User Interface - Bug #2832

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

GUI combo-box simple mode is broken

11/09/2015 07:38 PM - Hynek Cihlar

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Eugenie Lyzenko	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	GUI Support for a Complex ADM2 App		
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 11/09/2015 07:40 PM - Hynek Cihlar

In SIMPLE mode ComboBoxGuiImpl creates multiple sibling widgets (a FillIn entry field and a selection list) on the same level of widget tree hierarchy. These widgets are physically disconnected, but logically belong to the combo-box widget. ComboBoxGuiImpl attempts to manage these disconnected widgets as one logical piece but it brakes on other places where this structure is rightfully unexpected. An example is ENABLE ALL statement where the number of focusable widgets is greater than the expected. Other places where this breaks is during frame layout, size reporting, etc.

The proper solution would be (at least for the SIMPLE mode) to have the combo-box widget class inherit from AbstractContainer and attach all the individual widgets (entry field and selection list) to the container.

#2 - 11/09/2015 07:41 PM - Hynek Cihlar

DROP-DOWN mode is affected in the same way. In DROP-DOWN mode the fill-in entry field is used the same way as in SIMPLE.

#3 - 11/27/2015 02:16 PM - Greg Shah

- Start date deleted (11/09/2015)

- Assignee set to Eugenie Lyzenko

- Target version set to Milestone 12

#4 - 12/01/2015 08:41 PM - Eugenie Lyzenko

The investigation shows the SIMPLE mode internal logic is almost all broken. I've recovered most of all but not yet completed.

My plan is first to recover complete functionality that worked before. Then we can think whether we need to completely rework the combo-box GUI mode widget to be a kind of AbstractContainer or not.

#5 - 12/02/2015 02:39 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> The investigation shows the SIMPLE mode internal logic is almost all broken. I've recovered most of all but not yet completed.
>
> My plan is first to recover complete functionality that worked before. Then we can think whether we need to completely rework the combo-box GUI mode widget to be a kind of AbstractContainer or not.

Eugenie, if you recover the original design, the implementation will be still broken. Multiple places in the framework do not expect one widget to span multiple Widget implementations. There will be abends and unexpected behaviors.

#6 - 12/02/2015 04:06 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

> Eugenie, if you recover the original design, the implementation will be still broken. Multiple places in the framework do not expect one widget to span multiple Widget implementations. There will be abends and unexpected behaviors.

I do not expect everything will be working fine after recover. But to estimate the efforts I need to exactly know what and where is wrong and to select the best approach to implement. Moreover the FillInGuiImpl has some changes that out of the scope of this point and are the issues itself. We will have to fix it anyway for every class that will be inherited from FillInGuiImpl.

#7 - 12/06/2015 05:42 PM - Eugenie Lyzenko

The task 2832a has been created based on trunk revision 10956.

#8 - 12/07/2015 01:42 PM - Eugenie Lyzenko

The task branch 2832a updated for review at revision 10957.

The combo-box functionality restored in SIMPLE mode. There are some remaining GUI issues, working on, so the code is not cleaned from comments. But the basic implementation is here. Yes, there are two issues with this approach.
1. Early painting causes NPE.
2. Size calculation to be used by layout manager is dependent on the mode the combo-box is running. The SIMPLE mode has special approach, where inner-lines attribute is ignoring. We have to always set the size explicitly(4GL does not allow auto calculation based on inner-lines attribute).

So all two above issues are now fixed. So if there are no objection I would prefer to stay with this implementation to not over-complicate the widget. Or I need to know the cases when this approach will fail. Using AbstractContainer gives the same issues to resolve I guess.

#9 - 12/07/2015 02:04 PM - Eugenie Lyzenko

Rebased task branch 2832a from P2J trunk revision 10957, new branch revision is 10958.

#10 - 12/07/2015 03:41 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Or I need to know the cases when this approach will fail.

As an example, you can test the ENABLE ALL case as mentioned in note 1. Create a frame with a combo box in simple mode and use ENABLE ALL on the frame.

#11 - 12/07/2015 05:46 PM - Eugenie Lyzenko

- File cbb_test10_3_1_p2j_gui_20151207.jpg added

> As an example, you can test the ENABLE ALL case as mentioned in note 1. Create a frame with a combo box in simple mode and use ENABLE ALL on the frame.

