User Interface - Feature #2546

implement combo-box widget in GUI

04/09/2015 08:03 AM - Greg Shah

Status:	Closed		**Start date:**	04/09/2015
Priority:	Normal		**Due date:**	
Assignee:	Eugenie Lyzenko		**% Done:**	0%
Category:			**Estimated time:**	0.00 hour
Target version:	GUI Support for a Complex ADM2 App			
billable:	No		**vendor_id:**	GCD

Description

Related issues:		
Related to User Interface - Feature #2534: methods/attrs support for EDITOR, ...	**Closed**	**03/06/2015**

History

#1 - 04/09/2015 08:07 AM - Greg Shah

This task is for full implementation of the GUI widget implementation for combo-box. Please note that some attribute work is in progress in #2534 (which will probably be checked in before you can finish this task).

#2 - 04/21/2015 09:58 PM - Eugenie Lyzenko

- File cbb_p2j_sample_0_gui.jpg added

The attached sample is the current state of the GUI implementations. The plan is to attach drop-down button based on predefined "btn-down-arrow" image button already implemented in ButtonGuiImpl. And drop-down list itself.

The point to clarify is about colors we use to draw widgets. What is our color policy here? Do we plan to completely simulate 4GL colors? Or we define our own color set for P2J GUI implementations. I'm asking because currently our colors differs from what we have in 4GL. For button it is not so important but for combo-box it become a point to discuss. The focused combo-box has background color the same as the title background of the active frame. If we do the same - I think it become hard to read white text on light background.

So I taken BG color for entry field - SysColor.COLOR_HIGHLIGHT and it is not the same as frame active title. Moreover I think all who working on GUI widgets implementations should keep the similar standard for color choosing to have compatible results. What do think?

#3 - 04/22/2015 10:33 AM - Greg Shah

Good headway. Would you please also attach the Windows 4GL version of this same procedure?

> The plan is to attach drop-down button based on predefined "btn-down-arrow" image button already implemented in ButtonGuiImpl. And drop-down list itself.

This makes sense.

> Do we plan to completely simulate 4GL colors?

Yes. I know that we may be missing some SYSCOLOR definitions, but otherwise I thought that our color support was complete.

I'm asking because currently our colors differs from what we have in 4GL.

Please provide some specifics.

For button it is not so important but for combo-box it become a point to discuss.

We need to match exactly and now is the time to do it.

The focused combo-box has background color the same as the title background of the active frame. If we do the same - I think it become hard to read white text on light background.

Are you describing the 4GL behavior? If so, we must match it exactly. The customer will have already worked out ways to make the runtime color configuration and the programs work together such that the users can use the system effectively. We just need to exactly honor the config and programs in the same way.

So I taken BG color for entry field - SysColor.COLOR_HIGHLIGHT and it is not the same as frame active title. Moreover I think all who working on GUI widgets implementations should keep the similar standard for color choosing to have compatible results. What do think?

We cannot make arbitrary decisions like this. It must be fully compatible with the 4GL.

#4 - 04/22/2015 12:14 PM - Eugenie Lyzenko

- File cbb_test0_4GL_0_gui.jpg added

I'm asking because currently our colors differs from what we have in 4GL.

Please provide some specifics.

Take a look at the picture attached. This is 4GL sample to compare with P2J. The frame active background and window background differ from what we have in P2J. And these BG colors are the same for combo-box selection background.

We need to match exactly and now is the time to do it.

Agreed.

The focused combo-box has background color the same as the title background of the active frame. If we do the same - I think it become hard to read white text on light background.

Are you describing the 4GL behavior?

No, it is our P2J if we make this BG color the same as used for our frame title background.

If so, we must match it exactly. The customer will have already worked out ways to make the runtime color configuration and the programs work together such that the users can use the system effectively. We just need to exactly honor the config and programs in the same way.

Yes, I think this way too. In this case we need to change BG color used to draw frame and window backgrounds.

So I taken BG color for entry field - SysColor.COLOR_HIGHLIGHT and it is not the same as frame active title. Moreover I think all who working on GUI widgets implementations should keep the similar standard for color choosing to have compatible results. What do think?

We cannot make arbitrary decisions like this. It must be fully compatible with the 4GL.

In combo-box case I just measured the color used in 4GL, then having GRB values I've found respective SysColor in our palette and used it. So the SysColor.COLOR_HIGHLIGHT equals to color used in 4GL for combo-box selection.

#5 - 04/22/2015 12:34 PM - Greg Shah

If the 4GL inherits the color from the container, then we need to do so as well.

If our result is incorrect because our frame and/or window containers are using the wrong colors, please fix it in the frame and/or window.

> In combo-box case I just measured the color used in 4GL, then having GRB values I've found respective SysColor in our palette and used it. So the SysColor.COLOR_HIGHLIGHT equals to color used in 4GL for combo-box selection.

This process is almost correct. The only issue for me is that we really want to know what Windows system color constant is being used in this case. Our SysColor implementation is intended to map into the Windows system colors. We will allow customers to override the default system colors with a set from a runtime Windows system of their choice. If we match the correct syscolor constant usage, then when customers override these, they will get the same results as in the 4GL.

#6 - 04/22/2015 08:11 PM - Eugenie Lyzenko

- File evl_upd20150422b.zip added

- File cbb_p2j_sample_1_gui.jpg added

- File cbb_test9_1_4GL_0_gui.jpg added

This is the current update for combo-box implementation in addition to screens from P2J and 4GL. The drop-down button is done. I have rejected the idea to use button widget because the GUI button implementation has many features that useless in combo-box and it looks too heavy construction. Instead the image widget is used to draw different state of the drop-down button. Currently it works in pressed/released and disabled states.

Working on drop-down list implementation.

Constantin, I see the ScrollableListGuiImpl class implemented. Is it the GUI version of the ScrollableListImpl which is used by ChUI combo-box? Or it was designed for another usage?

#7 - 04/22/2015 08:30 PM - Greg Shah

Code Review evl_upd20150422b.zip

It is a good start.

#8 - 04/23/2015 03:14 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Constantin, I see the ScrollableListGuiImpl class implemented. Is it the GUI version of the ScrollableListImpl which is used by ChUI combo-box? Or it was designed for another usage?

Currently ScrollableListGuiImpl is used to scroll through the GUI message area texts. So I think it can be used (or adjusted to work) with the GUI combo-box.

#9 - 04/27/2015 11:09 PM - Eugenie Lyzenko

- File cbb_p2j_sample_2_gui.jpg added

This is the fist step of the drop-down implementation for GUI. But the analysis is complete. Here is the painting way of the drop-down. ind means terminal type independent code, ChUI and GUI mean respective specific code, ? means not clear enough or mixing architecture.
Drop-down drawing path:

ComboBox.activate() - ind
DropDown.constructor() - ChUI
TitledWindow.show() - ChUI
OuterFrame.draw() - ChUI
TitledWindow.draw() - ind
AbstractContainer.draw() - ind?
ScrollpaneGuiImpl.draw() - GUI
Viewport.draw() - ChUI?
ScrollableSelectionListGuiImpl.draw() - GUI

#10 - 04/28/2015 09:10 AM - Greg Shah

I don't really understand what you are saying here. Are you saying that the GUI support will require the ChUI code to be executed? If so, that sounds like the ChUI code should be common code (ind).

We definitely do NOT want to add more dependencies between ChUI and GUI.

#11 - 04/28/2015 02:15 PM - Eugenie Lyzenko

> I don't really understand what you are saying here. Are you saying that the GUI support will require the ChUI code to be executed? If so, that
> sounds like the ChUI code should be common code (ind).

I'm telling the some current ChUI code(for example OuterFrame.draw()) can have character mode specifics that is not working(or even causes the exception) when running in GUI output manager mode. So to avoid issues I have to check the mode by OutputManager.instance().isChui() for example.

> We definitely do NOT want to add more dependencies between ChUI and GUI.

OK. I prefer the ChUI and GUI specifics to be completely separated(of course there can be an exceptions). Is it acceptable approach?

#12 - 04/28/2015 02:27 PM - Greg Shah

> I prefer the ChUI and GUI specifics to be completely separated(of course there can be an exceptions). Is it acceptable approach?

Yes.

> (of course there can be an exceptions)

If there are exceptions, let's discuss them so that we can see how to separate the logic.

#13 - 04/28/2015 09:41 PM - Eugenie Lyzenko

- File cbb_p2j_sample_0428_2_gui.jpg added

- File cbb_p2j_sample_0428_0_gui.jpg added

- File cbb_p2j_sample_0428_1_gui.jpg added

- File evl_upd20150428a.zip added

This is another in progress update for review to keep you informed about the implementation. I have separated GUI part of the DropDown widget. But the OuterFrame still has the code available to be run in ChUI only so I have added the console type check in this class to skip this code for GUI. Not yet merged with 10841.

Not final version, need to be refined the drawing methods too. Currently it can open drop-down by keyboard, navigate through and select another item from the list. Then the drop-down is closing value changed event is generating. The screenshots also attached.

The mouse handling is not yet working. Moreover moving mouse in application workspace closes drop-down producing unexpected behavior, need debug this case.

Constantin, what events are generated by mouse move that can ends the TC.waitForWorker() infinite time call? From the changes you recently made I mean, do you have any comments?

#14 - 04/29/2015 01:40 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> The mouse handling is not yet working. Moreover moving mouse in application workspace closes drop-down producing unexpected behavior, need debug this case.

> Constantin, what events are generated by mouse move that can ends the TC.waitForWorker() infinite time call? From the changes you recently made I mean, do you have any comments?

For combo-box's drop-down, after it is opened, the event processing should go into TC.checkForSystemEvent and ComboBox.activate, so the triggers will be hidden while the drop-down is active. Mouse movement should not terminate the "system" wait-for loop; P2J either generates a GO event (incorrectly) or something else is happening (maybe an exception thrown?).

#15 - 05/01/2015 08:50 PM - Eugenie Lyzenko

- File evl_upd20150501b.zip added

This is the next step for review in combo-box GUI widget implementation. So what it can:
1. Complete keyboard support, can open drop-down, scroll and select by ENTER key.
2. Basic mouse support, can open drop-down and select item. Then drop-down is closing automatically.

What is TODO:
1. Scroll bar does neither drawing correctly nor takes scroll event from scrollbar buttons
2. Code clean and refine.
3. There are some issues I've found with events generated by mouse when drop-down is opening via keyboard.
4. Rescan to detect and separate GUI specific code from ChUI
5. Test and fix possible issues with coordinates/dimensions settings.

#16 - 05/06/2015 11:51 PM - Eugenie Lyzenko

- File evl_upd20150506a.zip added

This is the next step for review. Finally the scrollbar is working to use sroll-buttons to navigate drop-down.

The issue I had to solve is the drop-down is a kind of frame, not a just widget, it is adding to workspace in addition to other frames(in one of them we have the main combo-box body). The main drop-down worker is the ScrollPane inside it, we have to change the physical location of the scroll-pane but drop-down itself should stay as is(to be properly used inside AbstractContainer.findMouseSource() chain). However as physical bounds for drop-down we have to use child's ScrollPane physical location to properly capture mouse events.

Continue working to fix the issue for drop-down disappearing when it opens by keyboard and then mouse move over main workspace.

#17 - 05/07/2015 08:36 AM - Greg Shah

Code Review evl_upd20150506a.zip

Generally, this looks good.

Shouldn't the WidgetFactoryAdapter.createDropDown() just throw an exception? Under what conditions will it actually be called?

Can you have this cleaned up and ready for testing tonight?

#18 - 05/07/2015 11:56 AM - Eugenie Lyzenko

Shouldn't the WidgetFactoryAdapter.createDropDown() just throw an exception? Under what conditions will it actually be called?

You are right, this should be just exception generator.

Can you have this cleaned up and ready for testing tonight?

I have this plan too. Because it works for given combo-box state. I'll make cleaning and I'm going to fix drop-down disappearing/overlapping issue too.

Constantin, if you have some short answer please share. Does every mouse movement cause the window under mouse to be activated, moving to the front in Z-order and repaint(I saw activation call)? Do we really need this reaction on every mouse move?

#19 - 05/07/2015 12:02 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

Constantin, if you have some short answer please share. Does every mouse movement cause the window under mouse to be activated, moving to the front in Z-order and repaint(I saw activation call)? Do we really need this reaction on every mouse move?

I don't think z-order matters when taking about mouse movement. From a quick test, if you have two windows, the mouse changes to i.e. border arrow or even to indicate an editable field, regardless if the window is on top of z-order or not.

I think repainting might be too aggressive: unless there are widgets which can react on mouse hover (i.e. change color or something else), then we can avoid the repaint. But the event processing for mouse movement should not be removed.

#20 - 05/07/2015 05:56 PM - Eugenie Lyzenko

- File evl_upd20150507a.zip added

This is the cleaned code for review. The UI drop-down disappearing issue is still here. Working on it.

#21 - 05/07/2015 06:45 PM - Eugenie Lyzenko

- File evl_upd20150507b.zip added

Finally I have found the fix for drop-down disappearing. So the new update for review is attached. Certainly this is not complete combo-box implementation but I think it can be used for demo purpose.

#22 - 05/08/2015 12:17 AM - Eugenie Lyzenko

- File evl_upd20150507c.zip added

This is small protection improvement to the drop-down termination facility.

#23 - 05/08/2015 08:06 AM - Greg Shah

Code Review evl_upd20150507c.zip

The changes look good.

Please merge to rev 10849. Post the merged update and get this into regression testing. It should be pretty safe, but please do a manual test on MAJIC combo-box usage.

#24 - 05/08/2015 08:51 AM - Eugenie Lyzenko

- File evl_upd20150508a.zip added

Please merge to rev 10849.

Merged update is attached.

Post the merged update and get this into regression testing. It should be pretty safe, but please do a manual test on MAJIC combo-box usage.

OK. Starting the regression cycle test.

BTW. The font metrics approach was changed in the recent updates, so the default font settings produce distortions. Do I need some special settings to get the clear picture of the P2J application?

#25 - 05/08/2015 08:55 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Please merge to rev 10849.

Merged update is attached.

Post the merged update and get this into regression testing. It should be pretty safe, but please do a manual test on MAJIC combo-box usage.

OK. Starting the regression cycle test.

BTW. The font metrics approach was changed in the recent updates, so the default font settings produce distortions. Do I need some special settings to get the clear picture of the P2J application?

Can you post an example/screen shot of the distortions you mention? Do you rely on gd.getFontHeight/width for determining i.e. list item height? If so, please switch to FontManager's getFontHeight/Width APIs.

#26 - 05/08/2015 09:08 AM - Eugenie Lyzenko

- File cbb_p2j_sample_0508_0_distorted_gui.jpg added

Can you post an example/screen shot of the distortions you mention?

See the picture(compare to ..samlpe_04028_2.jpg posted here before).

Do you rely on gd.getFontHeight/width for determining i.e. list item height? If so, please switch to FontManager's getFontHeight/Width APIs.

The problem(in GUI button too as you see) is in widget width() to be returned in character units. This must not be font dependent, right. In my update the combo-box uses width from superclass, so I do not rely on font width.

#27 - 05/08/2015 09:16 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Can you post an example/screen shot of the distortions you mention?

> See the picture(compare to ..samlpe_04028_2.jpg posted here before).

> > Do you rely on gd.getFontHeight/width for determining i.e. list item height? If so, please switch to FontManager's getFontHeight/Width APIs.

> The problem(in GUI button too as you see) is in widget width() to be returned in character units. This must not be font dependent, right. In my update the combo-box uses width from superclass, so I do not rely on font width.

Well, you are relying on labelText.length() + 2 to determine the width in character units... in GUI, this is no longer valid. You need to determine the native width of this text then convert it into character units. Use FontManager.getTextWidth for this.

This can be better seen using this test:

```
DEF BUTTON b1 LABEL "a".
DEF BUTTON b2 LABEL "i".

UPDATE b1 b2 WITH FRAME f1.

message b1:width-chars b2:width-chars.
```

If your approach using the label's character count was correct, this would have shown buttons with the same width (unless you are using a fixed-width font). But this is not the case, the buttons have different widths, when using a non-fixed-width font.

#28 - 05/08/2015 09:26 AM - Eugenie Lyzenko

> Well, you are relying on labelText.length() + 2 to determine the width in character units... in GUI, this is no longer valid. You need to determine the native width of this text then convert it into character units. Use FontManager.getTextWidth for this.

So. you mean to get the widget width in character unit I need:
1. Calculate suggested new width in character unit.
2. Convert this value into pixel units.
3. Convert this pixels value back into character units.
4. Returns this double transformed value(which will certainly not the real character width value requested in #1 above).

This is the new approach? All widgets are affected by this? The value returned this way is not corresponding to the text used as base for character units calculation. Is it OK?

#29 - 05/08/2015 09:38 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Well, you are relying on labelText.length() + 2 to determine the width in character units... in GUI, this is no longer valid. You need to determine the native width of this text then convert it into character units. Use FontManager.getTextWidth for this.

> So. you mean to get the widget width in character unit I need:
> 1. Calculate suggested new width in character unit.
> 2. Convert this value into pixel units.
> 3. Convert this pixels value back into character units.
> 4. Returns this double transformed value(which will certainly not the real character width value requested in #1 above).

> This is the new approach? All widgets are affected by this? The value returned this way is not corresponding to the text used as base for character units calculation. Is it OK?

No, that's not correct. What you need to:
1. determine the label's width in native units. In GUI, these are pixels - so you need to determine the label's width when drawn with the font associated with the button. Same for height - the label's height when drawn using the widget's font is needed. Use the FontManager.getFont/TextWidth/Height APIs.
2. having the label's width in pixels, you need to determine the button's real width. You might need to add some constant values based on the button's top/bottom/left/right border.
3. you convert this value into character units. This will be the widget's width. Same for height.

Almost all widgets which can have an "implicit" size (i.e. are not forced to have a size-phrase) need to rely on the native metrics. And yes, this value may or may not be the same as the widget's width/height in ChUI.

Use the widget's HEIGHT/WIDTH-CHARS/PIXELS attributes to determine the correctness of the computed the metrics. Is important to get this right, otherwise there will be differences in the way the frame layouts the widgets (i.e. some widgets are supposed to fit in a single row, but they no longer do, as some widget is wider than is supposed to).

