=3.35, right=52.2]
widget= ImageGuiImpl id=117 graphics/treeview/minus.png bounds=Rectangle[top=0.24, left=0.0, bottom=0.62, right=2.8] updateRect=Rectangle[top=2.25, left=45.4, bottom=2.63, right=48.2]
```

2) opened the second node "Double"

```
widget= ButtonGuiImpl id=112 bounds=Rectangle[top=0.71, left=0.0, bottom=0.71, right=0.0] updateRect=Rectangle[top=1.96, left=44.6, bottom=2.21, right=44.65]
widget= ButtonGuiImpl id=112 bounds=Rectangle[top=0.71, left=0.0, bottom=0.71, right=0.0] updateRect=Rectangle[top=2.72, left=44.6, bottom=2.72, right=44.6]
widget= FrameGuiImpl id=134 bounds=Rectangle[top=0.0, left=0.0, bottom=0.81, right=0.2] updateRect=Rectangle[top=2.01, left=45.4, bottom=2.41, right=46.26]
widget= FrameGuiImpl id=134 bounds=Rectangle[top=0.0, left=0.0, bottom=0.81, right=26.0] updateRect=Rectangle[top=2.01, left=45.4, bottom=2.82, right=45.6]
widget= FrameGuiImpl id=134 bounds=Rectangle[top=0.0, left=0.0, bottom=0.81, right=26.0] updateRect=Rectangle[top=2.01, left=45.4, bottom=2.41, right=46.26]
widget= FrameGuiImpl id=134 bounds=Rectangle[top=0.0, left=0.0, bottom=1.66, right=26.0] updateRect=Rectangle[top=2.01, left=45.4, bottom=2.82, right=71.4]
widget= FrameGuiImpl id=134 bounds=Rectangle[top=0.0, left=0.0, bottom=1.66, right=26.0] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=46.26]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=73.2]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=73.2]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=4.24, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=73.2]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=5.95, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=6.25, right=73.2]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=5.95, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=8.58, right=48.89]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=5.95, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=7.96, right=73.2]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=5.95, right=28.6] updateRect=Rectangle[top=2.01, left=45.4, bottom=7.96, right=73.2]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=5.95, right=30.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=7.96, right=73.2]
widget= FrameGuiImpl id=109 bounds=Rectangle[top=0.0, left=0.8, bottom=5.95, right=30.8] updateRect=Rectangle[top=2.01, left=45.4, bottom=8.58, right=50.6]
widget= TextGuiImpl id=121 Twin bounds=Rectangle[top=2.62, left=6.8, bottom=3.19, right=12.6] updateRect=Rectangle[top=4.63, left=52.2, bottom=5.2, right=58.0]
widget= ImageGuiImpl id=114 graphics/treeview/dottedVerticalLine.png bounds=Rectangle[top=0.29, left=0.8, bottom=5.57, right=0.8] updateRect=Rectangle[top=2.3, left=46.2, bottom=7.58, right=46.2]
widget= ImageGuiImpl id=114 graphics/treeview/dottedVerticalLine.png bounds=Rectangle[top=0.29, left=0.8, bottom=5.57, right=0.8] updateRect=Rectangle[top=2.3, left=46.2, bottom=7.58, right=46.2]
widget= ImageGuiImpl id=122 graphics/treeview/level.png bounds=Rectangle[top=2.67, left=3.4, bottom=3.38, right=6.4] updateRect=Rectangle[top=4.68, left=48.8, bottom=5.39, right=51.8]
widget= ImageGuiImpl id=123 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=3.05, left=0.6, bottom=3.05, right=2.8] updateRect=Rectangle[top=5.06, left=46.0, bottom=5.06, right=48.2]
widget= TextGuiImpl id=124 Luxury Suite bounds=Rectangle[top=3.48, left=6.8, bottom=4.05, right=20.0] updateRect=Rectangle[top=5.49, left=52.2, bottom=6.06, right=65.4]
widget= ImageGuiImpl id=125 graphics/treeview/level.png bounds=Rectangle[top=3.52, left=3.4, bottom=4.23, right=6.4] updateRect=Rectangle[top=5.53, left=48.8, bottom=6.24, right=51.8]
widget= ImageGuiImpl id=126 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=3.91, left=0.6, bottom=3.91, right=2.8] updateRect=Rectangle[top=5.92, left=46.0, bottom=5.92, right=48.2]
widget= FrameGuiImpl id=134 bounds=Rectangle[top=0.0, left=4.0, bottom=1.66, right=30.0] updateRect=Rectangle[top=2.01, left=45.4, bottom=7.96, right=47.11]
widget= FrameGuiImpl id=134 bounds=Rectangle[top=2.57, left=4.0, bottom=4.23, right=30.0] updateRect=Rectangle[top=4.58, left=49.4, bottom=6.24, right=75.4]
widget= ImageGuiImpl id=137 graphics/treeview/dottedVerticalLine.png bounds=Rectangle[top=0.0, left=0.8, bottom=1.28, right=0.8] updateRect=Rectangle[top=4.58, left=50.2, bottom=5.86, right=50.2]
widget= TextGuiImpl id=138 Double - Sea View bounds=Rectangle[top=0.05, left=6.8, bottom=0.62, right=26.0] updateRect=Rectangle[top=4.63, left=56.2, bottom=5.2, right=75.4]
widget= ImageGuiImpl id=139 graphics/treeview/level.png bounds=Rectangle[top=0.1, left=3.4, bottom=0.81, right=6.4] updateRect=Rectangle[top=4.68, left=52.8, bottom=5.39, right=55.8]
```



```
widget= ImageGuiImpl id=140 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=0.48, left=0.6, bottom=0.48, right=2.8] updateRect=Rectangle[top=5.06, left=50.0, bottom=5.06, right=52.2]
widget= TextGuiImpl id=141 Double - King Bed bounds=Rectangle[top=0.91, left=6.8, bottom=1.48, right=25.0] updateRect=Rectangle[top=5.49, left=56.2, bottom=6.06, right=74.4]
widget= ImageGuiImpl id=142 graphics/treeview/level.png bounds=Rectangle[top=0.95, left=3.4, bottom=1.66, right=6.4] updateRect=Rectangle[top=5.53, left=52.8, bottom=6.24, right=55.8]
widget= ImageGuiImpl id=143 graphics/treeview/dottedHorizontalLine.png bounds=Rectangle[top=1.33, left=0.6, bottom=1.33, right=2.8] updateRect=Rectangle[top=5.91, left=50.0, bottom=5.91, right=52.2]
widget= ImageGuiImpl id=120 graphics/treeview/minus.png bounds=Rectangle[top=1.95, left=0.0, bottom=2.33, right=2.8] updateRect=Rectangle[top=3.96, left=45.4, bottom=4.34, right=48.2]
```

Finally, the last two nodes: "Twin" and "Luxury Suite " were not repainted.

#19 - 12/04/2017 03:10 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Finally, the last two nodes: "Twin" and "Luxury Suite " were not repainted.

Do you now the IDs for these two widgets, to look into the logs you posted for repaints?

#20 - 12/04/2017 03:25 PM - Sergey Ivanovskiy

Updated [#3394-18](#) with correct logs. Yes, they are 121 and 124.

#21 - 12/04/2017 03:27 PM - Sergey Ivanovskiy

The order of repaints can be the root cause?

#22 - 12/04/2017 04:00 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

The order of repaints can be the root cause?

No, as they just determine which widgets will be drawn, once every repaint rectangle is accumulated.

Please check this: use `screenPhysicalLocation` to translate the widget's native coordinates to coordinates relative to the parent window. This will allow you to measure exactly on the UI (via a Screen Ruler on ubuntu or gimp or something else) the pixel position of the widget and the pixel coordinates of the repaint rectangle.

#23 - 12/06/2017 03:24 PM - Sergey Ivanovskiy

With the current 3394a the Hotel GUI Check-in date picker doesn't repaint items correctly if the current month (year) is changed by clicking on the '>'('>>') button.

#24 - 12/06/2017 03:25 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

With the current 3394a the Hotel GUI Check-in date picker doesn't repaint items correctly if the current month (year) is changed by clicking on the '>'('>>') button.

Thanks for finding this, please look into it.

#25 - 12/07/2017 09:44 AM - Sergey Ivanovskiy

Please help to answer why in this code adatesd0.w for the date picker the x and y coordinates for the selected date rectangle are taken from the frame.