The next update for review - task branch 2832a at revision 10959. This solves the point #1 completely plus additional fixes for GUI drawing. The testcases for combo-box updated the revision 1438. The test you mentioned is combo_box/combo_box10_3_1.p. The screen shot is attached to demonstrate proper enable handling and size usage in frame layout manager utility.

The issue remaining is initial value set in scrollable list at start up. Working on this.

#12 - 12/07/2015 06:22 PM - Eugenie Lyzenko

- File cbb_test10_3_p2j_gui_20151207.jpg added

Task branch 2832a for review updated to revision 10960.

This is the initial value set fix for scrollable list. The screen shot for working test is attached.

Next planning step is to check the DROP-DOWN mode is working properly.

#13 - 12/08/2015 06:14 AM - Hynek Cihlar

I checked the problematic cases and the SIMPLE mode works fine.

I just noticed an abend when a simple-mode selection list item is clicked, see the exception callstack below. This can be reproduced in demo/demo_widgets.p, just convert the combo-box there to simple mode.

```
Thread [main] (Suspended (exception StringIndexOutOfBoundsException))
    String.substring(int, int) line: 1951
    EntryFieldGuiImpl(FillInGuiImpl).drawSelectionInt(String, int, ColorRgb, ColorRgb) line: 1012
    EntryFieldGuiImpl.drawSelection(String, int, ColorRgb, ColorRgb) line: 302
    FillInGuiImpl$2$2.run() line: 869
    SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
    FillInGuiImpl$2.run() line: 857
    SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
    EntryFieldGuiImpl(FillInGuiImpl).draw(Color) line: 804
    EntryFieldGuiImpl(FillIn<O,C>).draw() line: 1074
    SensitiveScrollContainer<O>(AbstractContainer<O>).draw() line: 410
    Viewport<O>.draw() line: 64
    BorderedPanelGuiImpl(AbstractContainer<O>).draw() line: 410
    BorderedPanelGuiImpl.access$4(BorderedPanelGuiImpl) line: 1
    BorderedPanelGuiImpl$1$1.run() line: 200
    SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
    BorderedPanelGuiImpl$1.run() line: 189
```

```
SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
BorderedPanelGuiImpl.draw() line: 136
ScrollPaneGuiImpl$2.run() line: 159
SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
ScrollPaneGuiImpl.draw() line: 150
FrameGuiImpl$2.run() line: 443
SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
FrameGuiImpl.draw(Point, Dimension) line: 429
FrameGuiImpl(Frame<O>).draw() line: 1903
NativeScrollContainer<O>(AbstractContainer<O>).draw() line: 410
Viewport<O>.draw() line: 64
BorderedPanelGuiImpl(AbstractContainer<O>).draw() line: 410
BorderedPanelGuiImpl.access$4(BorderedPanelGuiImpl) line: 1
BorderedPanelGuiImpl$1$1.run() line: 200
SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
BorderedPanelGuiImpl$1.run() line: 189
SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
BorderedPanelGuiImpl.draw() line: 136
ScrollPaneGuiImpl$2.run() line: 159
SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
WindowWorkSpace(ScrollPaneGuiImpl).draw() line: 150
WindowWorkSpace.draw() line: 87
BorderedPanelGuiImpl(AbstractContainer<O>).draw() line: 410
BorderedPanelGuiImpl.access$4(BorderedPanelGuiImpl) line: 1
BorderedPanelGuiImpl$1$1.run() line: 200
SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
BorderedPanelGuiImpl$1.run() line: 189
SwingGuiDriver(AbstractGuiDriver<F>).draw(NativePoint, NativeRectangle, Runnable) line: 1988
BorderedPanelGuiImpl.draw() line: 136
WindowGuiImpl(Window<O>).draw() line: 1373
WindowGuiImpl.draw() line: 469
GuiOutputManager(OutputManager<P>).setInvalidate(Widget<?>, boolean, boolean) line: 1275
ThinClient.eventDrawingBracket(Widget<?>, boolean, boolean, Runnable) line: 13840
MouseHandler.handleMouseEvent(int, MouseEvent) line: 248
SwingGuiDriver(AbstractGuiDriver<F>).handleMouseEvent(int, MouseEvent) line: 2393
WindowGuiImpl(TopLevelWindow<O>).processEvent(Event) line: 669
WindowGuiImpl.processEvent(Event) line: 1315
```

Hopefully the non-standard widget structure will not cause more problems in the future.