#30 - 05/08/2015 11:14 AM - Eugenie Lyzenko

The merge with 10850 plus small fixes for opened drop-down. Will be used for regression testing.

#31 - 05/08/2015 11:52 AM - Eugenie Lyzenko

Another questions:
1. What the widget is required to do to properly position/sizing with layout manager in GUI?
2. Is it enough to just implement width()/height() methods to do this?
3. Will the physical dimension(we use to draw the widget) be modified accordingly?
4. Does the widget label width change the widget width in case of non-sidelabeled?

I'm asking because changing the width returned by ComboBoxGuiIml has no affects on GUI representation/next widget position.

#32 - 05/08/2015 12:27 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Another questions:
> 1. What the widget is required to do to properly position/sizing with layout manager in GUI?

If the width/height return the correct values (in character units), then the layout will be OK.

> 2. Is it enough to just implement width()/height() methods to do this?

Yes.

> 3. Will the physical dimension(we use to draw the widget) be modified accordingly?

Yes, it should, but check the physicalDimension APIs (up the hierarchy) to ensure they rely on the dimension and do not do some explicit work...

> 4. Does the widget label width change the widget width in case of non-sidelabeled?

The side/top-label exists on its own, as a separate widget. So label width, unless is part of the widget, has no effect on it, regardless if its top-label or side-label.

> I'm asking because changing the width returned by ComboBoxGuiIml has no affects on GUI representation/next widget position.

The COMBO-BOX doesn't have a mandatory size phrase, and it relies internally on the widget's format to compute its width... change the widget's format to something large, i.e. "x(20)", you will see its width being re-adjusted. For FILL-IN, the formula was pretty tricky to determine, and it relied on the font's metrics... see FillinGuiImpl.nativeWidth for the full formula. If you are lucky, it can be as easy as multiplying the widget's format length with the avg font width; if not, it will be a head-ache to discover it.

For now, please work on the general combo-box behavior. We can come back and fix the width/height later.

I'll try to put what I've done for FILL-IN into a tutorial so it can be applied to other widgets.

#33 - 05/08/2015 12:30 PM - Greg Shah

Eugenie Lyzenko wrote:

> The merge with 10850 plus small fixes for opened drop-down. Will be used for regression testing.

The file is missing.

#34 - 05/08/2015 12:36 PM - Eugenie Lyzenko

- File evl_upd20150508b.zip added

The missing file.

#35 - 05/08/2015 04:08 PM - Eugenie Lyzenko

The question. The update I'm testing has no crossing with ecf_upd20150508a.zip and compatible on the source level with 10851. Do I need to restart regression testing?

#36 - 05/08/2015 04:12 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> The question. The update I'm testing has no crossing with ecf_upd20150508a.zip and compatible on the source level with 10851. Do I need to restart regression testing?

It depends on which phase you are with your testing. If P2J will be updated and built with rev 10851 at some point, you will need to reconvert, too.

Otherwise, if you've already started the runtime testing with 10850 and you are passed the P2J build phase, let it finish.

#37 - 05/08/2015 04:16 PM - Greg Shah

If you pass testing on 10850, you can check it in without re-testing.

#38 - 05/08/2015 04:18 PM - Eugenie Lyzenko

I have finished one main cycle of the runtime testing(and have the good chances to be passed). But need another one to confirm everything is OK. Then CTRL-C tests.

#39 - 05/08/2015 04:26 PM - Eugenie Lyzenko

Greg Shah wrote:

> If you pass testing on 10850, you can check it in without re-testing.

OK. In this case I'm resuming the tests and if everything is fine - commit. BTW one majic test with combo-box is also OK(GSO 146).

#40 - 05/08/2015 04:28 PM - Greg Shah

Sounds good. Please note that you will need to merge with Vadim's GuiWidgetFactory changes which are about to be checked in. You don't need to re-test after that merge. Just post the update here.

#41 - 05/08/2015 05:00 PM - Eugenie Lyzenko

- File evl_upd20150508c.zip added

Update merged with the 10852.

#42 - 05/08/2015 06:08 PM - Eugenie Lyzenko

- File evl_upd20150508d.zip added

Several cosmetic changes to have the better look for combo-box. Only GUI code changed, so no additional testing is required.

#43 - 05/08/2015 06:11 PM - Greg Shah

Code Review evl_upd20150508d.zip

Looks good.

Please coordinate with Hynek to determine which of you will check in first. I'm guessing that you will be through testing first, but I'm not sure.

#44 - 05/08/2015 06:15 PM - Eugenie Lyzenko

Greg Shah wrote:

> Please coordinate with Hynek to determine which of you will check in first. I'm guessing that you will be through testing first, but I'm not sure.

OK. I'm expecting to get the main cycle in a 1:40 hrs:min. Then will start CTRL-C tests.

#45 - 05/08/2015 08:02 PM - Eugenie Lyzenko

The main part completed, no regressions found. Waiting for CTRL-C results.

#46 - 05/08/2015 09:31 PM - Eugenie Lyzenko

The regression tests passed, results: 10850_eff7cc1_20150508_evl.zip. Had to start 3-way CTRL-C tests in separate session to get all passed.

Starting to commit and distribute the recent update version.

#47 - 05/08/2015 09:48 PM - Eugenie Lyzenko

The evl_upd20150508d.zip has been committed as 10853. Will be distributed shortly.

#48 - 05/12/2015 09:14 PM - Eugenie Lyzenko

- File cbb_p2j_sample_0512_0_gui.jpg added

- File evl_upd20150512a.zip added

This update for review has several small fixes for UI representation for the drop-down and scroll-bar inside.
1. The drop-down positioning approach changes to almost get rid of hardcoded coordinates. Now the Y shift has reasonable explanation and logic. Additional characters are required if there is a frame title in combo-box owner and top-label for combo-box.
2. The scrollbar position and size are now dependent on condition: if the parent container(ScrollPane) is boxed or not. This ensures the correct scroll-bar drawing.
3. The scroll-bar thumb size and position need one more parameter to be correctly calculated - visible range. Appropriate method to set/get this option was added to the ScrollBar common implementation. In ChUI this is not meaningful, the thumb has predefined one char size but in GUI it is important.

The screen-shot for changes is also attached here.

#49 - 05/14/2015 04:23 AM - Hynek Cihlar

I noticed drop downs are drawn using the same renderer as the parent Window. This means they can never be drawn outside of the window (for example when the drop down doesn't fit in the parent window). This is in contrary how 4GL/Windows drop downs work. There they are really a sepparate window with a special window class.

#50 - 05/14/2015 03:58 PM - Greg Shah

Code Review evl_upd20150512a.zip

I'm OK with the changes so far. ScrollableSelectionListGuiImpl, DropDownGuiImpl need history entries.

In regard to Hynek's point: please finish your current work with ComboBox. We will move it to a separate window in the next work.

Hynek: it seems that your current work for alter-box/dialog will be helpful when we move to a separate window, is that right? If so, then we should wait for that to be checked in.

#51 - 05/14/2015 04:47 PM - Eugenie Lyzenko

- File evl_upd20150514a.zip added

This update accumulates recent changes and fixes for combo-box. What's new:
1. The drop-down can now be left either by TAB key or pressing mouse outside the list.
2. The scrollbar now visible only if it is required(the list is more than visible rows)
3. The visible rows calculation is now reflects the 4GL approach. The INNER-LINES attribute is not changing but the visible rows adjusted to the current list size. If INNER-LINES > list size, the visible rows become equal to list size. Moreover in GUI it can not be less than 3 lines, in ChUI there is no such adjustment.
4. Several minor painting fixes to make GUI implementation better match to 4GL.

If it is required to move to another task - this update is a good stage to be tested and checked in.

#52 - 05/14/2015 05:29 PM - Greg Shah

Code Review evl_upd20150514a.zip

I like the code.

1. The ComboBox.initEventsSet() and ComboBox.addMouseExitEvent() should not be described as ChUI specific. These are the common events (shared by both ChUI and GUI). Please change the javadoc and comments to match, otherwise readers might think that code is ChUI-specific.

2. The ComboBoxGuiImpl.initEventsSet() and ComboBoxGuiImpl.addMouseExitEvent() should not have comments saying // call ChUI version first. It should say // call shared version first. Also, the ComboBoxGuiImpl.initEventsSet() has a javadoc reference to ChUI that should be GUI.

3. Since DropDown.fixVisibleRows() is ChUI specific, it should be in DropDownImpl. In the common code, please make the method definition abstract.

#53 - 05/14/2015 06:05 PM - Eugenie Lyzenko

Greg Shah wrote:

> 3. Since DropDown.fixVisibleRows() is ChUI specific, it should be in DropDownImpl. In the common code, please make the method definition abstract.

You mean we need to create new class DropDownImpl to concentrate all ChUI specifics?

#54 - 05/14/2015 06:11 PM - Greg Shah

> You mean we need to create new class DropDownImpl to concentrate all ChUI specifics?

Yes.

#55 - 05/14/2015 07:11 PM - Eugenie Lyzenko

- File evl_upd20150514b.zip added

The update has fixes for previous notes and creating new class DropDownImpl to handle ChUI specifics. The ChuiWidgetFactory has also changed to reflect this.

Also I've found adding left mouse anywhere event to exit list produces regression with scrollbar(pressing on scrollbar button dismisses the drop-down). So for now I have limited mouse exit events to MOUSE-MENU-DOWN and MOUSE-EXTEND-DOWN. Additional investigation is required

to find out better approach(say pressing mouse outside drop-down generates focus change which in turn generates LEAVE event that can be included as exit event instead of direct mouse click).

#56 - 05/15/2015 08:26 AM - Greg Shah

Code Review evl_upd20150514b.zip

I'm good with this version. Please just remove the ChUI reference in the DropDown.fixVisibleRows() javadoc.

Next steps: finish all todos except for making this work in a separate modal window. Please make a list of known issues, TODOs and of any additional testing that needs to occur.

#57 - 05/15/2015 09:02 PM - Eugenie Lyzenko

- File evl_upd20150515a.zip added

This is just today changes for review. Scrolling approach changed to reflect 4GL behavior. Also I have found in GUI the drop-down activation is happening on pressing drop-down button. UP/DOWN keys from keyboard acts in GUI as next/previous item selectors. This is now implemented with the changes.

Now about TODO list(this can be extended further):

Message processing part
1. Mouse click outside drop-down should dismiss it without any changes.
2. ESC key when drop-down is active must close drop-down not terminating the application itself.

GUI rendering part
3. Complete scrolling implementation in opened drop-down. The fist visible row should be the currently selected item. The paintings for keyboard and mouse scrolling are a bit different. Test and implement this.
4. Find out why there is a "phantom" drawing when drop-down is opening an fix.
5. Attach and test tooltip handler.
6. Investigate and fix calcLimit() approach to get rid of extra spacing.
7. Verify the drop-down closing procedure makes cleanup properly.

#58 - 05/18/2015 03:56 PM - Eugenie Lyzenko

- File evl_upd20150518a.zip added

The next update for review has fixes for both message processing TODO's. Also now the top row in opened drop-down is the currently selected items. Some details about message processing issues:

1. The main logic itself is OK. The root cause is in ChUI the key for dismiss drop-down is F4. This key generates only key press event(like direction keys). But in GUI the key to END-ERROR is ESC. And this key produces both key press and key type event. So we have two sequential END-ERROR event per one ESC key pressing. One event correctly dismisses the drop-down and the next event terminates application itself(which is not happening in 4GL). So we need to eat key press event in WinKeyboardReader for GUI mode to have the correct behavior.

2. The second message processing issue idea is to send leave event when mouse is clicking to widget other than drop-down. The fix will work in ChUI mode only.

The update is compatible with recent 10866 code base.

#59 - 05/18/2015 06:54 PM - Eugenie Lyzenko

- File evl_upd20150518b.zip added

The update has code merged with 10867.

#60 - 05/19/2015 03:07 PM - Eugenie Lyzenko

- File evl_upd20150519a.zip added

The update fixes the "phantom" drop-down drawing at incorrect position by delaying the first available draw to the moment just before the event processing loop for drop-down is starting. This allows to avoid some hardcoding in drop-down position and completely simulate 4GL drop-down positioning.

#61 - 05/19/2015 06:34 PM - Eugenie Lyzenko

Addition to GUI combo-box TODO:

8. Investigate and implement different drop-down list modes, particularly if it is possible to have multiple items selected in the list. Check possible keyboard/mouse handling changes for this case.

#62 - 05/26/2015 10:26 AM - Eugenie Lyzenko

Multiple selection investigation results. In ChUI only DROP-DOWN-LIST mode can be active(even if other specified). In this mode only single selection is possible. So multiple select does not work neither in GUI nor in ChUI.

#63 - 05/26/2015 08:40 PM - Eugenie Lyzenko

- File evl_upd20150526a.zip added

The update for review merged with recent code base. In addition it adds tooltip feature to the combo-box and implements key/mouse scrolling for GUI to be exactly the same as in 4GL. In GUI mode we have different behavior for cases when scroll is preformed by keyboard and using scroolbar by mouse. To separate these cases the new member for ScrollEvent has been introduced. It become TRUE when GUI version of the ScrollBar receives the mouse click event. The common version that used in ChUI leaves this member as FALSE meaning we scroll using keyboard.

Continue working on combo-box.

#64 - 05/27/2015 06:13 PM - Eugenie Lyzenko

- File evl_upd20150527a.zip added

The update for review has two new changes:
1. Implementation for first char search/select in GUI combo-box when drop-down is not active.
2. Investigated and implemented proper clean up for GUI mode when drop-down dismissing. We need to remove widgets added into registry when drop-down is creating.

The remaining TODO points is(for the current TODO list): calcLimit() fix(related to character width in GUI I guess) and implementation SIMPLE and DROP-DOWN combo-box modes.

#65 - 05/28/2015 04:50 PM - Eugenie Lyzenko

Constantin,

To implement different combo-box modes with editable entry field I'm planning to integrate fill-in GUI implementation. To avoid any crossing for possible changes in FillInGuiImpl I need to know what are your plans about FillInGuiImpl. Are you currently working on fill-in GUI implementation? Do you have any changes to be released soon? What features in FillInGuiImpl are critical and can not be changed in any cases? Another words what are the restrictions I should know adjusting fill-in to combo-box needs?

#66 - 05/29/2015 01:40 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Are you currently working on fill-in GUI implementation?

No, I'm not working on GUI FILL-IN now.

> Do you have any changes to be released soon?

No

> What features in FillInGuiImpl are critical and can not be changed in any cases? Another words what are the restrictions I should know adjusting fill-in to combo-box needs?

You need to be careful not to break the fill-in's APIs which compute the size, when a size is not specified explicitly; otherwise, I think using explicit location and size might help you reduce changes into FILL-IN code.

#67 - 05/29/2015 06:48 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

> What features in FillInGuiImpl are critical and can not be changed in any cases? Another words what are the restrictions I should know adjusting fill-in to combo-box needs?

> You need to be careful not to break the fill-in's APIs which compute the size, when a size is not specified explicitly; otherwise, I think using explicit location and size might help you reduce changes into FILL-IN code.

OK. Thanks.

#68 - 05/29/2015 09:12 PM - Eugenie Lyzenko

- File evl_upd20150529a.zip added

The update for review presents the current state of the combo-box. The server side updated to introduce new option - mode which has single value of DROP-DOWN-LIST for ChUI mode and two more modes in GUI mode. The mode value is passing via combo-box config to the client side and in GUI mode the ComboBoxGuiImpl reconstructs additional FillInGuiImpl widget with negative ID number, adding to registry, make it visible and enable. So we have FillInGuiImpl inside ComboBoxGuiImpl.

The next step is to seamlessly integrate the FillInGuiImpl into combo-box widget.

#69 - 06/01/2015 08:18 PM - Eugenie Lyzenko

- File cbb_p2j_sample_0601_1_gui.jpg added

- File evl_upd20150601a.zip added

- File cbb_p2j_sample_0601_0_gui.jpg added

The next update for review as current state reflection. This is DROP-DOWN mode first step with FillInGuiImpl integrated into combo-box. What it does:
1. The fill-in initialization code has been moved to combo-box painting code. The reason - for the fill-in to work we need it to be added to some frame. On the combo-box creation code it is not possible because at this time we have no frame on the client side. So we have to postpone fill-in init to the stage where the combo-box already attached to the frame. I have chosen the first painting of the combo-box.

2. FillInGuiImpl has been modified to have mode flag indicating it is used inside combo-box or not. Also according to this mode FillInGuiImpl has some UI adjustments to work inside combo-box as single widget.

The current state of the ComboBoxGuiImpl can be seen on the pictures attached. The code has some commented out code to be cleaned further.

Now the next stage is to test/implement entry field editing and message processing with enclosing combo-box including passing VALUE-CHANGED event to the combo-box. The final UI tuning will be done after all internal processing is completed.

#70 - 06/02/2015 08:31 PM - Eugenie Lyzenko

- File evl_upd20150602a.zip added

Today update for review has basic implementation of the DROP-DOWN mode. The entry field is editable and until another selection is made via drop-down list the current combo-box value is the modified value from fill-in. The VALUE_CHANGED event is rerouted from fill-in to combo-box if the fill-in is the part of the combo-box.

Starting implementation for SIMPLE mode.

#71 - 06/03/2015 07:12 AM - Constantin Asofiei

Eugenie, this are some notes about evl_upd20150602a.zip:

- DropDownGuiImpl.setHeightAndPos: you are assuming that the frame's title height is 1 character unit. This is not correct - see the FrameGuiImpl.captionHeight API which gives you the space occupied by the frame title. Also, you are assuming that the combo's top-label is one char height: this is not always correct, it may vary depending on the font being used, if there are stacked labels and maybe more. To me, the drop-down in GUI looks like is dependent on the size and coordinates of the associated combo-box widget. So, why are you relying on frame decorations to place the drop-down? Is there a problem using the combo's absolute coordinates, if needed?
- the ThinClient change related to focus lost: we already have something done related to this for the popup, via GuiDriver.listenAnyMouseEvent. This registers a listener on the driver for any mouse event and notifies the widget via its MouseListener APIs. You can check in the widget's mousePressed if the click is outside of the drop-down and if so, send leave/dismiss it.
- how did you determine the ScrollBar.visibleRange default value?
- please rename ScrollEvent.mouseOrigin to isMouseOrigin
- ScrollBarGuiImpl: you are depending on the parent's boxed state when determining the scroll bar's location and dimension. I see that this is attached to a ScrollPaneGuiImpl instance, so it is may be cleaner to translate the graphics origin before the scroll bar is drawn (ScrollPaneGuiImpl.drawScrollBars should be called in a explicit gd.draw call, so that the origin is translated if a box is used or not). The idea is: at the time the draw() is called for a widget, the graphics origin is translated properly so that the widget can draw itself without relying on some parent state.
- ComboBoxGuiImpl and FillInGuiImpl: can you think of a way to sub-class the FillInGuiImpl so that the combo-box related code resides in the sub-class?
- some new methods are out of place - please double check the order. Also, please double-check the spelling of the comments/history entries in this code.