```
IF dtLoop = TODAY
  THEN DO:
    CREATE RECTANGLE LhTodayRectangle
      ASSIGN FRAME = FRAME {&FRAME-NAME}:HANDLE
        X = LhButton[iCount + 7]:FRAME:X - 1
        Y = LhButton[iCount + 7]:FRAME:Y - 1
        WIDTH-PIXELS = {&BUTTON_WIDTH_PIXELS} + 2
        HEIGHT-PIXELS = {&BUTTON_HEIGHT_PIXELS} + 2
        EDGE-PIXELS = 1
        FGCOLOR = 12
        .
      LhTodayRectangle:MOVE-TO-TOP().
    END.
```

It seems that for this code it needs the button's x and y coordinates.

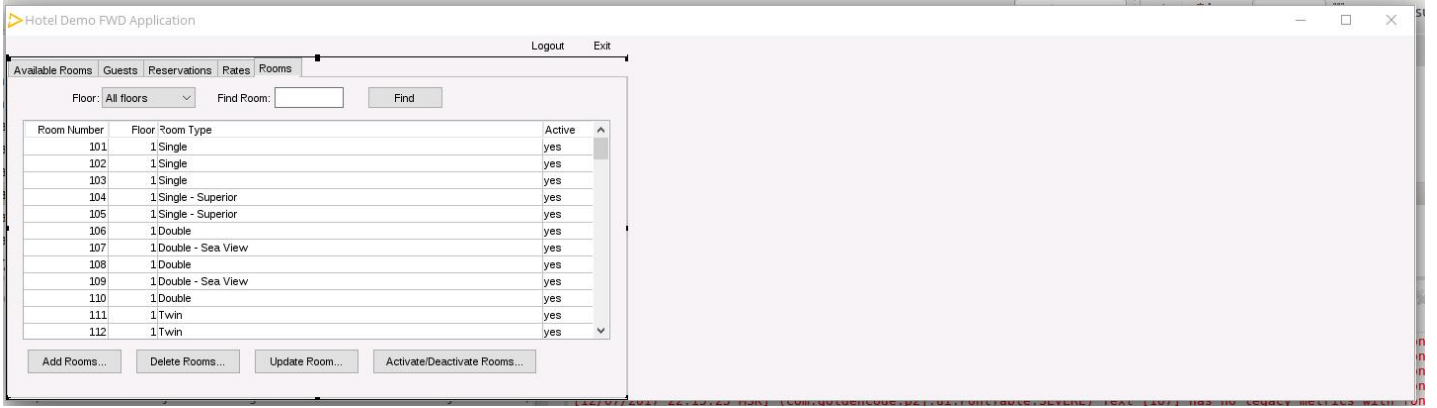
#26 - 12/07/2017 09:54 AM - Sergey Ivanovskiy

Don't mind, I answered by myself.

#27 - 12/07/2017 02:22 PM - Sergey Ivanovskiy

- File resizable.png added

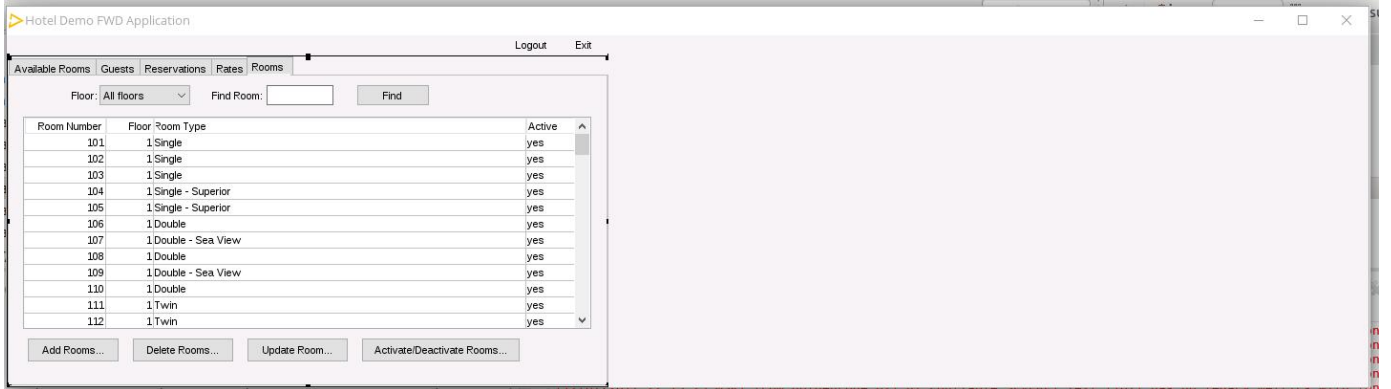
With the current version the main tab frame can be resizable and resize or editor grid is displayed. It wasn't observed with the previous versions.



#28 - 12/07/2017 02:33 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

With the current version the main tab frame can be resizable and resize or editor grid is displayed. It wasn't observed with the previous versions.



What's the frame state? Is it disabled?

Don't know it is an answer. The frame is folder-frm 27 Dimension[126.0,16.43] Point[0.0,1.0] visible true can false;

```
this    FrameGuiImpl  (id=2774)
  active      true
  allClear    false
  autoLayout  false
  autoPlacing false
  canHide     false
  canHideOthers false
  canWipeScreen false
  cc          CoordinatesConversion (id=2614)
  chooseHandler ChooseHandler<O> (id=2841)
  colorChanged false
  config      FrameConfig (id=2842)
  configUpdateActive false
  contentPane FrameGuiImpl$GuiScrollContainer (id=2708)
  current     ScrollPaneGuiImpl (id=2816)
  currentState null
  cursor      -1
  defaultMinimalHeight 0.86
  defaultMinimalWidth 3.6
  delayedRepaint false
  directManipulationActive true
  down        false
  downBody    Widget<O>[1][] (id=2847)
  dynamicCol  -2.147483648E9
  dynamicDown false
  dynamicRow  -2.147483648E9
  enabled     true
  explicitHide false
  explicitVirtualHeight false
  explicitVirtualWidth false
  fixedWidth  true
  flushed     false
  focusAttractor false
  focusId     -1
  focusListeners ArrayList<E> (id=2852)
  focusReset  false
  focusTransferManager FrameFocusTransferManager (id=2853)
  forcedRefresh false
  forcePause  false
  forceWipeScreen false
  frameScroll ScrollPaneGuiImpl (id=2816)
  frameTitle  BorderedPanelGuiImpl (id=2855)
  gc          GuiColorResolver (id=2856)
  gct         GuiColorResolver (id=2857)
  gd          SwingGuiDriver (id=2675)
  gf          GuiFontResolver (id=2858)
  headersHeight -2.147483648E9
  headersOutput null
  hidden      false
  highlighted  true
  honorNextConditionalDown false
  ignoreFocusSet null
  inScope     false
  insets      Insets (id=2682)
  insets      Insets (id=2860)
  isFixedRow  true
  keyListeners ArrayList<E> (id=2861)
  lastRow     -1
  layoutManager ZeroColumnLayout<O> (id=2762)
  leadingSkipSize -1.0
  lineBorder  LineBorderGuiImpl (id=2863)
  listeners   ArrayList<E> (id=2864)
  live        true
  location    Point (id=2865)
  mouseActions null
  mouseHoverAction null
  mousePtr    Optional<T> (id=2866)
  nativeInsets NativeInsets (id=2697)
  needClear   null
  needPause   false
```