#14 - 12/08/2015 11:20 AM - Eugenie Lyzenko

Task branch 2832a for review updated to revision 10961.

This is the fix for StringIndexOutOfBoundsException recently reported by Hynek. We need to use screen value to detect the last editable char, not value itself because it can have padding spaces to match the format value.

#15 - 12/08/2015 06:03 PM - Eugenie Lyzenko

Task branch 2832a for review updated to revision 10962.

Several fixes to improve EntryField text painting to match 4GL. So far the implementation for SIMPLE and DROP-DOWN modes completed and if no objection it is ready to be merged into trunk.

#16 - 12/09/2015 07:19 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Task branch 2832a for review updated to revision 10962.
>
> Several fixes to improve EntryField text painting to match 4GL. So far the implementation for SIMPLE and DROP-DOWN modes completed and if no objection it is ready to be merged into trunk.

The code looks ok to me.

Only small thing. I noticed the number of visible items in the simple list differs from 4GL. Is this expected?

#17 - 12/09/2015 07:37 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

> The code looks ok to me.
>
> Only small thing. I noticed the number of visible items in the simple list differs from 4GL. Is this expected?

What the test that shows this diff? Can you provide the both screens here for me to see this?

#18 - 12/09/2015 08:01 AM - Eugenie Lyzenko

> Only small thing. I noticed the number of visible items in the simple list differs from 4GL. Is this expected?

In SIMPLE mode the number of visible items defined by comb-box size and used font. The INNER-LINES attribute is ignoring. So I need to know particular condition for the deviation you see to tell if this is expected or not.

#19 - 12/09/2015 09:22 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> > Only small thing. I noticed the number of visible items in the simple list differs from 4GL. Is this expected?
>
> In SIMPLE mode the number of visible items defined by comb-box size and used font. The INNER-LINES attribute is ignoring. So I need to know particular condition for the deviation you see to tell if this is expected or not.

Try the following code sample.

```
DEF VAR cb AS CHAR FORMAT "x(8)" VIEW-AS COMBO-BOX simple LIST-ITEMS
   "item1", "item2", "item3", "item4", "item5" size 10 by 5.

ENABLE cb WITH FRAME f SIDE-LABELS.
WAIT-FOR GO OF FRAME f.
```

Different sizes produce various size differences between P2J and 4GL.

#20 - 12/09/2015 09:25 AM - Hynek Cihlar

Another issue I just noted is that 4GL seems to ignore the format specification for COMBO-BOX. Although this is not directly related to SIMPLE-LIST, it does affect it. Try the following code sample.

```
DEF VAR cb AS CHAR FORMAT "x(1)" VIEW-AS COMBO-BOX simple LIST-ITEMS
   "item1", "item2", "item3", "item4", "item5" size 10 by 5.
```

```
ENABLE cb WITH FRAME f SIDE-LABELS.
WAIT-FOR GO OF FRAME f.
```

#21 - 12/09/2015 09:50 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

> [...]
>
> Different sizes produce various size differences between P2J and 4GL.

Have you used the USE-3D-SIZE global option as "yes" or "no" in 4gl? In p2j?

#22 - 12/09/2015 10:10 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
> > [...]
> >
> > Different sizes produce various size differences between P2J and 4GL.
>
> Have you used the USE-3D-SIZE global option as "yes" or "no" in 4gl? In p2j?

Tested with USE-3D-SIZE set to yes for both 4GL and P2J.

#23 - 12/09/2015 12:10 PM - Eugenie Lyzenko

- File cbb_test20_1_p2j_3d_on_System_font_20151209.jpg added

- File cbb_test20_1_4gl_3d_on_gui.jpg added

Hynek Cihlar wrote:

> Tested with USE-3D-SIZE set to yes for both 4GL and P2J.

Take a look at the pictures attached. Both use System font and USE-3D-SIZE as yes.

The message area lines contain the height of the combo-boxes in chars(first line) and pixels(second line)

Is it the difference you noted? It will be good if you posted your pictures here for me to compare.

#24 - 12/09/2015 12:23 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Is it the difference you noted?