Otherwise, please let me know when you get close to finish the combo-box processing via mouse/keys/etc. We still need to determine how the combo's dimension is computed (using the explicit/implicit font, fixed or not).

#72 - 06/03/2015 08:27 AM - Eugenie Lyzenko

how did you determine the ScrollBar.visibleRange default value?

The same value as DefaultList.visibleRows.

#73 - 06/03/2015 10:26 AM - Eugenie Lyzenko

...To me, the drop-down in GUI looks like is dependent on the size and coordinates of the associated combo-box widget. So, why are you relying on frame decorations to place the drop-down? Is there a problem using the combo's absolute coordinates, if needed?

Yes, there is a problem with your approach. The combo-box and drop-down both belong to different frames. The parent frame for drop-down is border-less and header-less invisible frame. So when I get the physical location of the combo-box I have location related to frame the combo-box belongs to. If I use this coords to place drop-down I have XY shift according to combo-box's frame decoration. That's why I need to know this info.

#74 - 06/03/2015 11:56 AM - Greg Shah

That's why I need to know this info.

Please put comments in the code to explain this.

#75 - 06/03/2015 01:17 PM - Eugenie Lyzenko

the ThinClient change related to focus lost: we already have something done related to this for the popup, via GuiDriver.listenAnyMouseEvent. This registers a listener on the driver for any mouse event and notifies the widget via its MouseListener APIs. You can check in the widget's mousePressed if the click is outside of the drop-down and if so, send leave/dismiss it.

Not exactly. MousePopupable just hides popup if it is visible on any mouse press/click. I need to accept clicks when it is inside drop-down and let press to generate leave for drop-down and to be handled target widget if it is happening outside drop-down. I do not need to receive every mouse click/move into drop-down if it is happening everywhere. Moreover because drop-down is widget combination and not all widgets inside can have config/ID it is not clear how and when to properly register listener for DropDownGuiImpl.

What is wrong with the current drop-down outside click reaction approach?

#76 - 06/03/2015 01:54 PM - Eugenie Lyzenko

ComboBoxGuiImpl and FillInGuiImpl: can you think of a way to sub-class the FillInGuiImpl so that the combo-box related code resides in the sub-class?

I do not think it is convenient. I think there will be not too much combo-box specific to create new class and new config class(to be able to directly reconstruct new class by WidgetRegistry). According to 4GL docs, this is exactly the fill-in but working a bit differently.

But if you have the arguments for this separation please share.

#77 - 06/05/2015 09:14 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> ...To me, the drop-down in GUI looks like is dependent on the size and coordinates of the associated combo-box widget. So, why are you relying on frame decorations to place the drop-down? Is there a problem using the combo's absolute coordinates, if needed?

> The combo-box and drop-down both belong to different frames. The parent frame for drop-down is border-less and header-less invisible frame.

I'm OK with this.

> So when I get the physical location of the combo-box I have location related to frame the combo-box belongs to. If I use this coords to place drop-down I have XY shift according to combo-box's frame decoration. That's why I need to know this info.

Please use the -Dwidgetbrowser=yes when starting the GUI client to bring up the widget browser and see where everything is painted. From what I see, the DropDownGuiImpl (parent frame) has incorrect coordinates. You need to set proper coordinates for it (maybe when it gets created?) and after that, all child widgets (like ScrollPaneGuiImpl) will have relative coordinates to this parent. The idea is: DropDownGuiImpl acts like a GUI frame which is explicitly placed at some coordinates.

Why I'm against your current approach: it breaks our contract in which the child widgets are not aware of parent's location; with your approach, you mix things even further, relying on the coordinates/state of another frame... So, child widgets have their coordinates relative to a translated coordinate system, in which the (0,0) origin is the parent's top-left corner.

More, using a test where the frame is located somewhere else on screen, like this:

```
define var t-site as char view-as combo-box.
def var t-vend-num as int.
def var i as int.
def var ch as char.

on "1" anywhere do:
  run do-add.
end.

wait-for close of current-window.

procedure do-add.
   do on endkey undo, leave
      on error undo, leave:
      form
        t-site
        t-vend-num
        with overlay row 8 centered side-labels 1 columns
        title "Link Vendor" frame f-add.

      t-site:list-items in frame f-add = "site1,site2,site3".

      display t-site t-vend-num with frame f-add.
      update t-site t-vend-num with frame f-add.
   end.
  hide frame f-add no-pause.
end.
```

places the drop-down all over the place (press "1" to show the frame).

Finally: the drop-down is always shown bellow the combo-box widget, correct? So you need the absolute coordinates for the combo-box (you can get them with the screenPhysicalLocation which gives you actual location on screen, in pixels), the combo's height in native units and after this you can place the DropDownGuiImpl pseudo-frame bellow the combo-box.

#78 - 06/05/2015 09:24 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> the ThinClient change related to focus lost: we already have something done related to this for the popup, via
> GuiDriver.listenAnyMouseEvent. This registers a listener on the driver for any mouse event and notifies the widget via its MouseListener
> APIs. You can check in the widget's mousePressed if the click is outside of the drop-down and if so, send leave/dismiss it.

Not exactly. MousePopupable just hides popup if it is visible on any mouse press/click. I need to accept clicks when it is inside drop-down and let press to generate leave for drop-down and to be handled target widget if it is happening outside drop-down. I do not need to receive every mouse click/move into drop-down if it is happening everywhere. Moreover because drop-down is widget combination and not all widgets inside can have config/ID it is not clear how and when to properly register listener for DropDownGuiImpl.

What is wrong with the current drop-down outside click reaction approach?

I don't like it when we put very widget-specific code outside the widget's implementation class. From this check:

```
        if (inFocus != null && !isChui() && inFocus instanceof ScrollableList &&
            evt.getEvent().getID() == MouseEvt.MOUSE_PRESSED )
```

to me it means that the code should be executed by ScrollableListGuiImpl.mousePressed.

Anyway, you can leave it as is for now.

#79 - 06/05/2015 09:28 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> ComboBoxGuiImpl and FillInGuiImpl: can you think of a way to sub-class the FillInGuiImpl so that the combo-box related code resides in
> the sub-class?

I do not think it is convenient. I think there will be not too much combo-box specific to create new class and new config class(to be able to directly reconstruct new class by WidgetRegistry). According to 4GL docs, this is exactly the fill-in but working a bit differently.

But if you have the arguments for this separation please share.

I'm worried that we might need a "browser-mode" for the GUI BROWSER widget... and it might complicate things in the code. But your argument that we need new config class is good enough to leave it as is for now.

#80 - 06/05/2015 10:43 AM - Greg Shah

ComboBoxGuiImpl and FillInGuiImpl: can you think of a way to sub-class the FillInGuiImpl so that the combo-box related code resides in the sub-class?

I do not think it is convenient. I think there will be not too much combo-box specific to create new class and new config class(to be able to directly reconstruct new class by WidgetRegistry). According to 4GL docs, this is exactly the fill-in but working a bit differently.

Eugenie: actually I do think this is the time to separate this functionality. As Constantin notes, this is likely to be needed for other widgets that "embed" a fill-in, like browse.

Even if there is only a little bit of code in the sub-class, it is a much better approach to move it so that it is clear exactly which classes have combo-box functionality and which do not. I know it is extra effort, especially because you may have to provide some new methods which can be overridden in the subclass, but I think it is worth the effort.

#81 - 06/05/2015 11:51 AM - Eugenie Lyzenko

```
if (inFocus != null && !isChui() && inFocus instanceof ScrollableList &&
evt.getEvent().getID() == MouseEvt.MOUSE_PRESSED )
```

to me it means that the code should be executed by ScrollableListGuiImpl.mousePressed.

Anyway, you can leave it as is for now.

OK. Just to be clear the code:

```
...
        if (inFocus != null && !isChui() && inFocus instanceof ScrollableList &&
```

```
          evt.getEvent().getID() == MouseEvt.MOUSE_PRESSED )
    {
        // check for possible clicks inside drop-down but not in the list itself
        // it can be the drop-down buttons for example
        Widget<?> wAncest = mouseSource.ancestor();
        // if we are going outside the drop-down - send leave to it
        if (wAncest == null || !(wAncest instanceof DropDown))
...
```

means the current focus is on ScrollableList and mouse event source != DropDown or any child. This means we click outside the DropDown/ScrollableList.

#82 - 06/05/2015 12:08 PM - Eugenie Lyzenko

Eugenie: actually I do think this is the time to separate this functionality. As Constantin notes, this is likely to be needed for other widgets that "embed" a fill-in, like browse.

Even if there is only a little bit of code in the sub-class, it is a much better approach to move it so that it is clear exactly which classes have combo-box functionality and which do not. I know it is extra effort, especially because you may have to provide some new methods which can be overridden in the subclass, but I think it is worth the effort.

OK. If the fill-in is the base for some other derived classes except combo-box I agree, mixing all special features inside fill-in is not a good approach.

#83 - 06/05/2015 12:28 PM - Eugenie Lyzenko

Please use the -Dwidgetbrowser=yes when starting the GUI client to bring up the widget browser and see where everything is painted.

What I need to modify to bring up the widget browser? Is the change for client.sh below enough:

```
...
props="... -Dwidgetbrowser=yes"
...
```

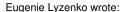**#84 - 06/05/2015 03:03 PM - Constantin Asofiei**

Eugenie Lyzenko wrote:

> Please use the -Dwidgetbrowser=yes when starting the GUI client to bring up the widget browser and see where everything is painted.

> What I need to modify to bring up the widget browser? Is the change for client.sh below enough:
> [...]

No, you need to modify the launch command. This is a JVM argument, and it goes before the main class:

```
cmd="exec $prog $srvr $dump $dtxt $lpath -Dwidgetbrowser=yes $cpath com.goldencode.p2j.main.ClientDriver $host
 $port $swing $remainder"
```

#85 - 06/09/2015 08:26 PM - Eugenie Lyzenko

- File evl_upd20150609a.zip added

The next update for review has suggested approach for SIMPLE mode implementation. The idea is a bit different that used for both drop-down modes. The selection list in simple mode is attaching to the same frame as combo-box widget itself and just looks like a drop-down but completely different entity(based on standalone ScrollPaneGuiImpl). May be we even need to create one more sub-class from ScrollPaneGuiImpl to simplify some painting utils.

The entry can be edited and be selected from the list. Both cases will generate VALUE_CHANGED event.

Continue testing and improvements. Need some feedback to be sure the way chosen is right.

#86 - 06/10/2015 08:56 PM - Eugenie Lyzenko

- File evl_upd20150610a.zip added

The next update for review has the good shape combo-box implementation with all three modes available. Key and mouse processing are available, the entry field selection properly reflects the current list selection, the selected color has been changed to be the same for entry field and selection list. Some GUI improvements to simulate 4GL look and feel for combo-box.

Continue with testing and improvement.

Constantin,

I think it is time now to clarify the character<->pixel coordinates state to eliminate hard coded adjustments. The main issue I see is to correctly get the text width/height in pixels for particular text in a way the resulting width/height can be used to build rectangle that can match the given text. The width/height should be no more and no less than required to fit the text(exact match is preferred).

Also we need to clarify color palette using. I mean we have to simulate 4GL color set, right? Our Window/Frame title does not match 4GL in Windows. This is not related to combo-box itself and I used colors best matched to 4GL in Windows to draw the combo-box but we need some agreement for the colors we can use to draw GUI widgets.

#87 - 06/10/2015 09:19 PM - Eugenie Lyzenko

- File cbb_p2j_simple_mode_0610_1_gui.jpg added

- File cbb_p2j_simple_mode_0610_0_gui.jpg added

- File cbb_p2j_simple_mode_0610_2_gui.jpg added

Some samples to illustrate SIMPLE mode of the combo-box.

#88 - 06/11/2015 02:36 PM - Greg Shah

Code Review evl_upd20150610a.zip

I think this is very good. I especially like the new EntryFieldGuiImpl.

My only question is about WidgetRegistry.addContainer(). How do those registered widgets get removed when they are no longer needed?

#89 - 06/11/2015 05:37 PM - Eugenie Lyzenko

My only question is about WidgetRegistry.addContainer(). How do those registered widgets get removed when they are no longer needed?

Now the widgets registered for permanent LIST mode(only static combo-box has been implemented). This means they will be there until application terminates(the list in this mode does exist all the application life time). It can be the issue for dynamic combo-box widget but this case is under investigation.

#90 - 06/11/2015 05:49 PM - Greg Shah

That will be a memory leak. Static widgets shouldn't last for the application lifetime. They should only live while their containing frame is not destroyed. Please implement support for removing the list widgets when the containing frame destroys the combo-box.

#91 - 06/11/2015 08:51 PM - Eugenie Lyzenko

- File evl_upd20150611a.zip added

That will be a memory leak. Static widgets shouldn't last for the application lifetime. They should only live while their containing frame is not

destroyed. Please implement support for removing the list widgets when the containing frame destroys the combo-box.

The today update has the fix for possible memory leak with SIMPLE mode list widget registration. New method added to WidgetRegistry. Also this clean up approach is now using when dismissing drop-down too.

In addition the combo-box all modes widget is modified to support combo-box disabled state. Had to modify FillInGuiImpl to properly handle disabled entry field widget text drawing. No possible to use keyboard/mouse for simple list when it is disabled.

#92 - 06/12/2015 06:05 AM - Constantin Asofiei

Eugenie, first of all, see note 77 and the testcase at this note - the drop-down's coordinates are still incorrect. We should fix this before releasing this update. Otherwise, I'm OK with the changes.

> I think it is time now to clarify the character<->pixel coordinates state to eliminate hard coded adjustments. The main issue I see is to correctly get the text width/height in pixels for particular text in a way the resulting width/height can be used to build rectangle that can match the given text. The width/height should be no more and no less than required to fit the text(exact match is preferred).

This is not the way to go. For widgets which are not static text (like LABEL, TEXT or LITERAL), their dimension is computed based on the widget's format, and not on the text's dimension. So we can't build a box around the text - the dimension you are referring is independent of the text being drawn in it.

On a side note, scaling the text to fit legacy metrics does not work - it does not produce good results when rendering, the text gets distorted and sometimes unreadable. So our approach is to use "best-of" fonts which are scaled so that their metrics (almost) match the legacy metrics of the font, and drawing with it produces a good fit (this solves the fact that java doesn't support bitmap fonts, too). Note that I've uploaded to the testcases project a modified directory.xml and .ttf files to the testcases/server/fonts/ folder, so you should be able to use true-type fonts under linux - to compare with windev01. Add this node to your directory.xml (at the same level as custom-fonts node):

```
        <node class="container" name="font-table">
          <node class="string" name="font8">
            <node-attribute name="value" value="Tahoma, size 14"/>
          </node>
          <node class="string" name="font9">
            <node-attribute name="value" value="Courier New, size 10"/>
          </node>
          <node class="string" name="font10">
            <node-attribute name="value" value="Segoe UI, size 10"/>
          </node>
        </node>
```

and change your progress.ini (get a sample from windev01:D:\temp\oe102bworking\ca\fonts\progress.ini) by adding these fonts to the [Fonts] section:

```
font8=Tahoma, size 14
font9=Courier New, size 10
font10=Segoe UI, size 10
```

This will allow you to use the same fonts on linux as on windev01, so you can get a better feel of how the widget really looks in P2J: set the combo's font via the FONT clause (i.e. FONT 8). To use the progress.ini, use this command: prowin32 -ininame progress.ini -basekey INI. To get rid of the hard-coded adjustments, I think it will be enough to determine the formula to compute the combo-box's and drop-down's dimension, based on the current format + font metrics, but there still might be some constants involved (like border/insets or dimensions of the "down-arrow" icon). To resume, the formula needs to cover these cases:

- implicit/explicit font
- fixed/variable width font
- font size from 1 to 100 (or maximum value which can be displayed in 4GL)
- 3d/no 3d
- format, covering lengths from 1 to 100 (or maximum value which can be displayed in 4GL). Side note: for fill-in format lengths 1 to 12 had had to

be treated distinctly.

- explicit size set via widget's width/height attributes - this should be simple, the "editable" part should be the full width without border/insets/down-arrow

I will work on putting together a tutorial on what I did to determine the formula for fill-in. Basically, it boils down to capturing metrics (on 4GL system) for what I mentioned above, and after this try to combine format length/font metrics/constants/pixels-per-row/pixels-per-column/3d-flag into a formula. Why we need this to be accurate: widget's dimension is used when performing frame layout, and if we get this wrong, the layout will be wrong.

Also we need to clarify color palette using. I mean we have to simulate 4GL color set, right? Our Window/Frame title does not match 4GL in Windows. This is not related to combo-box itself and I used colors best matched to 4GL in Windows to draw the combo-box but we need some agreement for the colors we can use to draw GUI widgets.

Windows might or might not have specific colors only for combo-box; so first check if its colors can be changed via the window's color scheme - if it can be changed, then determine which color names in SysColor are associated with it. Unfortunately, what we don't have currently is some documentation on which color name applies to which widget/widget part in Windows' color scheme. This can get confusing and even if a color matches (as RGB) to what you expect, its purpose might not match its usage.

Also, don't forget to test the BGCOLOR and FGCOLOR options with the combo-box and see if it behaves correct.