```
newRow      0
nextTabItemList  ArrayList<E> (id=2773)
nonbody     Widget<O>[0] (id=2868)
origHeight  0.0
originalHeightChars  4.15
paintable   true
parent      WeakReference<T> (id=2870)
pendingCursorPos  0
pendingCursorWrap  false
pendingDown  -1
pendingFixedDown  false
pendingNeedClear  false
pendingUnderline  false
physicalLocation  NativePoint (id=2874)
placedForRedirected  false
popupKeyListenerAdded  false
postponedPlace  false
postponedVisible  false
protect      false
redirected   false
relocY       HashMap<K,V> (id=2983)
resetPending  false
resizH       HashMap<K,V> (id=2988)
savedDim     Dimension (id=2989)
savedDown    1
saveDown     -1
savedPaneDim  Dimension (id=2990)
savedState   null
savedTabOrder  Widget<O>[4] (id=2991)
scopeNesting  0
screen       GuiOutputManager (id=2768)
screen       GuiOutputManager (id=2768)
size         Dimension (id=2993)
spareIteration  null
startRow     null
streamed     false
streamId     -1
streams      HashSet<E> (id=2994)
tabItemList  null
touched      null
trailingSkips  false
trailingSkipSize  -1.0
underline    null
underlinedRow  null
verticalStep  -1.9073486327847444E-7
verticalStepBoxed  -1.9073486327847444E-7
viewed       true
visibilityChanged  false
visible      true
widgets      ArrayList<E> (id=3000)
withEditors  false
frameTitleHeight  0
```

```
FrameGuiImpl.getResizeRectangle() line: 1392
MouseDirectManipulation.getFrameResizeHandles(Widget<?>) line: 1249
MouseDirectManipulation.isInsideFrameResizeHandle(NativePoint) line: 1091
MouseDirectManipulation.getNewPointer(MouseEvent) line: 1396
MouseDirectManipulation.mouseMoved(MouseEvent) line: 484
MouseHandler.processWidgetActions(int, MouseEvent) line: 456
MouseHandler.handleMouseEvent(int, MouseEvent) line: 295
SwingGuiDriver(AbstractGuiDriver<F>).handleMouseEvent(int, MouseEvent) line: 2827
```

#30 - 12/07/2017 02:50 PM - Sergey Ivanovskiy

This resize rectangle is displayed forever after its appearance on double click at the frame space right to the last tab.

#31 - 12/07/2017 02:53 PM - Sergey Ivanovskiy

Please answer to this conceptual question. What is the meaning of this flag and when it can be used (BaseConfig)? Can I use it as a redraw flag?

```
/**
 * Flag indicating if the widget has been placed by the layout. For top-label frames,
 * this is required to distinguish between widgets which have their location set before or
 * after they were attached to a frame.
 */
public transient boolean widgetPlaced = false;
```

#32 - 12/07/2017 03:17 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Please answer to this conceptual question. What is the meaning of this flag and when it can be used (BaseConfig)? Can I use it as a redraw flag?
[...]

What are you trying to solve? This flag is used only during layout, is not related to drawing.

#33 - 12/07/2017 03:18 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

This resize rectangle is displayed forever after its appearance on double click at the frame space right to the last tab.

Eugenie, if you can, please provide some hints for Sergey to solve. Otherwise, please look at it.

#34 - 12/07/2017 03:25 PM - Sergey Ivanovskiy

OK, I lost the conception of redrawing if the business methods change some widget (its position, its size, its content) that has been already placed at the frame. Now it seems that the system works incorrectly because there are at least two examples: the date picker and the compound tree control on which we can observe this issue that widget's updates are not reflected by the drawing.

#35 - 12/07/2017 03:32 PM - Eugenie Lyzenko

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

This resize rectangle is displayed forever after its appearance on double click at the frame space right to the last tab.

Eugenie, if you can, please provide some hints for Sergey to solve. Otherwise, please look at it.

The option:

```
directManipulationActive true
```

For the given frame is incorrect. It must be false.

#36 - 12/07/2017 03:32 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

OK, I lost the conception of redrawing if the business methods change some widget (its position, its size, its content) that has been already placed at the frame. Now it seems that the system works incorrectly because there are at least two examples: the date picker and the compound tree control on which we can observe this issue that widget's updates are not reflected by the drawing.

Please post a screenshot of the draw corruption you see in the date picker. The root cause is most likely a case where we either (or all):

1. do not raise a PaintEvent when we should have
2. the coordinates for the PaintEvent is incorrect
3. the repaint is discarded

#37 - 12/08/2017 04:43 AM - Sergey Ivanovskiy

- File *date_picker.mkv* added

Please look at this picture date_picker.mkv.

#38 - 12/08/2017 04:56 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Please look at this picture date_picker.mkv.

You refer to the border artefacts around the day button plus the fact that the day button remains in a pressed state?

#39 - 12/08/2017 04:58 AM - Sergey Ivanovskiy

Yes, and the fact that the month combo box is not repainted with actual value.

It seems that the `NativeRectangle.this.intersection` and `NativeRectangle.this.intersects` don't consider cases with empty rectangles. I didn't fix this because don't know their impacts on the performance. Empty rectangles can be materialized as drawing artifacts representing by segments of lines. It would be correct if `Rectangle.this.intersects` returns false for empty rectangle.

#40 - 12/08/2017 05:07 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Yes, and the fact that the month combo box is not repainted with actual value.

See `Combobox.set_Value` and other `set_Value` for i.e. `selection-list` and other widgets - this should call `repaint`, for `COMBO-BOX` it does not.

It seems that the `Rectangle.this.intersection` and `Rectangle.this.intersects` don't consider cases with empty rectangles.

You mean the case where the rectangle has the width or height 0 (it's actually a single line)? I think these edge cases need to be treated by `intersects`... please try to fix it.

It would be correct if `Rectangle.this.intersects` returns false for empty rectangle.

What if the rectangle is a single point? What's your definition of empty rectangle?

#41 - 12/08/2017 05:10 AM - Sergey Ivanovskiy

Yes I mean NativeRectangle, but Rectangle has the same issue. To clarify this bug with empty rectangles let us consider the following empty rectangle NativeRectangle : rect=[left=1, top=0, right=0, bottom=100] given by the coordinates of its sides. It follows that rect.width() = 0 and rect.empty() = true if it will be passed to fill rectangle driver api, then the vertical line will appear on the screen.