Yes. Let's take the combo 'cb' from the screen shots. While in P2J the font height in the simple list seems to be greater than in 4GL, the height of the P2J simple list is smaller than in 4GL. I would expect the height of the simple list in P2J proportionally greater.

#25 - 12/09/2015 12:55 PM - Eugenie Lyzenko

> Yes. Let's take the combo 'cb' from the screen shots. While in P2J the font height in the simple list seems to be greater than in 4GL, the height of the P2J simple list is smaller than in 4GL.

I think the font size is the same for both p2j and 4gl, check the height in pixels for both cases.

> I would expect the height of the simple list in P2J proportionally greater.

The problem I see here is the list portion of the P2J test is really smaller than in 4GL. The interesting is the real size for 4GL testcases is one pixel smaller than in HEIGHT-PIXELS attribute. So there is a weird widget height calculation I need to find out to eliminate this change.

The only thing that constant is the height in chars and height in pixels(that is computed from char->pixel conversion). And visible part of the widget cannot be more than height in pixels(but can be smaller, 1 or even 4 pixels).

#26 - 12/09/2015 01:01 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

> The only thing that constant is the height in chars and height in pixels(that is computed from char->pixel conversion).

This is because the conversion uses SESSION:PIXELS-PER-ROW/SESSION:PIXELS-PER-COLUMN.

#27 - 12/09/2015 10:17 PM - Eugenie Lyzenko

- File cbb_test20_1_p2j_fix_20151209.jpg added

Task branch 2832a for review updated to revision 10963.

This is the fix for visible items mismatch. However new issue discovered with this change. Looking at the picture attached the scrollbar is wrong when the visible items are smaller than total. This is caused by the ability to have partial visible item in the list. But scrollbar is defined by visible rows/total rows data. Need to find out more precise pixel based approach to compute scrolling for this case. Continue working.

#28 - 12/10/2015 04:05 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Task branch 2832a for review updated to revision 10963.
>
> This is the fix for visible items mismatch. However new issue discovered with this change. Looking at the picture attached the scrollbar is wrong when the visible items are smaller than total. This is caused by the ability to have partial visible item in the list. But scrollbar is defined by visible rows/total rows data. Need to find out more precise pixel based approach to compute scrolling for this case. Continue working.

IMHO the simplest solution is to change the scrolling unit from lines to pixel-line-height for ScrollableSelectionListGuiImpl. So instead of increment/decrement scroll position by 1 it would be the pixel height of one line. I think it should be enough to override getScrollDimension(), getVisibleDimension() and getStep() in ScrollableSelectionListGuiImpl and to convert the pixel value to rowTop in ScrollableSelectionListGuiImpl.scroll() as y-position/pixel-line-height.

#29 - 12/10/2015 11:20 AM - Eugenie Lyzenko

Task branch 2832a for review updated to revision 10964.

Fixes for partial item drawing. Unfortunately approach you mentioned(and really what I thought about too) to change to pixels does not work good enough. We need the item row logic to draw the list and scrolling. The alternative can be the change scroll dimension related api return values from NativeRectangle(int, int) to Dimension(double, double) and keep measure everything in char units.

Let me know if this update has objections, I'll think for how to rework if this approach is not acceptable.

#30 - 12/10/2015 12:16 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Task branch 2832a for review updated to revision 10964.
>
> Fixes for partial item drawing. Unfortunately approach you mentioned(and really what I thought about too) to change to pixels does not work good enough. We need the item row logic to draw the list and scrolling.

By "item row logic", do you mean that the scroll step should be the equivalent of one row? That is, one click on the scroll bar button should move the scrollable content by one row? If so, this can be achieved even with the pixel units by setting the scroll step (the return value of the overriden method getStep()) to the pixel height of one row.

> Let me know if this update has objections, I'll think for how to rework if this approach is not acceptable.

In ScrollableSelectionListGuiImpl line 179, why not to set the clipping rectangle height to the actual visible height?

#31 - 12/10/2015 12:33 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

> Let me know if this update has objections, I'll think for how to rework if this approach is not acceptable.

> In ScrollableSelectionListGuiImpl line 179, why not to set the clipping rectangle height to the actual visible height?

Because this is not common case. The ScrollableSelectionListGuiImpl is used to draw DROP-DOWN-* combo-box modes when this will not work.