#93 - 06/13/2015 05:55 PM - Eugenie Lyzenko

- File cbb_p2j_drop_down_0613_0_gui.jpg added

- File evl_upd20150613a.zip added

- File cbb_p2j_drop_down_0613_1_gui.jpg added

The update for review fixes the drop-down incorrect coordinates setting. The idea is based on the fact the drop-down is adding to the Window workspace(finally the combo-box is also indirectly belongs to the same workspace). So we need to know absolute screen coordinates for both combo-box and Window workspace, subtract second from first and add combo-box height to get the correct drop-down coordinates. And now every children for drop-down has zero shift.

To properly draw the drop-down the drawing routine was overridden in DropDownGuiImpl.

The pictures illustrated the correct working are also attached.

#94 - 06/14/2015 09:10 AM - Eugenie Lyzenko

- File evl_upd20150614a.zip added

Merged with recent 10879 code base.

#95 - 06/14/2015 06:04 PM - Eugenie Lyzenko

More features findings for 4GL combo-box.

1. When combo-box is in SIMPLE or DROP-DOWN modes the entry field keeps selected region while working with list with keyboard or mouse. In DROP-DOWN-LIST mode this is not the true.

2. When combo-box is in DROP-DOWN or DROP-DOWN-LIST mode when list is active and mouse is over the item in the list - the selection is performed even if the mouse button is not clicked(like the tooltip feature). In SIMPLE mode this is not the true.

#96 - 06/15/2015 08:30 AM - Eugenie Lyzenko

- File evl_upd20150615a.zip added

Small change to fix entry field selection draw in DROP-DOWN mode when drop-down list become active.

#97 - 06/15/2015 09:38 AM - Eugenie Lyzenko

Clarification about VALUE-CHANGED event generation:

In all three modes mouse click and keyboard UP/DOWN cause the VALUE-CHANGED generation. Even if the new item is the same as previous one.

But despite the fact in DROP-DOWN and DROP-DOWN-LIST modes placing mouse over item in the opened list causes the item to be selected(highlighted) - this does NOT generate VALUE-CHANGED event.

#98 - 06/15/2015 04:37 PM - Constantin Asofiei

The metrics for the combo-box widget have already been captured into uast/combo_box/metrics/ folder (they are all for 3D mode); following is a description of the steps taken to capture these metrics.

First step is to collect metrics for implicit and explicit font cases, combined with misc format lenghts. For this, a custom progress.ini.tpl is used as a template, where fonts are set as these:

```
DefaultFont=xxx
DefaultFixedFont=xxx
...
font9=xxx
```

The run_metrics.bat program will read this file and replace all xxx entries with the font name and font size (as in Courier New, size 10). The font list is specified in font-list.txt. The maximum font size can be set in the run_metrics.bat program, the maxsize variable. Experimentation is needed to determine which max font size can be treated by 4GL (for combo-box, I used 50 for a max format of "x(50)", so that the combo fits the window).

Next, we need to build a 4GL program which displays a number of widgets, each one with an explicit format set, incrementally. This program must be named sizing.p.tpl (otherwise run_metrics.bat will not pick it up) and it looks like (when using an explicit font; if an implicit font is used, the font 9 clause is removed):

```
def var ch1 as char init '1' format 'x(1)' bgcolor 4 font 9 view-as combo-box.
def var ch2 as char init '2' format 'x(2)' bgcolor 3 font 9 view-as combo-box.
def var ch3 as char init '3' format 'x(3)' bgcolor 4 font 9 view-as combo-box.
def var ch4 as char init '4' format 'x(4)' bgcolor 3 font 9 view-as combo-box.
...
def var ch50 as char init '50' format 'x(50)' bgcolor 3 font 9 view-as combo-box.

display
      ch1
      skip
      ch2
      skip
      ch3
```

```
        skip
        ch4
        skip
        ...
        ch50
        with frame fx no-labels size 650 by 200.

def stream rpt.
output stream rpt to value("yyy").
def var h as handle.
h = frame fx:first-child:first-child.
do while h <> ?:
    put stream rpt h:width-chars    space(1)
                  h:height-chars   space(1)
                  h:width-pixels   space(1)
                  h:height-pixels  space(1)
                  session:pixels-per-row space(1)
                  session:pixels-per-column space(1)
                  skip.
    h = h:next-sibling.
end.
output stream rpt close.

quit.
```

Note how the output is made to a file named yyy - this must be left unchanged, as run_metrics.bat will process this 4GL template program and replace the yyy string with a file name like fontname-fontsize.txt, as in Courier New-10.txt. So, metrics will be captured for each font, each font size and each format length, in its own file, using the format above. This gives as a large enough pool of data to determine and validate a generic formula which describes the widget's width/height.   The second step is to interpret these captured metrics and build a generic formula which can compute the metrics, based on the following:

- misc constants (as border insets, constant widths related to 3D mode, "down-arrow button", etc)
- font avg width, max width, font height
- format length
- type of font: fixed or non-fixed width
- if an implicit font is used or not
- if a fixed width font is used with a non-fixed-width widget (i.e. character fill-ins use implicitly the DEFAULT-FONT, which is assumed to be non-fixed-width, so fixed-width "flavor" of the font is ignored, if the widget does not expect a fixed width font to be used with it). The idea is: font's "fixed width" flavor might be less important than the fact that the widget expects a fixed width font or not.
- pixels-per-column and pixels-per-row values

Each line in the file with captured metrics is for a combo with a specific format length, given by the line's number in the file; the columns in each line are combo's metrics as reported by 4GL:

- width-chars
- height-chars
- width-pixels
- height-pixels
- pixels-per-row
- pixels-per-column

The CheckMetricsCb.java program outlies a simple "template" to verify attempted formulas, until the right one is found (you are free to modify it as you want). The checkMetrics method receives as parameter a file name from the sizing.implicit.font or sizing.explicit.font folders (from which the font name and size are extracted) and tests the formula against the captured metrics. If differences are found, these can be logged, manipulated, etc. When differences are no longer found between the 4GL metrics and your calculations, you can conclude the formula is correct.

Ideas:

- start with explicit and fixed width fonts
- get sample metrics (i.e. courier new size 12) into excel and try to assemble something there first (when importing, use "fixed width" - copy-pasting the file works too)
- the ods files from uast/fonts/widget.metrics may be of help - it already contains captured metrics for combo-box, it can get you into excel faster
- determine constants by building 4GL programs which show a combo with misc font sizes/format lengths and see how the width/height gets changed (I think the "down-arrow" button has a constant width, but its height follows the widget's).
- determine how the fmt length maps into the formula (usually, it gets multiplied by the font's (max)width or its adjusted width, see bellow)
- the "x + x / 2" concept ("adjusted font width") is used for fill-in and ppr/ppc formulas - so it may be of help
- when logging differences in the java program, if they seem the same (i.e. are constant, follow an arithmetic progression, etc), then you might have discovered some secondary element of the formula
- my first attempt would be to go with FILL-IN's formula and see if it can be adjusted to match the combo's width
- this is a trial-and-error approach, combine the font's metrics however you can think
- the metrics are for 3D mode - these already imposes some constant value added to the formula. Capture the metrics for non-3d-mode and see how they differ, to get the formula (it might be enough to just play with a 4GL program and check the metrics with 3d/non-3d for some fonts).

#99 - 06/16/2015 10:36 AM - Greg Shah

Code Reviews evl_upd20150611a.zip evl_upd20150613a.zip evl_upd20150614a.zip evl_upd20150615a.zip

I am generally OK with the changes. A few minor points:

1. I would prefer to implement the ThinClient.processProgressEvent() change in a way that is not specific to the drop-down. Can we define a method for this purpose in Widget and implement an empty version in AbstractWidget? Then the code that is specific to drop-down can be moved there.

I realize that we have a lot of similar code in TC, but I'm trying to move that code out. Since this code is new with this update, the time to move it out is now.

2. Hynek's work in #2559 (as posted in #2252) adds some similar recursive add/remove logic to the widget registry, as you have in your update. His update is already in regression testing and it will be checked in first. Expect to replace your code with the code that is there, unless you find a feature that requires a different implementation.

3. Expect that there will be some changes to merge from #2534 and #2563, both of which will probably be checked in before yours.

#100 - 06/16/2015 10:48 AM - Eugenie Lyzenko

Greg Shah wrote:

> Code Reviews evl_upd20150611a.zip evl_upd20150613a.zip evl_upd20150614a.zip evl_upd20150615a.zip
>
> I am generally OK with the changes. A few minor points:
>
> 1. I would prefer to implement the ThinClient.processProgressEvent() change in a way that is not specific to the drop-down. Can we define a method for this purpose in Widget and implement an empty version in AbstractWidget? Then the code that is specific to drop-down can be moved there.
>
> I realize that we have a lot of similar code in TC, but I'm trying to move that code out. Since this code is new with this update, the time to move it out is now.

I see. I'll investigate this point.

> 2. Hynek's work in #2559 (as posted in #2252) adds some similar recursive add/remove logic to the widget registry, as you have in your update. His update is already in regression testing and it will be checked in first. Expect to replace your code with the code that is there, unless you find a feature that requires a different implementation.

> 3. Expect that there will be some changes to merge from #2534 and #2563, both of which will probably be checked in before yours.

OK. I guess Hynek will be first because I'm still in a progress of debugging VALUE-CHANGED issue I've found for opened drop-down in combo-box.

#101 - 06/16/2015 08:51 PM - Eugenie Lyzenko

- File evl_upd20150616a.zip added

The today update has small changes to event processing. The new class I have introduced in one previous update - ComboBoxSimpleListGuiImpl has been removed because in current implementation it does nothing useful, the notification about simple list change is performing via selection listener usage.

This update also has 4GL like reaction to the key up/down selection - properly changing the entry field value for all three modes. However the dynamic drop-down list(DROP-DOWN and DROP-DOWN-LIST modes) still does not properly generates VALUE-CHANGED event for main combo-box part. Here is the investigation details:

The dynamic drop-down list in these modes uses ChUI approach to utilize the TC.waitForWorker() call with predefined exit events(RETURN, LEAVE, MOUSE-PRESS outside the drop-down). And this event list replaces the list of the events(including possible defined VALUE-CHANGED trigger) for the time the drop-down is active. I would say it works in "modal" mode(only events for drop-down itself are available) until it is closed. This is OK for ChUI mode where the processing goes exactly this way, the VALUE-CHANGED event can be generated after drop-down dismiss. But in GUI mode the reaction is different. The drop-down list generates and combo-box receive and execute respective trigger for VALUE-CHANGED event and possibly all other triggers defined before drop-down activation when the drop-down list is active.

So in this situation the currently used ChUI approach is poorly adjustable to GUI mode. And I'm inclining to leave the current approach for ChUI only mode and implement completely different approach in GUI. For GUI mode I suggest to use dynamic widget added to the same frame as combo-box main part and NOT to use additional TC.waitForWorker() loop letting all events for all combo-box part to go through worker responsible for main combo-box part. Like it is currently used in SIMPLE mode. This give us ability to use triggers defined before drop-down activation but decrease drop-down "modality".

What do you think? Is it acceptable? Any notes?

#102 - 06/17/2015 09:09 AM - Greg Shah

Code Review evl_upd20150616a.zip

The code changes are fine.

#103 - 06/17/2015 09:12 AM - Greg Shah

> And I'm inclining to leave the current approach for ChUI only mode and implement completely different approach in GUI. For GUI mode I suggest to use dynamic widget added to the same frame as combo-box main part and NOT to use additional TC.waitForWorker() loop letting all events for all combo-box part to go through worker responsible for main combo-box part. Like it is currently used in SIMPLE mode. This give us ability to use triggers defined before drop-down activation but decrease drop-down "modality".
>
> What do you think? Is it acceptable? Any notes?

Questions:

1. How does this work when we move to the "independent window" approach that we know will have to be implemented once Hynek's changes are

ready?

2. How can we hide as much of this processing inside the combo-box classes and leave the TC processing to be more generic?

#104 - 06/17/2015 09:49 AM - Constantin Asofiei

Eugenie, can you extract the changes in the Scrollbar* classes in a separate update, so they can be released? I think I need them to fix some scrolling issue for my EDITOR work (the thumb doesn't get resized based on the amount of data needed to be scrolled...).

#105 - 06/17/2015 10:34 AM - Eugenie Lyzenko

- File evl_upd20150617a.zip added

Constantin Asofiei wrote:

> Eugenie, can you extract the changes in the Scrollbar* classes in a separate update, so they can be released? I think I need them to fix some scrolling issue for my EDITOR work (the thumb doesn't get resized based on the amount of data needed to be scrolled...).

Here it is.

#106 - 06/17/2015 01:42 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Constantin Asofiei wrote:
>
>> Eugenie, can you extract the changes in the Scrollbar* classes in a separate update, so they can be released? I think I need them to fix some scrolling issue for my EDITOR work (the thumb doesn't get resized based on the amount of data needed to be scrolled...).
>
> Here it is.

ScrollEvent class is needed - please add this too.

Greg: this can go first, as it helps my editor code.

#107 - 06/17/2015 01:48 PM - Greg Shah

Code Review evl_upd20150617a.zip

I'm OK with the changes. Please get them regression tested.

#108 - 06/17/2015 02:39 PM - Eugenie Lyzenko

- File evl_upd20150617b.zip added

Constantin Asofiei wrote:

> Eugenie Lyzenko wrote:
>
>> Here it is.

> ScrollEvent class is needed - please add this too.

Done.

#109 - 06/18/2015 01:42 PM - Eugenie Lyzenko

One more point to clarify.

I'm trying to make the drop-down list event processing loop more reliable an responsible to external communication. The idea is to take the current event list that exists before starting the drop-down and integrate it into event list active when drop-down is active. I guess this can allow to handle both exit events and other triggers. The point not clear for me is the ThinClient.trigger() method:

```
...
        trv = server.trigger(triggerId, eventId, sourceId, resourceId, sb); // line 11080
...
```

For some reasons the result of this call to server depends on from what event loop it is called. If we call it from main application loop(if any) the trigger is calling fine with respective result and executing registered server reaction. But if we call it from drop-down waitForWorker() the result is null, server side is not executed(as I have expected to be). The point is the triggerId, eventId, sourceId, resourceId, parameters are absolutely the same.

The question: why the calls to the server can have different results for same input options? The result is depended from calling context/nesting level? What do I miss?

#110 - 06/18/2015 08:25 PM - Eugenie Lyzenko

Finally the main part of the testing for 0617b completed without regressions. I'm expecting to finish CTRL-C testing today in a 1.5 - 2 hours. Also commit and distribute the update then.

#111 - 06/18/2015 09:52 PM - Eugenie Lyzenko

Testing completed, no regressions, had to run 3-way CTRL-C tests in separate session. The results are: 10879_bb2f7ec_20150618_evl.zip. The update will be committed and distributed shortly.

#112 - 06/18/2015 10:07 PM - Eugenie Lyzenko

Committed in bzr as 10881.

#113 - 06/18/2015 10:44 PM - Eugenie Lyzenko

- File evl_upd20150618a.zip added

The update for review has subtracted Scrollbar related code that distributed separately. Also it contains suggested approach first part fix that attempts to execute the triggers defined before drop-down activation within the drop-down lifetime. To do this the current event list is taking from TC and chaining into the drop-down exit list.

Now the next step is to find out the server part does not react to the trigger called.

And the next step is to investigate the possibility to exclude drop-down related mouse processing code from TC.

#114 - 06/19/2015 04:08 PM - Eugenie Lyzenko

- File evl_upd20150619a.zip added

The update for review has implementation for ThinClient change related to the mouse clicks outside drop-down. The new method introduced for every widget, doing nothing by default. The ScrollableSelectionListGuiImpl does the real work for drop-down to function properly.

#115 - 06/22/2015 08:44 PM - Eugenie Lyzenko

- File evl_upd20150622a.zip added

The update contains combo-box changes merged with recent code base without recently distributed Scrollbar changes.

The event-list merge implementation is still under debugging.

#116 - 06/23/2015 08:30 PM - Eugenie Lyzenko

The root cause for ignoring trigger call has been found. During obtaining the server method from proxy class the all window ID's are requesting from WindowManager. And when the drop-down is active it also considered as Window(TitledWindow subclass). Because it does not have ID/config - the NPE is happening while working with ID, this interrupts the trigger call to server until drop-down is active. Working on solution.

#117 - 06/24/2015 03:57 AM - Constantin Asofiei

Review for 0622a.zip:

- ComboBoxConfig - the added imports are unused
- DropDown - the fixVisibleRows abstract method is in the wrong place
- ScrollableLis - the setTopRow abstract method is in the wrong place
- WidgetRegistry - addContainer and removeContainer duplicates the work in addWidgetsRecursive and removeWidgetsRecursive methods

added by Hynek - please remove yours and use/adjust the existing ones

Otherwise I'm good with the logic.

#118 - 06/24/2015 04:10 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> The root cause for ignoring trigger call has been found. During obtaining the server method from proxy class the all window ID's are requesting
> from WindowManager. And when the drop-down is active it also considered as Window(TitledWindow subclass). Because it does not have
> ID/config - the NPE is happening while working with ID, this interrupts the trigger call to server until drop-down is active. Working on solution.

Yes, the DropDown doesn't have an ID yet. You can add a virtual ID to it via WidgetId.nextID, before dropDown.show is called in
ComboBox.activate:1016, but P2J might not like windows with virtual IDs registered in WindowManager$WorkArea.windows, as there are calls like
tk.enableOSEvents in TC.applyChanges:11717 which expect the received IDs to have an associated registered window at the driver-level.

#119 - 06/24/2015 08:30 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

> Yes, the DropDown doesn't have an ID yet. You can add a virtual ID to it via WidgetId.nextID, before dropDown.show is called in
> ComboBox.activate:1016, but P2J might not like windows with virtual IDs registered in WindowManager$WorkArea.windows, as there are calls
> like tk.enableOSEvents in TC.applyChanges:11717 which expect the received IDs to have an associated registered window at the driver-level.

Even if drop-down looks like a window it is in not a window, does not have an config. It will be wrong to use it as another window, this produces NPE
in a cases when drop-down differs from a window and the code refers to objects that is null in drop-down.