```
/**
 * Check if rectangle is empty.
 *
 * @return true if rectangle is empty.
 */
public boolean empty()
{
    return top > bottom || right < left;
}
```

#42 - 12/08/2017 05:13 AM - Sergey Ivanovskiy

I meant the existing definition of NativeRectangle.this.empty() which is correct.

#43 - 12/08/2017 05:16 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

I meant the existing definition of NativeRectangle.this.empty() which is correct.

I don't understand; the coordinate Axis goes from top-left corner, increasing to the left (for X) and to bottom (for Y). So if top > bottom or right < left, this rectangle is actually invalid, as the coordinates don't make sense (it has a negative width or height).

What is your proposed change here?

#44 - 12/08/2017 05:21 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

See ComboBox.setValue and other setValue for i.e. selection-list and other widgets - this should call repaint, for COMBO-BOX it does not.

OK, I will try to fix it.

It would be correct if Rectangle.this.intersects returns false for empty rectangle.

What if the rectangle is a single point? What's your definition of empty rectangle?

It is OK if it the rectangle is a single point or a segment. A rectangle rect is empty if `rect.empty()==true`.

#45 - 12/08/2017 05:22 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Constantin Asofiei wrote:

See `ComboBox.setValue` and other `setValue` for i.e. `selection-list` and other widgets - this should call `repaint`, for `COMBO-BOX` it does not.

OK, I will try to fix it.

Keep in mind to call `repaint()` only if there was actually a change in the widget's UI state.

#46 - 12/08/2017 05:26 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

I meant the existing definition of `NativeRectangle.this.empty()` which is correct.

I don't understand; the coordinate Axis goes from top-left corner, increasing to the left (for X) and to bottom (for Y). So if `top > bottom` or `right < left`, this rectangle is actually invalid, as the coordinates don't make sense (it has a negative width or height).

I changed coordinates in [#3394-41](#). The rectangle is empty and is invalid but if it is passed to the driver, then the line will be on the screen since its width is zero and valid and the driver or upper api don't check if it is empty or not. I encountered such cases in 3394a.

What is your proposed change here?

To discard empty rectangles when getting clipping regions.

#47 - 12/08/2017 05:27 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

I changed coordinates in [#3394-41](#). The rectangle is empty and is invalid but if it is passed to the driver, then the line will be on the screen since its width is zero and valid and the driver or upper api don't check if it is empty or not. I encountered such cases in 3394a.

What is your proposed change here?

To discard empty rectangles when getting clipping regions.

This seems reasonable, please try it.

#48 - 12/08/2017 08:39 AM - Sergey Ivanovskiy

But it is possible that these empty clipping rectangles have been incorrectly added to ScreenBitmap. It needs to find root causes code. It is related to this [#3392](#) (the current rev 11224 in 3394a).

The following changes can help to fix combo box updates

```
=== modified file 'src/com/goldencode/p2j/ui/client/ComboBox.java'
--- src/com/goldencode/p2j/ui/client/ComboBox.java      2017-11-25 22:16:05 +0000
+++ src/com/goldencode/p2j/ui/client/ComboBox.java      2017-12-08 12:35:12 +0000
@@ -664,6 +664,8 @@

    ComboBoxModel<String> model = model();

+   boolean repaint = false;
+   for (int i = 0; i < items.length; i++)
+   {
+       if (items[i].getValue().toStringMessage().compareToIgnoreCase(value.toStringMessage()) == 0)
@@ -671,13 +673,21 @@
+       initValue = items[i].getValue();

-       String text = items[i].getLabel().toStringMessage();
-       model.select(formatValue(text));
+       String oldSelected = model.selected();
+       String item = formatValue(text);
+       model.select(item);
+       model.setSelectedIndex(i);

+       repaint = !item.equals(oldSelected);
+       break;
+   }
+   if (repaint)
+   {
+       repaint();
+   }
```

```
}  
/**
```

but I found another places in the code that should do such updates

```
com.goldencode.p2j.ui.client  
ComboBox<O extends OutputManager<?>>  
valueChangedImpl()  
com.goldencode.p2j.ui.client.gui  
ComboBoxGuiImpl  
setNewSelection(int)
```

#49 - 12/08/2017 08:50 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

The following changes can help to fix combo box updates
[...]
but I found another places in the code that should do such updates

setNewSelection needs to be fixed; valueChangedImpl already has repaint. Please look into other widgets, too, for setValue related APIs (selection-list, toggle-box, radio-set especially).

#50 - 12/08/2017 12:11 PM - Sergey Ivanovskiy

Please clarify what is incorrect in setNewSelection?

I traced ScreenBitmap.this.setState and found many cases with empty rectangles. They can affect the presentation

```
SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.  
SLF4J: Actual binding is of type [org.slf4j.impl.JDK14LoggerFactory]  
Empty rectangle: outer=NativeRectangle[top=0, left=0, bottom=-1, right=-1]  
UI Theme successfully changed to 'windows10'  
Empty rectangle: outer=NativeRectangle[top=0, left=0, bottom=-1, right=-1]  
Empty rectangle: outer=NativeRectangle[top=1, left=1, bottom=0, right=0]  
Empty rectangle: outer=NativeRectangle[top=1, left=1, bottom=0, right=400]  
Empty rectangle: r=NativeRectangle[top=1, left=1, bottom=0, right=0]  
Empty rectangle: r=NativeRectangle[top=1, left=1, bottom=0, right=400]
```

.....

#51 - 12/08/2017 03:22 PM - Sergey Ivanovskiy

Please review committed revision 11249 (3394a) that should discard empty rectangles from SceneBitmap. It seems that SceneBitmap.this.resetScreen can add empty rectangles in the trunc version and empty SceneBitmap.this.outerRectangle.

#52 - 12/08/2017 04:20 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Please review committed revision 11249 (3394a) that should discard empty rectangles from SceneBitmap. It seems that SceneBitmap.this.resetScreen can add empty rectangles in the trunc version and empty SceneBitmap.this.outerRectangle.

Please move in NativeRectangle methods the isEmpty() check at the begining. I think we need to make similar changes for Rectangle, too.

Does this solve the issues you found in Hotel GUI?

Nice work!

#53 - 12/08/2017 04:25 PM - Sergey Ivanovskiy

Constantin, no, updates for combo boxes and tree and another widgets are different issues. I didn't solve them. This fix was just to remove empty rectangles that can be added by resetScreen.

#54 - 12/11/2017 04:51 AM - Sergey Ivanovskiy

This function of OutputManager

```
public ScreenBitmap getAndResetBitmap(int windowId)
{
    ScreenBitmap bitmap = screenBitmap(windowId);

    ScreenBitmap copy = bitmap.getCopy();

    bitmap.addRectangle(null);

    return copy;
}
```

uses unclear logic bitmap.addRectangle(null); to reset screen if bitmap.outerRectangle != null and then sets bitmap.hasRectangles = true; but this function is described as

```
* Get a copy of the current screen bitmap and reset clipping entirely. All
* screen positions become available for output. The previous state can be
* restored using setBitmap() method.
```

It doesn't use bitmap.resetScreen(); that it has a clear meaning.

#55 - 12/11/2017 05:16 AM - Sergey Ivanovskiy

It looks like a bug that for a date picker button defined by

```
subscript (lhButton, iCount).unwrapSizeable().setWidthPixels(new integer(30));
subscript (lhButton, iCount).unwrapSizeable().setHeightPixels(new integer(20));
```

the system generates repaint events with update rectangles that are less one pixel in width and height simultaneously 29x19.