#32 - 12/10/2015 02:30 PM - Eugenie Lyzenko

Task branch 2832a for review updated to revision 10965.

This is the fix for format ignoring issue for combo-box widget. Now the format is auto detected from provided item list initials.

#33 - 12/10/2015 04:08 PM - Eugenie Lyzenko

Task branch 2832a for review updated to revision 10966.

The fix for recent note. Also I have found the format ignorance is limited to GUI mode for SIMPLE and DROP-DOWN combo boxes. Respective changes was added to use this option only when it is really required. Also I did some investigation and conclude the Windows 4gl also uses line based scrolling, not pixel one.

So I consider this update as release candidate for current task if there are no objections.

#34 - 12/11/2015 07:06 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Task branch 2832a for review updated to revision 10966.
>
> The fix for recent note. Also I have found the format ignorance is limited to GUI mode for SIMPLE and DROP-DOWN combo boxes. Respective changes was added to use this option only when it is really required. Also I did some investigation and conclude the Windows 4gl also uses line based scrolling, not pixel one.
>
> So I consider this update as release candidate for current task if there are no objections.

No objections here.

#35 - 12/11/2015 08:33 AM - Greg Shah

Code Review Task Branch 2832a Revision 10966

I'm good with the changes.

Constantin: do you have any concerns with the FILL-IN related changes?

If Constantin is OK, then please put this into runtime regression testing.

#36 - 12/11/2015 12:57 PM - Greg Shah

I agree that it would be optimal to move to an AbstractContainer approach at some point but considering our time constraints I think the current approach is OK.

#37 - 12/11/2015 02:43 PM - Eugenie Lyzenko

Rebased task branch 2832a from P2J trunk revision 10958, new branch revision is 10967.

Constantin,

If you have no objections I will start runtime testing for 10967?

#38 - 12/11/2015 02:58 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Rebased task branch 2832a from P2J trunk revision 10958, new branch revision is 10967.
>
> Constantin,
>
> If you have no objections I will start runtime testing for 10967?

I'm OK with the fillin related changes, but I have a note about ComboBoxGuiImpl.setEnabled - I think you should repaint only and only if the current widget state is **not** enabled... otherwise the widget will be repainted even if it is already enabled.

#39 - 12/11/2015 03:13 PM - Greg Shah

Eugenie: please go ahead and make this fix and then go into testing.

#40 - 12/11/2015 03:38 PM - Eugenie Lyzenko

Task branch 2832a for review updated to revision 10968.

Added conditional repaint when changing widget enable state to eliminate extra paintings. Is it OK now?

#41 - 12/11/2015 04:58 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2832a for review updated to revision 10968.
>
> Added conditional repaint when changing widget enable state to eliminate extra paintings. Is it OK now?

Yes, you can start testing.

#42 - 12/11/2015 05:58 PM - Eugenie Lyzenko

The testing is in progress, main part has been started.

#43 - 12/13/2015 01:12 AM - Eugenie Lyzenko

Testing completed the results: 2832a_10968_de19d84_20151213_evl.zip. No regressions, the TC-CODES-EMPLOYEES-021 and CTRL-C 3-way tests were executed in standalone mode to confirm test passing. The server stopped, starting merge 2832a into trunk.

#44 - 12/13/2015 01:22 AM - Eugenie Lyzenko

Branch 2832a was merged to trunk as revno 10959 then it was archived.

#45 - 12/14/2015 08:40 AM - Greg Shah

- Status changed from New to Closed

- % Done changed from 0 to 100

#46 - 11/16/2016 12:12 PM - Greg Shah

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

Files

cbb_test10_3_1_p2j_gui_20151207.jpg	84.9 KB	12/07/2015	Eugenie Lyzenko
cbb_test10_3_p2j_gui_20151207.jpg	88 KB	12/07/2015	Eugenie Lyzenko
cbb_test20_1_4gl_3d_on_gui.jpg	93.1 KB	12/09/2015	Eugenie Lyzenko
cbb_test20_1_p2j_3d_on_System_font_20151209.jpg	97.9 KB	12/09/2015	Eugenie Lyzenko
cbb_test20_1_p2j_fix_20151209.jpg	97.1 KB	12/10/2015	Eugenie Lyzenko