To avoid this I suggest to modify code working with window like this(WindowManager):

```
...
   public static int[] windowIds()
   {
      WorkArea wa = work.get();

      // get the windows id size excluding the drop-down
      List<Integer> ids = new ArrayList<Integer>();
      for (TitledWindow<?> w : wa.windows)
      {
         // excluding drop-down
         if (!(w instanceof DropDown))
         {
            ids.add(w.getId().asInt());
         }
      }

      // should have at least one main window
      return Utils.integerCollectionToPrimitive(ids);
   }
...
```

#120 - 06/24/2015 08:35 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

> Review for 0622a.zip:
>
> > - ComboBoxConfig - the added imports are unused

Please clarify. The only thing I've added is enum for different modes. And it is certainly used.

#121 - 06/24/2015 08:54 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Constantin Asofiei wrote:
>
> > Review for 0622a.zip:
> >
> > > - ComboBoxConfig - the added imports are unused
>
> Please clarify. The only thing I've added is enum for different modes. And it is certainly used.

Lines 49 and 50 of this file:

```
import com.goldencode.p2j.ui.client.format.*;
import com.goldencode.p2j.util.*;
```

#122 - 06/24/2015 09:04 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Constantin Asofiei wrote:

Yes, the DropDown doesn't have an ID yet. You can add a virtual ID to it via WidgetId.nextID, before dropDown.show is called in ComboBox.activate:1016, but P2J might not like windows with virtual IDs registered in WindowManager$WorkArea.windows, as there are calls like tk.enableOSEvents in TC.applyChanges:11717 which expect the received IDs to have an associated registered window at the driver-level.

Even if drop-down looks like a window it is in not a window, does not have an config. It will be wrong to use it as another window, this produces NPE in a cases when drop-down differs from a window and the code refers to objects that is null in drop-down.

To avoid this I suggest to modify code working with window like this(WindowManager):
[...]

The plan is to make the DropDown parented to an actual window (so that the body can go beyond the current window's border), correct? So DropDown will become a window, as far as the client-side is concerned. But for server-side, there will be no correspondent for it. Thus instead of checking instanceof DropDown let windowIds return only the IDs of top-level windows (instanceof TopLevelWindow).

#123 - 06/24/2015 09:22 AM - Greg Shah

In addition to making the processing safer for drop-down, isn't the real problem that the trigger processing should be seeing the combo-box id as the source? Correct me if I am wrong, but I have been assuming that the drop-down is not considered a real widget by the 4GL (you can't have triggers on the drop-down, only on the combo-box). If that is right, then the drop-down should never be the "source" of an event.

#124 - 06/24/2015 09:41 AM - Eugenie Lyzenko

Greg Shah wrote:

> In addition to making the processing safer for drop-down, isn't the real problem that the trigger processing should be seeing the combo-box id as the source? Correct me if I am wrong, but I have been assuming that the drop-down is not considered a real widget by the 4GL (you can't have triggers on the drop-down, only on the combo-box). If that is right, then the drop-down should never be the "source" of an event.

Yes, the drop-down is a part of the combo-box, so any changes(that can be the 4GL events) in drop-down should produce the change in combo-box and then event generation with combo-box as the source widget. Not depending if the drop-down is opened or not(in GUI mode). In ChUI mode all event processing is performed after drop-down disappears.

#125 - 06/25/2015 07:47 PM - Eugenie Lyzenko

The condition of the application to finish is the pressing RETURN key. It is very strange because it is happening even when removing the ENTER key from exit event list. Still looking for a root cause. It is happening only when drop-down become active and only in GUI mode. Something became wrong with the TC event processing.

Continue working.

#126 - 06/26/2015 08:14 PM - Eugenie Lyzenko

- File evl_upd20150626a.zip added

Finally the update merged with the recent code base is ready with all fixes required:
1. Resolution for previous update notes.
2. Fixes in ThinClient for drop-down related calls and ZOrder strange logic.
3. Event loop is now working properly with drop-down.
4. GUI combo-box event processing logic significantly reworked to match the GUI specific.
5. The server triggers are now can be called when drop-down is active, the current event list from TC is now chaining with drop-down events.
6. Removed WidgetRegistry class from update. All required new method already there from Hynek's change.

Continue working with combo-box for testing the approach and verify all events cases. However this update can be tested and distributed if required.

#127 - 06/28/2015 07:58 PM - Eugenie Lyzenko

- File evl_upd20150628a.zip added

Merge with recent code base for review. In addition the event list chaining in drop-down cycle is now filtering to contain only combo-box related events.

#128 - 06/30/2015 06:46 PM - Eugenie Lyzenko

- File evl_upd20150630a.zip added

This merged with recent code update for review adds support for current item selection by mouse hovering. The idea is to track mouse move in ScrollableSelectionListGuiImpl and when mouse become over new item - change the current one. ScrollPaneGuiImpl has also modified to route mouse move to the child ScrollableSelectionListGuiImpl class.

#129 - 07/01/2015 09:28 PM - Eugenie Lyzenko

- File cbb_p2j_sample_0701_1_gui.jpg added

- File cbb_p2j_sample_0701_0_gui.jpg added

- File evl_upd20150701a.zip added

- File cbb_p2j_sample_0701_2_gui.jpg added

The next update for review presents the scaling factor implementation to eliminate hardcoding adjustments. The idea is to compute the difference between expected text width based on single column size and the width of the average text with the given length. The sample for average text is the ' ' chars filling. It gives good results as you can see on the pictures attached for all 3 modes. For fixed fonts the scaling will be 1.0.

#130 - 07/02/2015 05:51 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> The next update for review presents the scaling factor implementation to eliminate hardcoding adjustments. The idea is to compute the difference between expected text width based on single column size and the width of the average text with the given length. The sample for average text is the ' ' chars filling. It gives good results as you can see on the pictures attached for all 3 modes. For fixed fonts the scaling will be 1.0.

Is scaling supposed to be a method to compute the widget's width? I've tried the following test:

```
define var t1 as char font 8 format "x(10)" view-as combo-box list-items "wwwwwwwww","a","b","c","d".
define var t2 as char font 8 format "x(10)" view-as combo-box list-items "iiiiiiiii","a","b","c","d".
define var t3 as char font 8 format "x(10)" view-as combo-box list-items "MMMMMMMMM","a","b","c","d".
define var t4 as char font 8 format "x(10)" view-as combo-box list-items "WWWWWWWWW","a","b","c","d".

FORM t1 skip t2 skip t3 skip t4 WITH FRAME f1.
MESSAGE t1:WIDTH-PIXELS IN FRAME f1.
MESSAGE t2:WIDTH-PIXELS IN FRAME f1.
MESSAGE t3:WIDTH-PIXELS IN FRAME f1.
MESSAGE t4:WIDTH-PIXELS IN FRAME f1.

update t1 t2 t3 t4 with frame f1 side-labels.
```

and the width is not computed OK.

font 8 is font8=Tahoma, size 14 in progress.ini - use the directory.xml font-table as in note 92. The custom-fonts should be already in the bzr's directory.xml. Also, when trying smaller font sizes, the combo-box width seems to grow instead of shrink; more, the combo's height is dependent on the widget's font, too, not only width.

Another issue with this test is some z-order problem: when the drop-down covers one or more widget, currently the covered widget captures the mouse event, instead of the current active drop-down.

If you want to start working on the widget dimension, please finish the combo behaviour which is not dependent on this (like events, z-order, and anything else you have on your list) first. After this is stabilized, working properly and released, you can go back and finish the dimension - this is a challenge on its own, and the goal is to have a formula which computes the same metrics as the captured ones (as mentioned in note 98).

#131 - 07/03/2015 06:07 AM - Eugenie Lyzenko

Another issue with this test is some z-order problem: when the drop-down covers one or more widget, currently the covered widget captures the mouse event, instead of the current active drop-down.

Confirmed, working on solution.

If you want to start working on the widget dimension, please finish the combo behaviour which is not dependent on this (like events, z-order, and anything else you have on your list) first.

Agreed, this is something that can be postponed with the current state.

After this is stabilized, working properly and released, you can go back and finish the dimension - this is a challenge on its own, and the goal is to have a formula which computes the same metrics as the captured ones (as mentioned in note 98).

OK. One note here. I do not think the formula is such complicated as you described. I see in 4GL on Windows the picture is not ideal too(the drop-down visible lines not always match the requested lines, the combo-box width computing in way to see the full format(or current selection) characters for the entry-field combo-box part). This is also a kind of "best possible" solution because the fonts are not fixed width.

So I think the best way is to know how this scaling is performed under Windows itself because 4GL uses Windows native combo-box control to implement combo-box.

The other task from TODO list is to implement/verify dynamic combo-box and to support all possible attributes/methods for combo-box widget.

#132 - 07/03/2015 05:16 PM - Eugenie Lyzenko

Created task branch 2546a from P2J trunk revision 10893.

#133 - 07/03/2015 07:29 PM - Eugenie Lyzenko

Please review task branch 2546a, revision 10894.

This is my first usage of the branch upload process. All current work state has been committed into this branch. In addition to the recent update uploaded into Redmine this revision for review has the fix for Constantin's note about incorrect routing of the mouse events for widgets that are covered bu opened drop-down. The reason was the drop-down too early closing(by mouse press). In this model the following mouse release and mouse click event were handled by TC when there is no drop-down visible and active. Changing drop-down dismiss to the mouse click solves this issue.

On the other hand I see another issue with this testcase. The opened drop-down is not dismissing when pressing the other drop-down to open(if there are several combo-boxes in a frame). Working on solution.

PS: The branch code has some debug points, to be removed in final version. For now I need this to simplify mouse events tracking.

#134 - 07/06/2015 09:48 AM - Greg Shah

> The root cause was too early dismiss the drop-down. I have preferred to use mouse press due to mouse press does immediate reaction, while mouse click has delay(until mouse release). But this has a side effect in having mouse events for widget that does not exist anymore. Using mouse click fixes this issue.

Does the 4GL code (actually, it is really the Windows drop-down control) dismiss the drop-down on press or click? We cannot be incompatible with this.

Constantin will review your code.

#135 - 07/06/2015 04:30 PM - Eugenie Lyzenko

Greg Shah wrote:

> Does the 4GL code (actually, it is really the Windows drop-down control) dismiss the drop-down on press or click? We cannot be incompatible with this.
>
> Constantin will review your code.

The drop-down is opening on mouse press but closing on mouse click(release to be exact).

BTW, since weekend the Windows 4GL system is waiting to be restarted. Should I try to reboot it(not sure if I have enough permissions)? Or this will be done as planned restarting by the customer admins?

#136 - 07/06/2015 04:42 PM - Eugenie Lyzenko

The exception from drop-down dismiss rule is mouse events in combo-box entry field or outside drop-down or combo-box(when drop-down is opened). In this case drop-down is closing on mouse press event.

#137 - 07/06/2015 05:03 PM - Greg Shah

> Should I try to reboot it(not sure if I have enough permissions)?

No, let the customer handle that.

> The exception from drop-down dismiss rule is mouse events in combo-box entry field or outside drop-down or combo-box(when drop-down is opened). In this case drop-down is closing on mouse press event.

Please fix that for us.

#138 - 07/06/2015 09:30 PM - Eugenie Lyzenko

Rebased task branch 2546a from P2J trunk revision 10894.

#139 - 07/06/2015 10:49 PM - Eugenie Lyzenko

Please review task branch 2546a, revision 10897.

Adding drop-down press opening fix. The main logic of drop-down opening is moved to mousePressed() method to reflect 4GL behavior on Windows. Also here added the code to dismiss drop-down when pressing second time into entry field or drop-down button if drop-down already opened.

Working on the fix for closing drop-down in multiple combo-box case.

#140 - 07/07/2015 07:59 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Please review task branch 2546a, revision 10897.

The only issue I have is with code like this: if (child != null && child instanceof ScrollableSelectionListGuiImpl) in ScrollPaneGuiImpl. I would prefer if you can find a way to use findMouseSource (override it if necessary) so that the source is the ScrollableSelectionListGuiImpl instead of ScrollPaneGuiImpl - this way the mouse action ends up directly where it belongs, without needing to route it.

#141 - 07/07/2015 01:56 PM - Eugenie Lyzenko

Constantin Asofiei wrote:

> The only issue I have is with code like this: if (child != null && child instanceof ScrollableSelectionListGuiImpl) in ScrollPaneGuiImpl. I would
> prefer if you can find a way to use findMouseSource (override it if necessary) so that the source is the ScrollableSelectionListGuiImpl instead of
> ScrollPaneGuiImpl - this way the mouse action ends up directly where it belongs, without needing to route it.

The reason why the ScrollableSelectionListGuiImpl is not becoming the mouse event target not the wrong/missing findMouseSource chaining. All
mouse events are routed by MouseHandler for widgets that registered by ID. ScrollableSelectionListGuiImpl has no config, no widget ID so only
nearest registered parent with ID can receive the mouse events(ScrollPaneGuiImpl in our case).

I would prefer not to create special config for ScrollableSelectionListGuiImpl, need to find the other way. As I can see all mouse event should be
passed through MouseHandler.handleMouseEvent(ID, MouseEvent)? This is mandatory condition to keep events consistency, correct?

#142 - 07/07/2015 02:41 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Constantin Asofiei wrote:
>
> > The only issue I have is with code like this: if (child != null && child instanceof ScrollableSelectionListGuiImpl) in ScrollPaneGuiImpl. I would
> > prefer if you can find a way to use findMouseSource (override it if necessary) so that the source is the ScrollableSelectionListGuiImpl
> > instead of ScrollPaneGuiImpl - this way the mouse action ends up directly where it belongs, without needing to route it.
>
> The reason why the ScrollableSelectionListGuiImpl is not becoming the mouse event target not the wrong/missing findMouseSource chaining.
> All mouse events are routed by MouseHandler for widgets that registered by ID. ScrollableSelectionListGuiImpl has no config, no widget ID so
> only nearest registered parent with ID can receive the mouse events(ScrollPaneGuiImpl in our case).
>
> I would prefer not to create special config for ScrollableSelectionListGuiImpl, need to find the other way. As I can see all mouse event should be
> passed through MouseHandler.handleMouseEvent(ID, MouseEvent)? This is mandatory condition to keep events consistency, correct?

Yes, mouse events pass through MouseHandler.

Your reasoning is good enough to leave it as is... I can't see an easy way out of this, without complicating things with new config classes.

#143 - 07/07/2015 07:54 PM - Eugenie Lyzenko

Please review task branch 2546a, revision 10898.

This update has new approach that fixes the multiple combo-box open/close issue when opening one combo-box if another is already opened. The idea is to defer mouse press event until old combo-box completely closing. The mouse press event is reposting to the event queue to be after LEAVE for previous one processing. Otherwise the old drop-down will remain opened. To separate the mouse events the special private flag has been introduced to mark the reposted artificial mouse events and avoid infinite loop in checking for leave requirements. Also the leave check is moved to early step where we can compute the real current focused widget before it can be changed by mouse event processing. This is the way to react on mouse press instead of click as it was implemented before.

The feedback and a bit more testing required to be sure the approach is completely safe and does not produce the message mess.

#144 - 07/08/2015 05:46 PM - Eugenie Lyzenko

Task branch 2546a for review updated to revision 10899.

The enhancements has been made to mouse event handling:
- only mouse press/release/click events are now involved into leave check and getting the current focus. The most of the mouse events are just move and get the current focus is complicated procedure so mouse move should be excluded.
- the event reposting must be done for all 3 mouse press related events because in a real life we must keep and preserve the click sequence - press/release/click. Otherwise it is possible to have bad side effects or other artifacts. We can not just eat release and click - some widgets can stuck on pressed state.

#145 - 07/09/2015 12:51 PM - Eugenie Lyzenko

- File cbb_test15_4GL_0_gui.jpg added

- File cbb_test15_p2j_0_gui.jpg added

Task branch 2546a for review updated to revision 10900.

The updates included change for drop-down button relocation for dynamically created combo-box. The dynamic combo-box is working itself.

The only issue I see for now is the proper location in case we have NOT side labeled widgets inside the frame the dynamic combo-box is placing. Compare two attached pictures of the same test in 4GL and P2J. As you can see in both cases the ROW coordinate of the combo-box is the same but real location is different. In P2J the combo-box is located at the real row it has and overlap the fill-in widget and it is expected because the fill-in is shifted down to draw the label. In P2J the screen coordinate of the combo-box has changed to avoid overlapping so combo-box:ROW is 4 I guess, not 3 as declared in message area.

I think this is not combo-box specific issue, but all dynamic widgets with explicit coordinates set are involved in this effect. So the question: do we need to implement this shift/adjust on client side?

The remaining TODO list for combo-box:
1. Check and implement if required the possibility to handle not character data by combo-box. I mean if the combo-box can handle character, integer pair as item list entry and return integer when value changed or getting current value.
2. Verify and implement all attribute and method for both static and dynamic combo-box.
3. Fix the knowing font scaling issue.

#146 - 07/09/2015 03:13 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546a for review updated to revision 10900.

I'm OK with the changes.

#147 - 07/09/2015 05:43 PM - Greg Shah

> 2. Verify and implement all attribute and method for both static and dynamic combo-box.

Can you let me know what you are thinking here? #2534 already has quite a few changes for combo-box. Are there other changes we need to implement that I am forgetting?

#148 - 07/09/2015 06:49 PM - Eugenie Lyzenko

Greg Shah wrote:

> Can you let me know what you are thinking here? #2534 already has quite a few changes for combo-box. Are there other changes we need to implement that I am forgetting?

This is the list I see.

Attributes:

AUTO-COMPLETION
AUTO-ZAP (must be implemented in 2534)
CURSOR-OFFSET (must be implemented in 2534)
DELIMITER
DISABLE-AUTO-ZAP
EDIT-CAN-PASTE
EDIT-CAN-UNDO
FORMAT
MAX-CHARS (must be implemented in 2534)
MENU-KEY
MENU-MOUSE
NUM-ITEMS
SELECTION-END
SELECTION-START
SELECTION-TEXT
SIDE-LABEL-HANDLE
SUBTYPE
TAB-POSITION
TEXT-SELECTED
UNIQUE-MATCH

Methods:

ADD-FIRST
ADD-LAST
CLEAR-SELECTION
DELETE
EDIT-CLEAR
EDIT-COPY
EDIT-CUT method
EDIT-PASTE
EDIT-UNDO
END-FILE-DROP
ENTRY
GET-DROPPED-FILE
INSERT
LOAD-MOUSE-POINTER
LOOKUP
MOVE-AFTER-TAB-ITEM (already working?)
MOVE-BEFORE-TAB-ITEM (already working?)
MOVE-TO-BOTTOM (already working?)
MOVE-TO-TOP (already working?)
REPLACE   (must be implemented in 2534)
SET-SELECTION   (must be implemented in 2534)
VALIDATE (already working?)