```
PaintEvent: source=ButtonGuiImpl id=354 rect=NativeRectangle[top=134, left=10, bottom=153, right=39]
PaintEvent: source=ButtonGuiImpl id=382 rect=NativeRectangle[top=156, left=10, bottom=175, right=39]
PaintEvent: source=ButtonGuiImpl id=382 rect=NativeRectangle[top=156, left=10, bottom=175, right=39]
PaintEvent: source=ButtonGuiImpl id=354 rect=NativeRectangle[top=134, left=10, bottom=153, right=39]
PaintEvent: source=ButtonGuiImpl id=354 rect=NativeRectangle[top=134, left=10, bottom=153, right=39]
PaintEvent: source=ButtonGuiImpl id=358 rect=NativeRectangle[top=134, left=42, bottom=153, right=71]
PaintEvent: source=ButtonGuiImpl id=358 rect=NativeRectangle[top=134, left=42, bottom=153, right=71]
```

#56 - 12/11/2017 08:44 AM - Sergey Ivanovskiy

Constantin, you are correct that AbstractWidget.afterConfigUpdateBase (trunc version) looks incorrect because of public Rectangle(double top, double left, double bottom, double right)

```
public <C extends WidgetConfig> void afterConfigUpdateBase(C beforeUpdate)
{
    C c = config();

    .....
    if (c instanceof BaseConfig)
    {
        BaseConfig bc = (BaseConfig) c;
        BaseConfig beforeConfig = (BaseConfig) beforeUpdate;

        if (Double.compare(bc.column, beforeConfig.column) != 0 ||
            Double.compare(bc.row, beforeConfig.row) != 0 ||
            Double.compare(bc.widthChars, beforeConfig.widthChars) != 0 ||
            Double.compare(bc.heightChars, beforeConfig.heightChars) != 0)
        {
            // use the old coordinates to repaint
            Rectangle bounds = bounds();
            bounds = bounds.translate(beforeConfig.column - bc.column, beforeConfig.row - bc.row);

            double dwidth = beforeConfig.widthChars - bc.widthChars;
            double dheight = beforeConfig.heightChars - bc.heightChars;
            Point unit = screen().coordinates().baseUnits();
            bounds = new Rectangle(bounds.top(),
                                   bounds.left(),
                                   bounds.width(unit) + dwidth,
                                   bounds.height(unit) + dheight);

            .....
        }

        .....
    }
}
```

But if you fix this formula to

```
bounds = new Rectangle(bounds.top(),
    bounds.left(),
    bounds.height(unit) + dheight,
    bounds.width(unit) + dwidth);
```

then the tree widget and the date picker issues are still present. I tried to change this unclear formula to this one without success.

```
double dwidth = beforeConfig.widthChars - bc.widthChars;
double dheight = beforeConfig.heightChars - bc.heightChars;
Point unit = screen().coordinates().baseUnits();
Point p = location().translate(beforeConfig.column - bc.column, beforeConfig.row - bc.row);
Dimension d = dimension().enlargedBy(new Insets(0, 0, dheight, dwidth));
Rectangle bounds2 = new Rectangle(p, d, unit);
```

These bounds2 and bound are different rectangles. We should discuss here the correct formula for the update rectangle.

#57 - 12/11/2017 09:23 AM - Sergey Ivanovskiy

This trace should be discussed too

```
PaintEvent: source=WindowTitleBar id=-49 rect=NativeRectangle[top=1, left=1, bottom=30, right=0]
java.lang.Exception
    at com.goldencode.p2j.ui.client.event.PaintEvent.<init>(PaintEvent.java:108)
    at com.goldencode.p2j.ui.client.widget.AbstractWidget.repaint(AbstractWidget.java:980)
    at com.goldencode.p2j.ui.client.widget.AbstractWidget.repaint(AbstractWidget.java:915)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.repaint(AbstractContainer.java:1058)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.setSize(AbstractContainer.java:1236)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.setWidth(AbstractContainer.java:1256)
    at com.goldencode.p2j.ui.client.gui.WindowLayout.layoutRegular(WindowLayout.java:351)
    at com.goldencode.p2j.ui.client.gui.WindowLayout.doLayout(WindowLayout.java:213)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.doLayout(AbstractContainer.java:411)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.doLayout(WindowGuiImpl.java:561)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.lambda$doResizeTo$6(WindowGuiImpl.java:2621)
    at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:15525)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:15469)
    at com.goldencode.p2j.ui.chui.ThinClient.independentEventDrawingBracket(ThinClient.java:15339)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.doResizeTo(WindowGuiImpl.java:2620)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.resizeTo(WindowGuiImpl.java:1004)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.afterConfigUpdate(WindowGuiImpl.java:1495)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.afterConfigUpdate(WindowGuiImpl.java:232)
    at com.goldencode.p2j.ui.ConfigSyncManager.markScopeEnd(ConfigSyncManager.java:299)
    at com.goldencode.p2j.ui.ConfigManager.syncConfigChanges(ConfigManager.java:549)
```



```

at com.goldencode.p2j.ui.client.Window$3.run(Window.java:846)
at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:15525)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:15469)
at com.goldencode.p2j.ui.client.Window.pushConfig(Window.java:832)
at com.goldencode.p2j.ui.chui.ThinClient.pushWindow(ThinClient.java:8717)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:124)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:757)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:412)
at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:348)
at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1170)
at com.goldencode.p2j.net.Queue.transact(Queue.java:641)
at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
at com.sun.proxy.$Proxy4.standardEntry(Unknown Source)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:361)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:158)
at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:250)
at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)

```

```

PaintEvent: source=FrameGuiImpl id=88 rect=NativeRectangle[top=31, left=1, bottom=30, right=440]
java.lang.Exception

```

```

at com.goldencode.p2j.ui.client.event.PaintEvent.<init>(PaintEvent.java:108)
at com.goldencode.p2j.ui.client.widget.AbstractWidget.repaint(AbstractWidget.java:980)
at com.goldencode.p2j.ui.client.gui.FrameGuiImpl.repaintCaption(FrameGuiImpl.java:1591)
at com.goldencode.p2j.ui.client.gui.FrameGuiImpl.access$000(FrameGuiImpl.java:183)
at com.goldencode.p2j.ui.client.gui.FrameGuiImpl$1.onFocusLost(FrameGuiImpl.java:457)
at com.goldencode.p2j.ui.client.event.FocusEvent.dispatch(FocusEvent.java:119)
at com.goldencode.p2j.ui.client.widget.AbstractWidget.processFocusEvent(AbstractWidget.java:746)
at com.goldencode.p2j.ui.client.widget.AbstractWidget.processEvent(AbstractWidget.java:1619)
at com.goldencode.p2j.ui.client.widget.TitledWindow.processEvent(TitledWindow.java:268)
at com.goldencode.p2j.ui.client.gui.ModalWindow.processEvent(ModalWindow.java:253)
at com.goldencode.p2j.ui.client.gui.DialogBoxWindow.processEvent(DialogBoxWindow.java:325)

```

and these logs of the calculated update rectangles([#3394-56](#))