What I'm thinking is in first place we need to implement methods/attribute that are specific for combo-box or internal parts(selection-list or fill-in). All other attributes/methods should be implemented on the more generic level(common for given attribute or method).

#149 - 07/09/2015 06:54 PM - Eugenie Lyzenko

1. Check and implement if required the possibility to handle not character data by combo-box. I mean if the combo-box can handle character, integer pair as item list entry and return integer when value changed or getting current value.

This feature is already working.

#150 - 07/09/2015 11:35 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> The only issue I see for now is the proper location in case we have NOT side labeled widgets...

Please commit the tests you are using for the screens you posted...

#151 - 07/09/2015 11:54 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> I think this is not combo-box specific issue, but all dynamic widgets with explicit coordinates set are involved in this effect. So the question: do we need to implement this shift/adjust on client side?

I think you are correct - P2J uses the ROW relative as to the top frame border, while 4GL uses the ROW as relative to the top frame header (row 1 is just below the frame header). This is related to layout.

#152 - 07/10/2015 07:54 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

> Please commit the tests you are using for the screens you posted...

Done. Committed to revision 1289.

#153 - 07/10/2015 08:04 AM - Eugenie Lyzenko

- File cbb_test15_1_4GL_0_gui.jpg added

> I think you are correct - P2J uses the ROW relative as to the top frame border, while 4GL uses the ROW as relative to the top frame header (row 1 is just below the frame header). This is related to layout.

I guess the situation is a bit more complicated. Take a look at one more screen attached from 4GL system (combo_box15_1.p). The ROW coordinate for dynamic combo-box is the same - 3. But the location is different comparing to the previous test(combo_box15.p). The difference is the test combo_box15_1.p uses SIDE-LABELS frame option while in combo_box15.p the frame without side labels option. I think 4GL makes the row reposition at runtime on the layout stage so the screen coordinate for combo_box15.p is actual 4 row instead of requested 3 row.

#154 - 07/10/2015 08:54 AM - Greg Shah

> This is the list I see.

> ...

Yes, I know that we have many attributes and methods that are not yet finished. But we cannot work on these now:

1. They are not needed for our current conversion projects.
2. Our current conversion projects do need other GUI features to be finished (e.g. other widgets).

Thus we must prioritize other work higher than these.

Don't worry about the methods and attributes.

#155 - 07/10/2015 08:58 AM - Greg Shah

> I think 4GL makes the row reposition at runtime on the layout stage so the screen coordinate for combo_box15.p is actual 4 row instead of
> requested 3 row.

Constantin will add this to the layout work he is doing right now.

#156 - 07/10/2015 09:29 AM - Greg Shah

Eugenie: the only attribute that needs implementation is MAX-CHARS. That was something that could not be done in [#2534](#) because it is
GUI-specific. Please add support for that in this task.

#157 - 07/10/2015 09:38 AM - Eugenie Lyzenko

Greg Shah wrote:

> Eugenie: the only attribute that needs implementation is MAX-CHARS. That was something that could not be done in [#2534](#) because it is
> GUI-specific. Please add support for that in this task.

OK.

#158 - 07/10/2015 10:33 AM - Eugenie Lyzenko

The testcase set updated to rev 1290. The only change is in combo_box15.p. For dynamic widget we have to place INNER-LINES setting before VISIBLE and SENSITIVE to have proper scrollbar. This is the effect after changes from Igor in INNER-LINES attribute implementation.

#159 - 07/10/2015 10:34 AM - Greg Shah

For dynamic widget we have to place INNER-LINES setting before VISIBLE and SENSITIVE to have proper scrollbar. This is the effect after changes from Igor in INNER-LINES attribute implementation.

Is this different from how the 4GL works?

#160 - 07/10/2015 11:19 AM - Eugenie Lyzenko

Is this different from how the 4GL works?

Yes in 4GL there is no such condition, INNER-LINES setting can be after VISIBLE. I think 4GL accumulates all create widget... internal assignments and makes single realization. In P2J every assignment inside create widget... causes the pushScreenDefinition() call and almost everything after VISIBLE setter are coming when widget already created and initialized. And it is a bit too late. So we start dynamic widget painting when not all requested attributes are properly set.

#161 - 07/10/2015 11:39 AM - Greg Shah

In P2J every assignment inside create widget... causes the pushScreenDefinition() call and almost everything after VISIBLE setter are coming when widget already created and initialized. And it is a bit too late. So we start dynamic widget painting when not all requested attributes are properly set.

For static frames, we handle this in GenericFrame.setupFrame(). In other words, in GenericFrame.pushScreenDefinition() we defer the actual call to the client, until the batch flag is false.

For CREATE WIDGET we need to do the same thing, right?

#162 - 07/10/2015 12:09 PM - Eugenie Lyzenko

For static frames, we handle this in GenericFrame.setupFrame(). In other words, in GenericFrame.pushScreenDefinition() we defer the actual call to the client, until the batch flag is false.

For CREATE WIDGET we need to do the same thing, right?

Yes, you are right, we need to defer real pushing until all done.

Consider the generated P2J code for dynamic widget(from combo_box15.p):

```
...
            DynamicWidgetFactory.createComboBox(hcbb);
            hcbb.unwrapWidget().setFrameHandle(cbbFrFrame.asWidget());
            hcbb.unwrapCommonField().setFormat(new character("x(13)"));
            hcbb.unwrapCommonListWidget().setListItems(new character("Item 0,Item 1,Item 2,Item 3,Item 4,Item
5,Item 6,Item 7,Item 9,Item A,Item B,Item C,Item D,Item E,Item F"));
            hcbb.unwrapInnerLines().setInnerLines(new integer(9));
            hcbb.unwrapSensitive().setSensitive(new logical(true));
            hcbb.unwrapWidget().setVisible(new logical(true));
            hcbb.unwrapWidget().setRow(new integer(3));
            hcbb.unwrapWidget().setColumn(new integer(4));
...
```

It is clear when start batching but it is not clear when to stop it. Every CREATE WIDGET... internal assignment is a standalone command after dynamic widget creation. We need to generate special marker indicating the last internal assignment in CREATE WIDGET... may be.

#163 - 07/10/2015 12:10 PM - Eugenie Lyzenko

Rebased task branch 2546a from P2J trunk revision 10895, new branch revision is 10902.

#164 - 07/10/2015 01:29 PM - Greg Shah

We need to generate special marker indicating the last internal assignment in CREATE WIDGET... may be.

Correct. This is easy to do. Use an ascent rule and match on the KW_ASSIGN that is the child of the CREATE_WIDGET. When we ascend back to that node, all setters have been called and the batch can be closed.

This is very similar to what we do with RecordBuffer batching during the ASSIGN statement.

#165 - 07/10/2015 05:38 PM - Eugenie Lyzenko

I put the following rule into convert/ui_statements.rules(end of ascent-rules section):

```
...
      <rule>type == prog.kw_assign and parent.type == prog.create_widget
         <action>methodText = "DynamicWidgetFactory.pushScreenUnlock"</action>
         <action>methodType = java.static_method_call</action>
         <action>
            createPeerAst(methodType, methodText, closestPeerId)
         </action>
      </rule>
...
```

But nothing is emitting in converted code after dynamic widget internal assignments. Please clarify what I'm doing wrong?

#166 - 07/13/2015 08:46 AM - Greg Shah

Try using createJavaAst() instead of createPeerAst().

#167 - 07/13/2015 12:02 PM - Constantin Asofiei

Greg Shah wrote:

> Eugenie: the only attribute that needs implementation is MAX-CHARS. That was something that could not be done in #2534 because it is GUI-specific. Please add support for that in this task.

For EDITOR, MAX-CHARS was implemented as part of #2563 (conversion + runtime). Conversion + server-side part should be common.

#168 - 07/13/2015 01:25 PM - Eugenie Lyzenko

For MAX-CHARS implementation in COMBO-BOX:

In all modes, the attribute is ignoring, MAX-CHAR maximum value is 32767 then the attribute become negative. Can be set to negative value but not in define widget statement(it can be set to negative in define widget statement if the value is more than 32767) otherwise the application will not even be compiled.

So I can conclude the attribute is 2 bytes signed integer and absolutely useless. What we have to do to support it is to make proper value wrapping around 0 for the value to be set. This can be done on the server side no changes for client side code. For ChUI mode the behavior is the same. Looks like in 4GL this attribute is something that is waiting to be implemented.

#169 - 07/13/2015 01:44 PM - Eugenie Lyzenko

More finding for MAX-CHARS in 4GL COMBO-BOX.

The default value is 0, not 255, like as the reaction to setting to unknown(?) value is also 0. For both GUI and ChUI modes.

#170 - 07/13/2015 05:38 PM - Eugenie Lyzenko

Task branch 2546a for review updated to revision 10903.

This update includes the changes required to complete the MAX-CHARS attribute for combo-box(for the server side). The idea behind is to change the internal data type for attribute storage to signed short to automatically wrap the negative-positive shift. Also the default value has changed from 255 to 0 to reflect the current 4GL behavior.

As for 4GL it is possible to assign the value more than 65536. In this case the bitwise AND is performing to truncate everything outside two first bytes before assigning the new value.

#171 - 07/13/2015 09:19 PM - Eugenie Lyzenko

Task branch 2546a for review updated to revision 10904.

This update adds batch mode lock when widget attributes are set with create widget statement. The rule condition was modified for ascent rule to properly work. When the last setter is calling the DynamicWidgetFactory executes pushScreenUnlock() method. This method sets the previously saved frame into batch(false, true) state. This release screen definition pushing and force sending all deferred screen definitions to the client side. The moment when this lock is started is the time when dynamic widget gets the valid frame handle. This frame is stored for further push unlock call. Not sure, may be it is better to implement all unlock code in GenericFrame class, so please give your feedback.

#172 - 07/14/2015 04:00 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546a for review updated to revision 10904.

Consider that the widget's FRAME is not necessarily set at its CREATE ... stmt. 4GL allows code like this:

```
DEF VAR h AS HANDLE.

CREATE FILL-IN h.
h:BGCOLOR = 3.
h:FGCOLOR = 4.
```

```
h:ROW = 1.
h:COLUMN = 2.

DEFINE FRAME f1.
h:FRAME = FRAME f1:HANDLE.

ENABLE ALL WITH FRAME f1 SIZE 20 BY 20.
WAIT-FOR GO OF FRAME f1 .
```

Is this compatible with your current approach?

In 4GL is not mandatory to set the FRAME at the CREATE's ASSIGN clause: so, please emit the pushScreenUnlock() only if the ASSIGN is setting the widget's FRAME attribute.

Also, when setting the widget's FRAME attribute, you need to put the frame in "batch" mode only and only if this assignment is from the CREATE's ASSIGN clause - otherwise, batch mode will never be unset, as there will never be a pushScreenUnlock.

You may want to track if the CREATE's ASSIGN has a FRAME attribute, and if so, bracket the entire ASSIGN as in:

```
          DynamicWidgetFactory.createComboBox(hcbb);
          hcbb.unwrapWidget().pushScreenLock();
...
          hcbb.unwrapWidget().pushScreenUnlock();
```

This way, a pending "lock" is registered with the widget; when its FRAME is set, will see that there is a pending lock and will set the frame in batch mode. When the lock will be released, the frame batch mode is reset and all changes are pushed to the client.

#173 - 07/14/2015 04:03 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546a for review updated to revision 10903.

I'm OK with these changes.

#174 - 07/14/2015 04:13 PM - Greg Shah

- Status changed from New to WIP

Wouldn't it be better to always bracket CREATE WINDOW assign by setting batch mode on in GenericFrame and then turning it off at the end of the ASSIGN?

#175 - 07/14/2015 04:22 PM - Greg Shah

I should say this a different way.

I understand that we can't depend on there being a FRAME defined in the ASSIGN. The solution must tolerate this.

But I don't want to require the programmer (that maintains this converted code) to remember to do an unlock if there is a FRAME attribute every added to the assignments later. It seems fragile.

We should be able to bypass pushing screen definitions until the frame is set. Batch mode should always be present for the ASSIGN processing as a set. The combination of both of these (frame is set and we exit batch mode) must come to pass before the screen def is pushed.

#176 - 07/14/2015 04:38 PM - Constantin Asofiei

Greg Shah wrote:

> We should be able to bypass pushing screen definitions until the frame is set.

Yes, this is already done, pushScreenDefinition is not performed if the GenericWidget.frame is not set. The problem is when multiple attribute assignments need to be performed at the same time, in "batch mode".

> Batch mode should always be present for the ASSIGN processing as a set. The combination of both of these (frame is set and we exit batch mode) must come to pass before the screen def is pushed.

Then the alternative is to make the pushScreenUnlock() call mandatory for a CREATE ... ASSIGN stmt (as Eugenie already implemented it). To cover the case of setting the FRAME attr in and outside the CREATE ... ASSIGN clause, we can:

1. turn a "pending batch mode" flag at the dynamic widget, right after it is created. But what if the CREATE has no ASSIGN clause? I don't think we can get away without some way of marking the beginning of the CREATE ... ASSIGN - we can make the pushScreenLock mandatory, too...
2. if FRAME is set at the CREATE ... ASSIGN, the frame is placed in batch mode
3. emit pushScreenUnlock() after the last assignment: if FRAME was set, then it will proceed with pushing the widget definition to the frame. Also, the widgets "pending batch mode" flag is set to false (regardless if the FRAME was set or not)
4. if FRAME attribute is set outside CREATE ... ASSIGN, then this will see that there is no "pending batch mode" and will execute the pushScreenDefinition immediately.

#177 - 07/14/2015 04:41 PM - Greg Shah

To cover the case of setting the FRAME attr in and outside the CREATE ... ASSIGN clause, we can:
...

Yes, good.

#178 - 07/14/2015 07:46 PM - Eugenie Lyzenko

Can we defer all screen definition pushing until first GenericFrame.enable() call? Or other frame related call that makes it visible?

#179 - 07/14/2015 07:49 PM - Eugenie Lyzenko

Task branch 2546a for review updated to revision 10905.

This small update fixes the dynamic combo-box behavior when there are no initial selection for entry field until first selection is performed.

#180 - 07/15/2015 01:59 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

Can we defer all screen definition pushing until first GenericFrame.enable() call? Or other frame related call that makes it visible?

I don't think is OK to do this; there are attributes which, when set, force widget realization and deferring the screen definition push may interfere with those. More, once the FRAME attribute is set, the dynamic widgets should behave the same way as a static widget, when their attributes are set.

#181 - 07/15/2015 03:47 PM - Eugenie Lyzenko

Some consideration for introducing the push screen definition lock to discuss.

I would suggest another way of implementation. As we know this fix was inspired by the fact the INNER-LINES attribute stops working with dynamic widget. There are no more issues found with the current push screen code. I have performed additional investigation and found the code for setting INNER-LINES has wrong behavior itself. Consider this(ComboBoxWidget.java):

```
...
  public void setInnerLines(double lines)
  {
    int value = (int) lines;
    if (value < 0)
    {
      ErrorManager.recordOrShowError(4082,
            String.format(
            "Attribute INNER-LINES must be greater than or equal to 0 on %s %s",
            type(), widgetName()),
            false);
      return;
```

```
          }

      config.innerLinesAssignments++;
      if (!config.realized) <-- This prevents real attribute setting
      {
          config.innerLines = value;
      }
      else
      {
          config.innerLines = Math.min(value,
                  config.innerLinesLimit - config.innerLinesAssignments);
      }
      if (frame != null)
      {
          config.heightChars = config.innerLines + 2;
      }
      pushScreenDefinition();
   }
...
```

My testing in 4GL system shows no matter when to set the INNER-LINES for dynamic widget, before or after realization - the value will be set properly.

So my suggestion is to change the code above to:

```
...
      if (!config.realized || config.dynamic) <-- This completely fixes the issue
      {
          config.innerLines = value;
      }
...
```

With this fix we do not need to implement push screen lock for now. Yes, I understand we need this functionality anyway but I suggest to defer this implementation because this is not combo-box specific feature. And, which is more important, the lock implementation is pretty complicated, it can take a time to have perfect solution because too many conditions to consider to have the good code. I do not want to hurry and make fast possibly buggy prone code.

What do you think? Is this acceptable?

#182 - 07/15/2015 08:20 PM - Eugenie Lyzenko

Task branch 2546a for review updated to revision 10906.

In this update:
1. According to my previous suggest the push screen lock implementation has been undone.
2. The respective fix for dynamic combo-box and INNER-LINES attribute included.
3. The fix for incorrect ID assignment while setting ITEM-LIST attribute outside define ... combo-box statement included.

The fix for INNER-LINES was tested and working fine. So with this update the remaining thing is scaling issue I'm currently working on.

#183 - 07/16/2015 10:08 AM - Greg Shah

> Some consideration for introducing the push screen definition lock to discuss.
> ...

Constantin will handle the push screen definition lock. Your approach is fine for combo-box.

#184 - 07/16/2015 09:04 PM - Eugenie Lyzenko

Task branch 2546a for review updated to revision 10907.

The update has improved implementation for width scaling issue. Something made more simple, something - more complicated. For explicitly width setting the scaling is 1.0. For other cases the scaling is computing for width more than 10.0 where the mismatch become significant. And some chars added to reserve space for drop-down button.

I consider this as the candidate for testing if there are no objections.

#185 - 07/17/2015 06:51 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546a for review updated to revision 10907.

> The update has improved implementation for width scaling issue.

The remaining task is computing the combo's dimension as in 4GL. Consider the test at note 130 with a font8 set to Tahoma, size 14 - the combo's width/height are not computed properly (the widget's area is significantly smaller than what 4GL draws). More, the combo's dimension attributes are reported as 0 in P2J, they do not reflect the client-side (implicit) computed value.

So:

1. if all issues (except dimension) are fixed, please get it into regression testing and release it
2. start working on the widget's dimension; read notes 92/98 and determine a formula. Once you have a standalone java program which validates the formula with all the captured 4GL metrics, then you can integrate it into the P2J code. To reiterate, why we need this to be 100% accurate:

implicit widget's dimension is used by automatic layout; if we get the widget's dimension wrong, the layout will be wrong. The only part where we can compromise is actual text drawing: we can't match 100% text metrics, so we do a "best off".

#186 - 07/17/2015 08:39 AM - Greg Shah

if all issues (except dimension) are fixed, please get it into regression testing and release it

Constantin: he is going to need to rebase after your merge to trunk. There are overlapping changes between your update and his. Do you want him to start testing before your merge to trunk?

#187 - 07/17/2015 08:59 AM - Constantin Asofiei

Greg Shah wrote:

> if all issues (except dimension) are fixed, please get it into regression testing and release it

> Constantin: he is going to need to rebase after your merge to trunk. There are overlapping changes between your update and his. Do you want him to start testing before your merge to trunk?

I'll merge to trunk in ~2-3 hours, after testing completes. So is better to rebase first and test after this.

#188 - 07/17/2015 12:23 PM - Eugenie Lyzenko

The remaining task is computing the combo's dimension as in 4GL. Consider the test at note 130 with a font8 set to Tahoma, size 14 - the combo's width/height are not computed properly (the widget's area is significantly smaller than what 4GL draws). More, the combo's dimension attributes are reported as 0 in P2J, they do not reflect the client-side (implicit) computed value.

Constantin,

Can you upload here the picture for this mismatch?

#189 - 07/17/2015 02:06 PM - Eugenie Lyzenko

Rebased task branch 2546a from P2J trunk revision 10897 (new branch revision 10908 - a few words added in comments for one file).

Preparing to start regression runtime testing.

#190 - 07/17/2015 04:52 PM - Constantin Asofiei

- File combo_tahoma_14.png added

Eugenie Lyzenko wrote:

> The remaining task is computing the combo's dimension as in 4GL. Consider the test at note 130 with a font8 set to Tahoma, size 14 - the combo's width/height are not computed properly (the widget's area is significantly smaller than what 4GL draws). More, the combo's dimension attributes are reported as 0 in P2J, they do not reflect the client-side (implicit) computed value.

> Constantin,

> Can you upload here the picture for this mismatch?

See attached - no side-labels, as it is easier to see the differences.

Also, somethign else: P2J automatically selects first element in this test, while 4GL does not...

#191 - 07/18/2015 09:20 AM - Eugenie Lyzenko

The main part for 2546a finished without regressions, running the CTRL-C one.

#192 - 07/18/2015 11:34 AM - Eugenie Lyzenko

Testing completed without regressions. Had to start CTRL-C 3-way tests in a separate session. The results are in 2546a_10908_bb2f7ec_20150718_evl.zip.

So the branch 2546a is ready to be merged into trunk.

#193 - 07/18/2015 11:46 AM - Eugenie Lyzenko

Greg,

According to handbook rules I'm waiting for approval confirmation to start merging the branch into the trunk.

#194 - 07/20/2015 08:14 AM - Greg Shah

Please merge task branch 2546a to trunk, commit and send the notification email.

#195 - 07/20/2015 10:28 AM - Eugenie Lyzenko

Also, somethign else: P2J automatically selects first element in this test, while 4GL does not...

For this issue, font scaling and all issues I'll possibly found during this work I'm going to create new branch 2645b.

#196 - 07/20/2015 10:41 AM - Eugenie Lyzenko

The branch 2546a has been merged and committed to trunk as revision 10897. Archived.

#197 - 07/20/2015 10:41 AM - Eugenie Lyzenko

Created task branch 2546b from P2J trunk revision 10897.

#198 - 07/20/2015 05:49 PM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10898.

The fixes including:
1. Fix for wrong initial selection the item in combo-box when nothing is specified as initial value.
2. Rework for selected flag setting of the ComboBox.java to be used in ComboBoxGuiImpl.java removing the same flag in ComboBoxGuiImpl.java to unify the classes.
3. Fixed NPE for getting event list in certain condition in EventList.getNonExitEventsForWidget(), adding pointer check and including ANYWHERE events into return list.
4. Fixed array out of bound exception in ScrollableSelectionListGuiImpl(). Adding negative array index protection.

Continue working on font scaling issue.

#199 - 07/21/2015 06:30 PM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10899.

This is the alternative scaling implementation for combo-box widget in GUI. The test combo_box14.p now works as expected. The picture is attached.

About WIDTH-PIXELS attribute that is 0. I guess this is the common widget feature, not combo-box specific for the cases we do not explicitly specify this value. Do we need this to be implemented as a part of the combo-box implementation?

#200 - 07/21/2015 06:31 PM - Eugenie Lyzenko

- File cbb_test14_p2j_20150722_gui.jpg added

The picture...

#201 - 07/22/2015 04:23 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546b for review updated to revision 10899.
>
> This is the alternative scaling implementation for combo-box widget in GUI. The test combo_box14.p now works as expected. The picture is attached.

Have you proved the formula using the captured metrics?

> About WIDTH-PIXELS attribute that is 0. I guess this is the common widget feature, not combo-box specific for the cases we do not explicitly specify this value. Do we need this to be implemented as a part of the combo-box implementation?

No, other widgets work fine. The problem is with the combo-box implementation.

#202 - 07/22/2015 09:20 AM - Eugenie Lyzenko

> No, other widgets work fine. The problem is with the combo-box implementation.

Yes, you are right. And this is strange because the width setter is not widget specific. OK, I'll investigate this point.

#203 - 07/22/2015 03:37 PM - Constantin Asofiei

Eugenie, please check the uast/demo/movie-ratings-static.p - the combo does not behave OK, the drop-down list is not shown.

#204 - 07/22/2015 05:00 PM - Eugenie Lyzenko

- File movie-ratings-demo.jpg added

Task branch 2546b for review updated to revision 10900.

Fixed the WIDTH-PIXELS attribute getting issue.

Eugenie, please check the uast/demo/movie-ratings-static.p - the combo does not behave OK, the drop-down list is not shown.

In my build(2546b version 10900) it works, see the picture.

#205 - 07/22/2015 06:33 PM - Eugenie Lyzenko

Have you proved the formula using the captured metrics?

Constantin, let's clarify this point. Is there the approach to do this? What is the prove criteria? I see the CheckMetriscCb.java application that should compute the expected pixel width. I need to use scaling approach to compute expected width for single char in pixels? Is this what you mean? Or something other?

#206 - 07/23/2015 03:51 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Have you proved the formula using the captured metrics?

> Constantin, let's clarify this point. Is there the approach to do this? What is the prove criteria? I see the CheckMetriscCb.java application that should compute the expected pixel width. I need to use scaling approach to compute expected width for single char in pixels? Is this what you mean? Or something other?

See the testcases/uast/fillin/CheckMetricsFillin.java for an example - the widget width is not dependent on the "single char width in pixels"; instead is related to the format length, font type (fixed/non-fixed), if the font is explicit or not, font size, etc.

What you need to do is this: determine the relation between the format length and the legacy font metrics (or even PIXELS-PER-COLUMN) so that your compute widget width (in pixels) matches the legacy widget width. For start, you can try using legacy avg font width multiplied by the format length and see what it gives you - I suspect some cases will match, and some others will not. Also, for combo-box, there may be some constant values for the width of the right-side knob and some other insets (border, for example).

#207 - 07/23/2015 04:37 AM - Eugenie Lyzenko

Some suggestions:

1. What is the "base" unit in GUI mode? Correct me if I'm wrong but looks like it is still chars. We form the main window, frame, do the layout(this is especially significant) considering coordinates and size using character units. The pixel units are used when widget is about to be drawn. The pixel coords are derived from the character ones. Correct?

2. However the pixels-chars coordinates are mutually dependent on the widget config level. If we change pixel coordinates - the character coordinates will be changed too(based on the default font width/height, default font number is -1) and vise versa. The main application size is calculated using the default font, right?

3. What you did for FILL-IN widget is to implement the algorithm to convert the widget size given in character units to pixel based units to perform correct painting. Another word if say the FILL-IN width is 10 chars in ChUI mode - in GUI mode it will have the same 10 character in width. Is it true?

I need to clarify this because the approach I'm using is not the same as I suggest for yours in #3 above.

#208 - 07/23/2015 05:55 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Some suggestions:
>
> 1. What is the "base" unit in GUI mode?

The base unit in GUI is the pixel - all clipping, ScreenBitmap, etc is done using pixels.

> Correct me if I'm wrong but looks like it is still chars. We form the main window, frame, do the layout(this is especially significant) considering coordinates and size using character units. The pixel units are used when widget is about to be drawn. The pixel coords are derived from the character ones. Correct?

The character units are dependent on the current DEFAULT-FONT, from which the PIXELS-PER-ROW/COLUMN values are computed. This gives you the number of pixels for 1 row or 1 column, in character units, used as default. From there on, once you have the widget's dimension in pixels, you can compute its equivalent dimension in character units.

Yes, the layout is done using character units - but these are computed AFTER the widget's width/height (in pixels) is resolved, using the widget's explicit or implicit font. After this, the pixel values are divided by the PPR/PPC constants, and the character values are resolved. CoordinatesConversion.width/heightFromNative is responsible for this.

> 2. However the pixels-chars coordinates are mutually dependent on the widget config level. If we change pixel coordinates - the character coordinates will be changed too(based on the default font width/height, default font number is -1) and vise versa.

Yes, they are dependent. If you change one, the other one is computed automatically - AspectJ instrumentation is used to automate this. See the classes from the p2j.aspects.ui package.
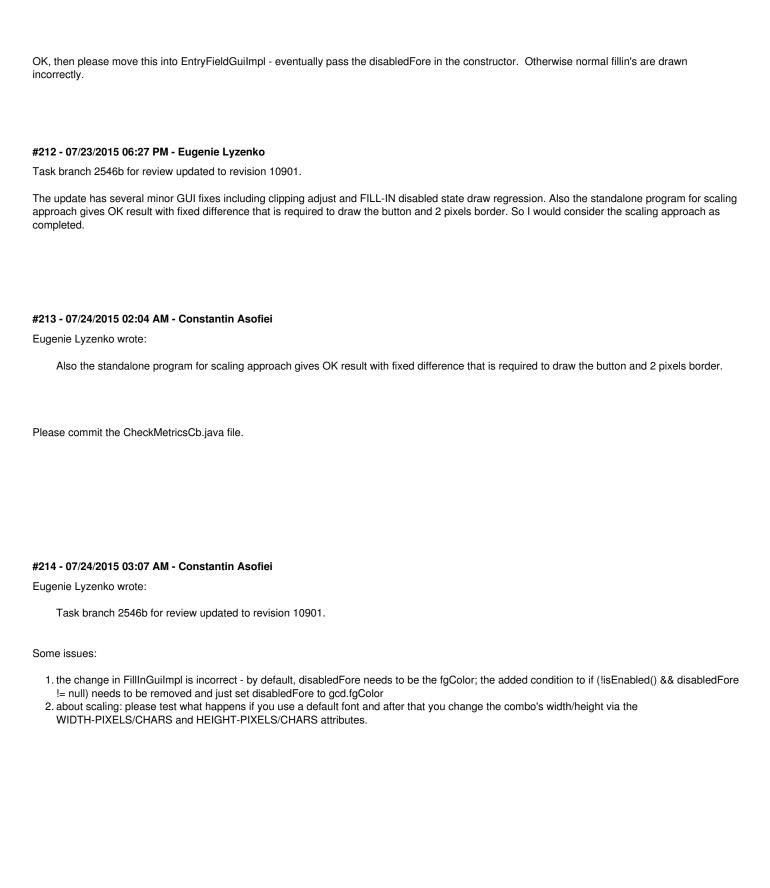
> The main application size is calculated using the default font, right?

If there is no explicit font at the frame/widget, then yes, the DEFAULT-FONT (or DEFAULT-FIXED-FONT, in case of widgets with numeric format) is used. For combo-box, I think only the DEFAULT-FONT is needed.

3. What you did for FILL-IN widget is to implement the algorithm to convert the widget size given in character units to pixel based units to perform correct painting.

No, the start is always the pixel dimension - if an explicit font is used, the pixel dimension will be independent of the DEFAULT-FONT. After the pixels values are found, the character values are computed using CoordinatesConversion APIs, which use the PIXELS-PER-ROW/COLUMN values to convert it.

Another word if say the FILL-IN width is 10 chars in ChUI mode - in GUI mode it will have the same 10 character in width. Is it true?

This is true in both GUI and ChUI only if you assign explicitly 10 chars to the widget's width. What is complicated is the implicit width, when only the widget's format and font is known: in this case, the char width in ChUI and GUI will be different. So, these will be the same only and only if you set them explicitly - the implicit values are almost always different.

#209 - 07/23/2015 04:30 PM - Constantin Asofiei

Eugenie, about H009 in FillinGuiImpl: does the FILL-IN in case of combo-box show a gray foreground when disabled? Because for normal FILL-IN's, the default foreground color is always black... only EDITOR uses a gray color when disabled.

#210 - 07/23/2015 05:14 PM - Eugenie Lyzenko

Constantin Asofiei wrote:

Eugenie, about H009 in FillinGuiImpl: does the FILL-IN in case of combo-box show a gray foreground when disabled?

Yes, it does.

#211 - 07/23/2015 05:16 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

Constantin Asofiei wrote:

Eugenie, about H009 in FillinGuiImpl: does the FILL-IN in case of combo-box show a gray foreground when disabled?

Yes, it does.

OK, then please move this into EntryFieldGuiImpl - eventually pass the disabledFore in the constructor. Otherwise normal fillin's are drawn incorrectly.

#212 - 07/23/2015 06:27 PM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10901.

The update has several minor GUI fixes including clipping adjust and FILL-IN disabled state draw regression. Also the standalone program for scaling approach gives OK result with fixed difference that is required to draw the button and 2 pixels border. So I would consider the scaling approach as completed.

#213 - 07/24/2015 02:04 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Also the standalone program for scaling approach gives OK result with fixed difference that is required to draw the button and 2 pixels border.

Please commit the CheckMetricsCb.java file.

#214 - 07/24/2015 03:07 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546b for review updated to revision 10901.

Some issues:

1. the change in FillInGuiImpl is incorrect - by default, disabledFore needs to be the fgColor; the added condition to if (!isEnabled() && disabledFore != null) needs to be removed and just set disabledFore to gcd.fgColor
2. about scaling: please test what happens if you use a default font and after that you change the combo's width/height via the WIDTH-PIXELS/CHARS and HEIGHT-PIXELS/CHARS attributes.

#215 - 07/24/2015 11:44 AM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10902.

The fix for fill-in disabled color regression.

#216 - 07/24/2015 01:34 PM - Constantin Asofiei

Eugenie, about combo_box10.p, there are some issues:

- TAB pressing in the combo's fill-in is not switching focus
- I can't find a way to press the EXIT button via mouse - is this working in rev 10898?
- when pressing TAB in the ch fill-in the app is terminating

#217 - 07/24/2015 01:35 PM - Constantin Asofiei

Constantin Asofiei wrote:

- I can't find a way to press the EXIT button via mouse - is this working in rev 10898?

I'm rephrasing this: EXIT button does not terminate program when is pressed via mouse.

#218 - 07/24/2015 01:45 PM - Eugenie Lyzenko

Constantin Asofiei wrote:

Eugenie, about combo_box10.p, there are some issues:

- TAB pressing in the combo's fill-in is not switching focus

Yes, I know. There is a key processing issues in combo-box simple mode(when fill-in is integrated). It is happening after selection some item from the list and trying to edit new item in entry field. Working on this. TAB key issue is new one for me. I'll debut this too.

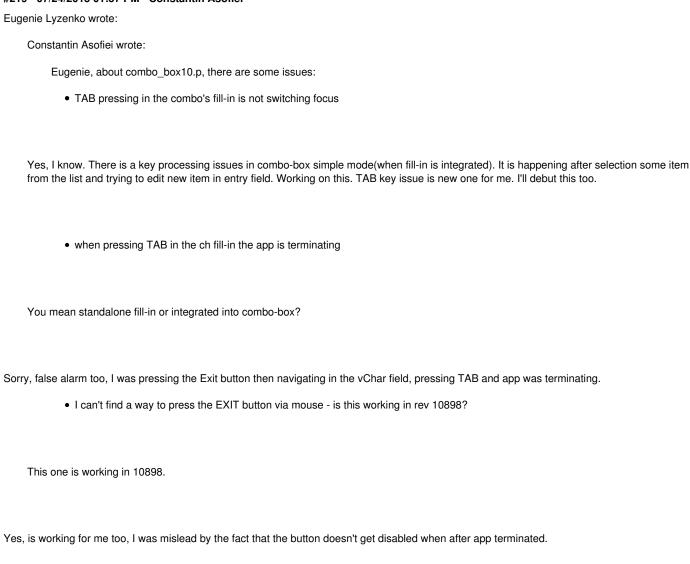- when pressing TAB in the ch fill-in the app is terminating

You mean standalone fill-in or integrated into combo-box?

- I can't find a way to press the EXIT button via mouse - is this working in rev 10898?

This one is working in 10898.

#219 - 07/24/2015 01:57 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Constantin Asofiei wrote:
>
>> Eugenie, about combo_box10.p, there are some issues:
>>
>> - TAB pressing in the combo's fill-in is not switching focus
>
> Yes, I know. There is a key processing issues in combo-box simple mode(when fill-in is integrated). It is happening after selection some item from the list and trying to edit new item in entry field. Working on this. TAB key issue is new one for me. I'll debut this too.
>
>> - when pressing TAB in the ch fill-in the app is terminating
>
> You mean standalone fill-in or integrated into combo-box?

Sorry, false alarm too, I was pressing the Exit button then navigating in the vChar field, pressing TAB and app was terminating.

> - I can't find a way to press the EXIT button via mouse - is this working in rev 10898?
>
> This one is working in 10898.

Yes, is working for me too, I was mislead by the fact that the button doesn't get disabled when after app terminated.

#220 - 07/24/2015 03:09 PM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10903.

The small fix in scrollbar location for border-less panel.