```

d=19 bounds=Rectangle[top=0.0, left=0.0, bottom=24.57, right=32.0] bounds2=Rectangle[top=0.0, left=0.0, bottom=24.52, right=31.8]
id=19 bounds=Rectangle[top=0.0, left=0.0, bottom=25.19, right=128.8] bounds2=Rectangle[top=0.0, left=0.0, bottom=25.14, right=128.6]
id=10 bounds=Rectangle[top=-2.147483648E9, left=-2.147483648E9, bottom=2.1, right=41.4] bounds2=Rectangle[top=-2.147483648E9, left=-2.147483648E9, bottom=-2.14748364595E9, right=-2.1474836068E9]
id=10 bounds=Rectangle[top=0.0, left=0.0, bottom=2.1, right=41.4] bounds2=Rectangle[top=0.0, left=0.0, bottom=2.05, right=41.2]
id=10 bounds=Rectangle[top=11.43, left=0.0, bottom=2.1, right=41.4] bounds2=Rectangle[top=11.43, left=0.0, bottom=13.48, right=41.2]

```

#58 - 12/11/2017 02:23 PM - Constantin Asofiei

Sergey, when you mentioned the `afterConfigUpdateBase`, it got me thinking... we need to post physical boundaries and not relative, plus some other changes. See 3394a revisions 11252 and 11253. The old approach was completely wrong... for multiple reasons:

- the comparison of old/new boundaries was incorrect
- the posting was done via `repaint(Rectangle)` which in some cases was dropping the `PaintEvent`, as it didn't intersect with the current parent boundary
- the boundary was computed incorrect

It should be fixed now, please try to break it a little. If it works, I want you to continue with these cases (just a starting point):

- `ComboBox.refreshItems`, `setValue`, `valueChangedImpl()` - these should call `repaint()` only if the displayed text has changed
- `FillIn.setEnabled()` - call `repaint()` only if previous state was not enabled
- `RadioButton.setLocation`, `SelectionListBody.setLocation` - repaint only if location has changed
- `RadioSet.afterConfigUpdate` - repaint only if horizontal flag has changed
- `RadioSet.refreshItemsImpl` - call `repaint()` only if something has changed

You can do a search in Eclipse for `Widget.repaint()` references and see other obvious cases where `repaint()` is called too aggressively; don't spend too much on these, just what you can solve tonight/tomorrow morning; after that, you can move to the Web-related bugs.

#59 - 12/11/2017 02:32 PM - Sergey Ivanovskiy

OK. Did you try to draw sequence diagrams related to drawings? It seems there are different scenarios that can invoke `OutputManager.setInvalidate`. It seems there are gaps in these scenarios because there are no obvious scenario, for an example the `DataContainer` on `setValue`, and the button widget on `setText`. It seems that MVC can help to fix them if all scenarios will be taken into account. I have no such vision now.

#60 - 12/11/2017 02:37 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

OK. Did you try to draw sequence diagrams related to drawings? It seems there are different scenarios that can invoke `OutputManager.setInvalidate`. It seems there are gaps in these scenarios because there are no obvious scenario, for an example the `DataContainer` on `setValue`, and the button widget on `setText`.

No, but `repaint()` should be called only by APIs (or cases) when widget's UI state has changed (i.e. enabled/disabled, visible/hidden, moved, resized, content change, bg/fgcolor, etc).

These cases worked previously because we were always drawing everything - so code which explicitly should have called `repaint()` was missed. When we find a case like `setValue()` it's better to re-check every definition, as most likely more than one case was missed.

It seems that MVC can help to fix them if all scenarios will be taken into account. I have no such vision now.

Yes, but it's pretty complex to ensure everything is standardized... for now, please check the `setValue` and `setText` APIs (on all widgets) if they call `repaint()` properly (only if something has changed).

#61 - 12/11/2017 02:56 PM - Sergey Ivanovskiy

Constantin, your changes fixed the tree control and all date picker issues. Thank you for help. There are only visible screen changes during the repaints on "model" changes but it is another issue. Planning to check another widgets tomorrow at the morning. I would like that new task to apply MVC approach to the current design was created.

#62 - 12/12/2017 04:48 AM - Sergey Ivanovskiy

In the current version of 3394a (rev 11256) there is at least one known regression related to the combo box dropdown focus changes on TAB/SHIFT-TAB. If we press TAB on the opened dropdown, then the next focused widget is incorrectly detected.

#63 - 12/12/2017 05:14 AM - Sergey Ivanovskiy

- File *EmptyRectanglesFromRepaint.txt* added

Constantin, empty rectangles are generated from size or another updates on widget.repaint(). Please look at this log, where PaintEvent events having empty update rectangles are logged.

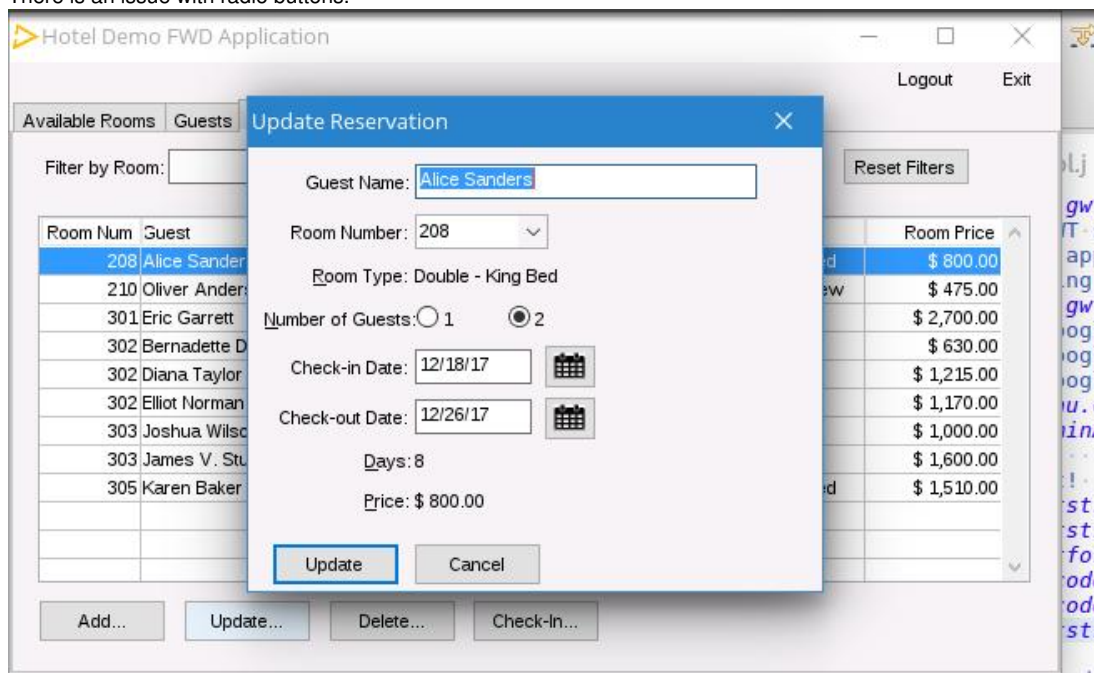
#64 - 12/12/2017 06:05 AM - Sergey Ivanovskiy

Please review these minor changes in the committed revision 11257 that discarded PainEvents with empty update rectangles.

#65 - 12/12/2017 07:02 AM - Sergey Ivanovskiy

- File *RadioButtonsGroup.png* added

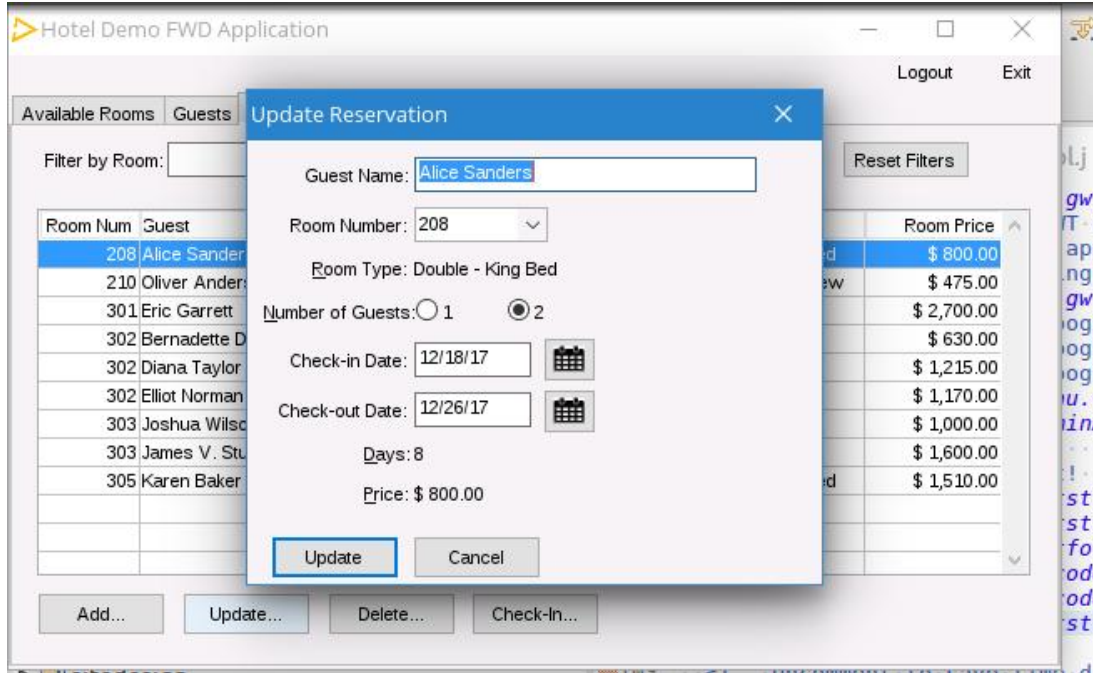
There is an issue with radio buttons.



#66 - 12/12/2017 07:15 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

There is an issue with radio buttons.



Please describe it.

The changes in 3394a rev 11257 look OK.

#67 - 12/12/2017 07:24 AM - Sergey Ivanovskiy

I am just looking into it but if we open Update Reservation for rooms with 3 and 4 guests, then this dialog displays 4 radio buttons (items), and if we open reservations for rooms with 2 guests, then there are only 2 displayed buttons (initially 4 items are present).

#68 - 12/12/2017 07:28 AM - Stanislav Lomany

Quantity of radio buttons should match the maximum number of guests for the selected room type. So far I see no issues.

#69 - 12/12/2017 07:30 AM - Sergey Ivanovskiy

OK, thank you.

#70 - 12/12/2017 08:44 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

In the current version of 3394a (rev 11256) there is at least one known regression related to the combo box dropdown focus changes on TAB/SHIFT-TAB. If we press TAB on the opened dropdown, then the next focused widget is incorrectly detected.

Is this issue known? It looks that some last changes break the focus traversal. The focus moves to the parent and to the first widget in the container. Should I take this task?

#71 - 12/12/2017 09:26 AM - Greg Shah

In the current version of 3394a (rev 11256) there is at least one known regression related to the combo box dropdown focus changes on TAB/SHIFT-TAB. If we press TAB on the opened dropdown, then the next focused widget is incorrectly detected.

Is this issue known? It looks that some last changes break the focus traversal. The focus moves to the parent and to the first widget in the container. Should I take this task?

Yes, you can take it.

First, please finish the work listed by Constantin in [#3394-58](#) and [#3394-60](#).

#72 - 12/12/2017 09:42 AM - Sergey Ivanovskiy

OK. Related to the Hotel GUI I didn't detect artifacts except the focus traversal chain started from the opened dropdown was broken.

#73 - 12/13/2017 08:35 AM - Sergey Ivanovskiy

It seems that the root cause of the combo box issue is the current focus window is still a drop down overlay when the current focus traversal flow is finished and the current work point is in `ThinClient.waitForWorker` and `FM.adjustFocus()` is called. Working on the fix.

#74 - 12/13/2017 09:03 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

It seems that the root cause of the combo box issue is the current focus window is still a drop down overlay when the current focus traversal flow is finished and the current work point is in `ThinClient.waitForWorker` and `FM.adjustFocus()` is called. Working on the fix.

Please review this fix, committed revision 11261. The issue is that the current top window becomes focused window later after the TAB/SHIFT-TAB keys are processed and the focus traversal chain has been set. Probably, there are different fixes. But this change looks reasonable.

#75 - 12/13/2017 09:22 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Sergey Ivanovskiy wrote:

It seems that the root cause of the combo box issue is the current focus window is still a drop down overlay when the current focus traversal flow is finished and the current work point is in `ThinClient.waitForWorker` and `FM.adjustFocus()` is called. Working on the fix.

Please review this fix, committed revision 11261. The issue is that the current top window becomes focused window later after the TAB/SHIFT-TAB keys are processed and the focus traversal chain has been set. Probably, there are different fixes. But this change looks reasonable.

My concern is this: top level window might not be the active (or current) window; Hynek, can you take a look too, please?

Does the same bug happen on the Web client, too? I ask because I have some more generic fixes related to focus window (which affect Swing mostly), and these are not seen in Web.

Also, you have a copy-paste error on line 34 in `UiUtils`.

#76 - 12/13/2017 09:35 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Does the same bug happen on the Web client, too? I ask because I have some more generic fixes related to focus window (which affect Swing mostly), and these are not seen in Web.

Yes, in the web client too. Please check if your changes can help to fix this issue in which the current top window becomes focused after TAB/SHIFT-TAB keys have been processed. Any case the `UiUtils.getCurrentFocus()` works incorrectly in the considered case, alternatively we can change its usages.

Also, you have a copy-paste error on line 34 in `UiUtils`.

Thanks, it was fixed in 11262.

#77 - 12/15/2017 10:24 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

.. I ask because I have some more generic fixes related to focus window (which affect Swing mostly), and these are not seen in Web.

Constantin, when do you plan to commit these changes? I found that after deactivation window message, the system sent the "to place window on top" message. The last message impacts the activation task bar buttons in the virtual desktop mode. I can postpone fixing this until you commit your generic changes.

#78 - 12/15/2017 01:00 PM - Sergey Ivanovskiy

I think that this implementation `WindowGuiImpl.iconify(postEvent)` to iconify window is incorrect

```
@Override
public void iconify(boolean postEvent)
{
    if (isMinimized())
    {
        return;
    }

    saveWindowState();
}
```

```

WindowConfig cfg = config();
config.windowState = windowState = CommonWindow.WINDOW_STATE_MINIMIZED;

if (config.showInTaskbar)
{
    gd.iconifyWindow(getId().asInt());
}
else
{
    updateTitleBar();
    doLayout();
    repaint();
}

if (postEvent)
{
    WindowEvent evt = new WindowEvent(this, Keyboard.SE_WINDOW_MINIMIZED);
    evt.setOrigin(getId().asInt());
    ThinClient.getInstance().postOSEvent(evt);
}

TopLevelWindow wnd = WindowManager.getNextActiveWindow();
if (wnd != null)
{
    WindowManager.activateWindow(wnd);
}
}

```

It is supposed that the current window is deactivated, but it is not happened at this point and `TopLevelWindow wnd = WindowManager.getNextActiveWindow();` returns the current window.

```

WindowManager.getNextActiveWindow() line: 1627
WindowGuiImpl.iconify(boolean) line: 842
CaptionButton.mouseClicked(MouseEvent) line: 254
MouseHandler.applyMouseEvent(Widget<GuiOutputManager>, MouseEvent) line: 357
MouseHandler.handleMouseEvent(int, MouseEvent) line: 282
GuiWebDriver(AbstractGuiDriver<F>).handleMouseEvent(int, MouseEvent) line: 2827
WindowManager.processWindowEvent(Event, TopLevelWindow<?>) line: 1707
WindowGuiImpl.processEvent(Event) line: 1360

```

Finally it gets the "to place window on top" event.

#79 - 12/15/2017 01:18 PM - Sergey Ivanovskiy

This implementation of `WindowGuiImpl.this.isDesktopMinimized()` is confusing

```
/**
 * Returns true when the window is in the state "desktop-minimized", that is
 * the window is minimized but its SHOW-IN-TASKBAR attribute is set to
 * false.
 *
 * @return See above.
 */
@Override
public boolean isDesktopMinimized()
{
    return windowState == CommonWindow.WINDOW_STATE_MINIMIZED && !config.showInTaskbar;
}
```

and this implementation of `WindowManager.getNextActiveWindow()` is confusing too

```
/**
 * The window calculates the next main window that should eventually
 * become active after the currently active window is deactivated.
 *
 * @return See above.
 */
public static TopLevelWindow<?> getNextActiveWindow()
{
    for (TitledWindow<?> tw : work.get().fixedOrderWindows)
    {
        if (!tw.isVisible() || !(tw instanceof GuiWindow) ||
            ((GuiWindow) tw).isShareActivationWithOwner() ||
            ((GuiWindow) tw).isDesktopMinimized())
        {
            continue;
        }

        return (TopLevelWindow<?>) tw;
    }

    return null;
}
```

I guess that `config.showInTaskbar` is always true for the regular desktop window and false for the overlay window. Correct?

#80 - 12/15/2017 06:24 PM - Constantin Asofiei

3394a rev 11269 was rebased with trunk rev 11210. 3394a new rev is 11271.

#81 - 12/17/2017 04:52 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

I guess that `config.showInTaskbar` is always true for the regular desktop window and false for the overlay window. Correct?

Sergey, `config.showInTaskbar` will be false for regular desktop window when `WINDOW:SHOW-IN-TASKBAR` is set to `FALSE`. An overlay window should never be displayed on the task bar.

#82 - 12/18/2017 12:28 PM - Sergey Ivanovskiy

Please review committed revision 11275 - tried to fix activations of task icons in the virtual desktop.

#83 - 12/18/2017 06:29 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Constantin Asofiei wrote:

.. I ask because I have some more generic fixes related to focus window (which affect Swing mostly), and these are not seen in Web.

Constantin, when do you plan to commit these changes? I found that after deactivation window message, the system sent the "to place window on top" message. The last message impacts the activation task bar buttons in the virtual desktop mode. I can postpone fixing this until you commit your generic changes.

My focus related changes (together with the browse event changes) are in 3394a 11278.

#84 - 12/19/2017 08:26 AM - Greg Shah

Sergey Ivanovskiy wrote:

Please review committed revision 11275 - tried to fix activations of task icons in the virtual desktop.

Code Review Task Branch 3394a Revision 11275

I'm OK with the changes.

Hynek: could you look at that revision please?

#85 - 12/19/2017 11:10 AM - Hynek Cihlar

Greg Shah wrote:

Sergey Ivanovskiy wrote:

Please review committed revision 11275 - tried to fix activations of task icons in the virtual desktop.

Code Review Task Branch 3394a Revision 11275

I'm OK with the changes.

Hynek: could you look at that revision please?

The changes look ok to me. Just a small suggestion. Sergey, please revert the changes in WindowManager and instead change isDesktopMinimized() to isMinimized() in getNextActiveWindow(). This should yield the same behavior and simplify the class interface.

#86 - 12/19/2017 02:16 PM - Sergey Ivanovskiy

OK, please review the committed revision 11282.

#87 - 12/19/2017 02:47 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

OK, please review the committed revision 11282.

The changes look good.

#88 - 12/21/2017 07:22 AM - Hynek Cihlar

3394a revision 11294 fixes browse column widths due to incorrectly resolved column font.

#89 - 12/22/2017 08:33 AM - Constantin Asofiei

Rebased 3394a from trunk rev 11213 - new revision 11304.

3394a rev 11305 contains javadoc and history header fixes.

#90 - 12/22/2017 11:34 AM - Greg Shah

I've tested 3394a revision 11307 with Hotel GUI. It fixes the abend (Check-In then Add) and it fixes the tree control drawing issues, both of which are present in trunk.

Overall this is a big improvement over trunk and we can go ahead with merging to trunk if the ChUI regression testing and the large GUI application testing passes.

#91 - 12/25/2017 04:03 PM - Eric Faulhaber

There is a regression in task branch 3394a. ETF search testing grinds to a halt after a few minutes with multiple "No servers available" messages. Based on the server log, this seems to be related to the shared variable manager changes. Logs are at #1868-84.

#92 - 12/26/2017 04:06 AM - Constantin Asofiei

Eric Faulhaber wrote:

There is a regression in task branch 3394a. ETF search testing grinds to a halt after a few minutes with multiple "No servers available" messages. Based on the server log, this seems to be related to the shared variable manager changes. Logs are at #1868-84.

Please try 3394a 11309 - I think I had an unbalanced scope start/delete in SVM.

#93 - 12/26/2017 10:42 AM - Eric Faulhaber

Constantin Asofiei wrote:

Please try 3394a 11309 - I think I had an unbalanced scope start/delete in SVM.

That did it! All search tests passed. Will update later for remaining ETF categories.

#94 - 12/26/2017 03:47 PM - Eric Faulhaber

3394a/11309 has passed ETF full search, developer, and batch tests.

#95 - 12/27/2017 09:29 PM - Greg Shah

I've done one run of main ChUI regression testing. 6 unexpected failures occurred, but they all seem like false negatives. I'm running another round now (though the time is not great for the job clock tests).

#96 - 12/28/2017 03:17 AM - Greg Shah

The next run of ChUI regression completed. Only 2 failures overlapped. Both of those are likely false negatives, time/job clock related. I'm rerunning now.

#97 - 12/28/2017 09:13 AM - Greg Shah

The only overlap this time is tc_job_clock_004 which was expected to fail due to the time of the run. At this point I'm going to call this a pass. I will merge to trunk shortly.

#98 - 12/28/2017 10:57 AM - Greg Shah

Task branch 3394a has been merged to trunk as revision 11214.

I've archived 3394a.

#99 - 12/28/2017 10:57 AM - Greg Shah

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

#100 - 01/03/2018 02:03 PM - Greg Shah

- Related to Feature #3246: reduce the amount of data being sent to the client-side when an UI attribute is being changed added

Files

resizable.png	39.5 KB	12/07/2017	Sergey Ivanovskiy
date_picker.mkv	1.73 MB	12/08/2017	Sergey Ivanovskiy
EmtyRectanglesFromRepaint.txt	73.9 KB	12/12/2017	Sergey Ivanovskiy
RadioButtonsGroup.png	66 KB	12/12/2017	Sergey Ivanovskiy