#221 - 07/24/2015 04:08 PM - Eugenie Lyzenko

Rebased task branch 2546b from P2J trunk revision 10900 (new branch revision 10906).

#222 - 07/24/2015 09:01 PM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10907.

The update has fix for unable to edit new selected item from simple list. In this case ScrollableSelectionListGuiImpl should bypass leave generation for outside mouse click. Continue working on TAB key issue for entry field.

#223 - 07/27/2015 11:26 AM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10908.

This is the TAB key processing fix. Preparing to rebase and clean up/more testing.

#224 - 07/27/2015 12:09 PM - Eugenie Lyzenko

Rebased task branch 2546b from P2J trunk revision 10901, new branch revision is 10909.

#225 - 07/27/2015 01:28 PM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10910.

Some cleanup, code unifying and testing performed. I guess can be tested for merge if no objections.

BTW, I have found the BACK_TAB key handling does not work as expected. And this is NOT combo-box specific, this is true for frame without combo-box.

#226 - 07/27/2015 03:43 PM - Eugenie Lyzenko

Constantin Asofiei wrote:

> Eugenie Lyzenko wrote:
> Please commit the CheckMetricsCb.java file.

Done.

#227 - 07/27/2015 08:15 PM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10911.

This changes for the following note:

> please test what happens if you use a default font and after that you change the combo's width/height via the WIDTH-PIXELS/CHARS and HEIGHT-PIXELS/CHARS attributes.

It is possible to change the width, here we need to adjust button location for every painting. Unfortunately, this is not very smart but because the width can be changed at any time - we have to go this way. But the widget height can not be changed after realization - we need to generate the error message. Appropriate change was made for CombiBoxWidget server side class.

#228 - 07/28/2015 07:40 AM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10912.

The error message tool for ComboBoxWidget is changed to recordOrShowError to be able to display message box.

#229 - 07/29/2015 10:15 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546b for review updated to revision 10912.

- ComboBoxGuiImpl:497 - please fix in SMILE or DROP-DOWN mode comment.

Otherwise, you need to change your scaling approach in ComboBoxGuiImpl so that the dimension is computed using the formula you found and proved in the CheckMetricsCb.java program: the width for this widget is as simple as formatLength * avgFontWidth + 28. Your current approach is over-complicated and we can't be sure it covers all cases safely. More, you are assuming that k gives you the average font width, but there is already an API for this: FontManager.getFontWidth.

And again: we need to match the native dimension in pixels as given by the widget's font (which may or may not be specified), as the character dimension is dependent on the DEFAULT-FONT currently in use.

#230 - 07/29/2015 01:42 PM - Eugenie Lyzenko

> Otherwise, you need to change your scaling approach in ComboBoxGuiImpl so that the dimension is computed using the formula you found and proved in the CheckMetricsCb.java program: the width for this widget is as simple as formatLength * avgFontWidth + 28. Your current approach is over-complicated and we can't be sure it covers all cases safely. More, you are assuming that k gives you the average font width, but there is already an API for this: FontManager.getFontWidth.

Yes, we can use simple algorithm I've found with all cases not using "k"-based calculations. This gives the pretty good result.

But the main issue on my guess is wrong calculation(or not enough precision) of the FontManager.getFontWidth(). I think it should be placed in double value, not int. For average font calculation itself. The approach when we sum the width of all characters and divide to the number of them is not correct(or does not fit our needs) I think. It can be mathematically correct from "average" definition but can be useless in our case, when the correct value can be (Min + Max)/2. At least we need to change the precision for average value I guess. The difference between 5 and 6 for example is too big to ignore, need to have fractional values between 5 and 6.

> And again: we need to match the native dimension in pixels as given by the widget's font (which may or may not be specified), as the character dimension is dependent on the DEFAULT-FONT currently in use.

I see, my complication is the attempt to adjust a bit for rare cases (font -1 && config().widthChars 0) - default font usage and big enough strings when small average font loss precision can be accumulated in significant mismatch. Do you think we can ignore this?

#231 - 07/29/2015 02:05 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> But the main issue on my guess is wrong calculation(or not enough precision) of the FontManager.getFontWidth(). I think it should be placed in double value, not int.

4GL (in windows) uses GetTextMetricsA GDI function to retrieve the font metrics. This returns a TEXTMETRICS structure, in which the font width tmAveCharWidth is represented as a LONG - so we can't switch to double, as long as 4GL uses int values. This is valid for text metrics, too - no double values are used.

> The approach when we sum the width of all characters and divide to the number of them is not correct(or does not fit our needs) I think. It can be mathematically correct from "average" definition but can be useless in our case, when the correct value can be (Min + Max)/2. At least we need to change the precision for average value I guess. The difference between 5 and 6 for example is too big to ignore, need to have fractional values between 5 and 6.

If you look in FontManager.getFontWidth, you will see that the GuiDriver.getFontWidth is called ONLY AND ONLY if there is no legacy metrics captured in the src/font-metrics.xml file - this file contains all font metrics, as captured on windev01. If a font does not have its metrics recorded in this file (or in font-metrics-ext.xml, we can't guarantee the results will be as on the legacy systems. So, GuiDriver.getFontWidth should be used only in very-specific cases, and never when computing legacy metrics for widgets; for legacy metrics, the FontManager.getFontWidth needs to be used.

> > And again: we need to match the native dimension in pixels as given by the widget's font (which may or may not be specified), as the character dimension is dependent on the DEFAULT-FONT currently in use.

> > I see, my complication is the attempt to adjust a bit for rare cases (font -1 && config().widthChars 0) - default font usage and big enough strings when small average font loss precision can be accumulated in significant mismatch. Do you think we can ignore this?

I don't understand this one: to me, it looks like in this case there is an implicit font and the width is not set explicitly, so the width needs to be computed either via the frame's font or the default font, if the frame doesn't have an explicit font. Do you have a test to demonstrate this case?

#232 - 07/29/2015 02:26 PM - Eugenie Lyzenko

> I don't understand this one: to me, it looks like in this case there is an implicit font and the width is not set explicitly, so the width needs to be computed either via the frame's font or the default font, if the frame doesn't have an explicit font. Do you have a test to demonstrate this case?

Consider button testcases, one example is the button like "Enable on/off". The button text consumes almost all button surface and it is not very good.

OK. If we usually do not compute average font width as you say using the metrics given from the system - there is nothing to do and adjust here. Moreover even if sometimes this complication can give better results, there can be cases when it will not work and give bad picture. Well, I clean the code and upload the final version soon.

#233 - 07/29/2015 02:47 PM - Eugenie Lyzenko

Task branch 2546b for review updated to revision 10913.

Scaling clean up completed.

#234 - 07/29/2015 03:10 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Consider button testcases, one example is the button like "Enable on/off". The button text consumes almost all button surface and it is not very good.

This should not be needed, as we already scale the font so that it matches (to some "Best of" extent) the legacy metrics. But if you want a string to match some legacy metrics, you need to use GuiDriver.drawStringScaled, although this will not provide a very good picture. More, you need to have already captured the legacy metrics for the strings. For this, we have the src/tools/get-text-metrics.p program which needs to be ran on windev01, and accepts as input the following files:

- a file with a font list, one per line, as src/tools/font-list.txt
- a file with a text list, one per line, for which the metrics are required, as in src/tools/text-list.txt. To populate this file, copy and paste the content of the conversion generated ui_strings.txt file.

This will create a text-metrics.xml file which you will need to add to your classpath, when running the programs. Just copying it manually to the src/ folder of the generated java sources and re-building works for now.

#235 - 07/29/2015 07:34 PM - Eugenie Lyzenko

...This will create a text-metrics.xml file which you will need to add to your classpath, when running the programs. Just copying it manually to the src/ folder of the generated java sources and re-building works for now.

OK. While this can help I would suggest suspend the further implementation because now we have the scaling we can use to make proper widget layout. May be we can return to this later and without reference to the particular widget because the issue can be more generic than we can imagine now. Despite of the modern systems WYSIWYG declaration the real life is different and 100% matching is actually unreachable. I mean we need to define the point where we can stop(especially in a case of a lot of work to do until known deadline).

So I suggest to test and commit 2546b to shift to the selection-list widget.

#236 - 07/30/2015 05:47 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546b for review updated to revision 10913.

What I meant is to remove completely the scalingW and scalingH calculations and implement pixel-based width/height... ComboBoxGuiImpl.width() should look like:

```
public double width()
{
   // update widget width if neccessary
   if (config.widthChars != 0.0)
   {
      return config.widthChars;
   }

   // get font to use in combo-box
   int font = gf.resolveFontNum();

   int wFont = FontManager.getFontWidth(window, font, gd);
   // determine the native width in pixels and convert it to character units
   int nwidth = fmtLength * wfont + 28;

   return screen().coordinates().widthFromNative(nwidth);
}
```

And something similar for height().

#237 - 07/30/2015 08:16 AM - Eugenie Lyzenko

What I do not understand here is the difference between your offering and what we currently have in 10913? And why it is better if there is no difference?

#238 - 07/30/2015 08:47 AM - Greg Shah

I'll let Constantin answer the note 237.

> Despite of the modern systems WYSIWYG declaration the real life is different and 100% matching is actually unreachable. I mean we need to define the point where we can stop(especially in a case of a lot of work to do until known deadline).

> So I suggest to test and commit 2546b to shift to the selection-list widget.

We have accepted that the font rendering may be slightly different in our system. But it must be very close, such that the user is not negatively affected by the difference. This difference must only be in the text itself.

The layout of the widgets MUST be 100% exactly the same as in the 4GL. They must be identical on a pixel-by-pixel basis. Getting these metrics exactly right is critical. Now is the time to finish this properly.

#239 - 07/30/2015 10:36 AM - Eugenie Lyzenko

Rebased task branch 2546b from P2J trunk revision 10902, new branch revision is 10914.

#240 - 07/30/2015 11:32 AM - Eugenie Lyzenko

> We have accepted that the font rendering may be slightly different in our system. But it must be very close, such that the user is not negatively affected by the difference. This difference must only be in the text itself.

Agree. This is what to be done unconditionally.

> The layout of the widgets MUST be 100% exactly the same as in the 4GL. They must be identical on a pixel-by-pixel basis. Getting these metrics exactly right is critical. Now is the time to finish this properly.

OK. That's why I have introduced separate scalings for width and height. We can calculate these values only once, no need to do this every time the width() or height() methods are calling because the proportion between current font and default one is not changing. This is done to include some fine tuning without performance impact.

The fine tune I've removed in 10913 serves the same goal - to be pixel-by-pixel compatible with 4GL. But if you think it does overcomplicate the code - it can be removed.

#241 - 07/30/2015 01:43 PM - Constantin Asofiei

Eugenie, your scaling approach is still faulty; easiest way to prove is to use an explicit font with a small size and a DEFAULT-FONT with a large size, as in:

```
            <node class="container" name="font-table">
              <node class="string" name="default-font">
                <node-attribute name="value" value="Tahoma, size 28"/>
              </node>
              <node class="string" name="font10">
                <node-attribute name="value" value="Courier New, size 4"/>
              </node>
```

and in progress.ini:

```
DefaultFont=Tahoma, size=28
font10=Courier New, size 4
```

and this test:

```
define var t1 as char font 10 format "x(10)" view-as combo-box list-items "wwwwwwwww","a","b","c","d".
define var t2 as char font 10 format "x(10)" view-as combo-box list-items "iiiiiiiii","a","b","c","d".
define var t3 as char font 10 format "x(10)" view-as combo-box list-items "MMMMMMMMM","a","b","c","d".
define var t4 as char font 10 format "x(10)" view-as combo-box list-items "WWWWWWWWW","a","b","c","d".

FORM t1 skip t2 skip t3 skip t4 WITH FRAME f1 SIZE 40 BY 40.
MESSAGE t1:WIDTH-PIXELS IN FRAME f1 t1:height-pixels in frame f1.
MESSAGE t2:WIDTH-PIXELS IN FRAME f1 t1:height-pixels in frame f1.
MESSAGE t3:WIDTH-PIXELS IN FRAME f1 t1:height-pixels in frame f1.
MESSAGE t4:WIDTH-PIXELS IN FRAME f1 t1:height-pixels in frame f1.

ON 'a' ANYWHERE
DO:
   t1:FONT IN FRAME f1 = 8.
   RETURN.
END.
update t1 t2 t3 t4 with frame f1 no-labels.
```

The pixel width/height in 4GL is 58/14 and in P2J is 107/9.

More, in 4GL the widget's font can be changed at any time (see the trigger) and its dimensions are re-computed immediately.

So, the scaling code needs to be replaced with what I noted above. Is not required to re-comput the width/height each time, only when the font changes.

Side note: the font indexes need to be consecutive, so if you are missing font8/font9, please add some bogus entries in both directory.xml and progress.ini.

Greg: 2546b can be released as is, as the remaining changes should be only in ComboBoxGuiImpl and its base class.

#242 - 07/30/2015 01:56 PM - Greg Shah

Eugenie: please rebase 2546b from 10903 in the trunk. Then put that version into regression testing. If it passes testing without needed any changes, then you may go ahead and merge to trunk immediately.

If changes are needed, we will get you a review quickly so you can get into testing again. The intention is for you to get this checked in before you leave for vacation, if possible. Of course, if not possible then it is not possible.

Create a new 2546c task branch for the scaling improvements. Please make those your priority over selection list.

#243 - 07/30/2015 02:50 PM - Eugenie Lyzenko

> Eugenie: please rebase 2546b from 10903 in the trunk. Then put that version into regression testing. If it passes testing without needed any changes, then you may go ahead and merge to trunk immediately.

OK.

Rebased task branch 2546b from P2J trunk revision 10903, new branch revision is 10915.

> If changes are needed, we will get you a review quickly so you can get into testing again. The intention is for you to get this checked in before you leave for vacation, if possible. Of course, if not possible then it is not possible.

OK. Starting the regression tests.

> Create a new 2546c task branch for the scaling improvements. Please make those your priority over selection list.

Understood.

#244 - 07/30/2015 03:13 PM - Eugenie Lyzenko

Looks like I need another rebase from 10904 and restart testing. Starting.

#245 - 07/30/2015 03:31 PM - Eugenie Lyzenko

Rebased task branch 2546b from P2J trunk revision 10904, new branch revision is 10916.

#246 - 07/31/2015 05:36 AM - Eugenie Lyzenko

2546b passed the regression testing. The results - 2546b_10916_16cd2a2_20150731_evl.zip. Starting to merge into main trunk.

#247 - 07/31/2015 05:49 AM - Eugenie Lyzenko

The branch 2546b has been merged and committed to trunk as revision 10905. Archived.

#248 - 07/31/2015 05:56 AM - Eugenie Lyzenko

Created task branch 2546c from P2J trunk revision 10905.

#249 - 07/31/2015 08:06 AM - Constantin Asofiei

Eugenie, in case you need to determine if the widget's dimension was set explicitly, there are the fixedWidth and fixedHeight fields in the BaseConfig class.

#250 - 07/31/2015 04:23 PM - Eugenie Lyzenko

Rebased task branch 2546c from P2J trunk revision 10906, new branch revision is 10906.

#251 - 07/31/2015 08:05 PM - Eugenie Lyzenko

After some investigations especially for widgets without explicit width/height set I have found the character based value can be fractional, can be 3.63 or 0.21. So I think the main question is to understand how 4GL decides what will be the pixels or characters width/height in this case? Is the character based values derived from pixels based? Or character values are the "master"?

Constantin, do you have any inside in this area, share please. If we properly understand the 4GL algorithm - we can 100% reproduce it in our P2J project. Is there any general approach the 4GL uses to set pixel width of the window, frame, and so on...?

Then we need do image what is changing if we specify the size explicitly.

This is the plan for scaling issue resolution.

#252 - 08/12/2015 07:12 PM - Eugenie Lyzenko

Rebased task branch 2546c from P2J trunk revision 10919, new branch revision is 10919.

#253 - 08/13/2015 07:40 AM - Eugenie Lyzenko

Rebased task branch 2546c from P2J trunk revision 10920, new branch revision is 10920.

#254 - 08/13/2015 09:20 PM - Eugenie Lyzenko

Task branch 2546c for review updated to revision 10921.

The idea behind is as I know the base for size measurement in GUI is the pixels. Then we compute size in chars. The testing shows this behavior in 4GL(at least because we have fractional size in character units depending on font size). So I conclude we need to adjust character sizes from pixel ones, not vise versa. So if we have correct pixel coords and correct frame/window size chars/pixels mapping - we have will have correct size for combo-box.

The update has changes to be able to get the row height from combo-box for drop-down and computing the height in chars accordingly. Continue working. If there are some notes/objections for this time - let me know.

#255 - 08/14/2015 04:16 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546c for review updated to revision 10921.

The approach looks good - this is what I wanted to see, the actual pixel-based formula used in the widget's height/width APIs. I guess the scalingW, scalingH and all other scaling-related code can be removed, correct?

Also, please do something like this in the widget's height/width APIs: check if config.fixedWidth or config.fixedHeight is set, and if so, return immediately, as we have an explicitly set width or height:

```
    if (config.fixedHeight)
    {
       return config.heightChars;
    }
```

> So I conclude we need to adjust character sizes from pixel ones, not vise versa. So if we have correct pixel coords and correct frame/window size chars/pixels mapping - we have will have correct size for combo-box.

Exactly, this is the approach.

#256 - 08/14/2015 10:05 AM - Eugenie Lyzenko

Rebased task branch 2546c from P2J trunk revision 10921, new branch revision is 10922.

#257 - 08/14/2015 08:49 PM - Eugenie Lyzenko

Task branch 2546c for review updated to revision 10923.

This is the candidate. The entry field now also uses the combo-box size as the base. The ComboBoxGuiImpl.width() must be simplified because we do width calculation in calcLimit() method. Need to perform some additional testing, clean up and remove commented out code. And yes, we do not need the old scaling code anymore.

#258 - 08/15/2015 07:01 AM - Eugenie Lyzenko

Rebased task branch 2546c from P2J trunk revision 10923, new branch revision is 10925.

#259 - 08/17/2015 11:13 AM - Eugenie Lyzenko

Task branch 2546c for review updated to revision 10926.

This is the final cleanup/adjustment candidate. Please review and if there are no objection I would like to insert this update into current testing plan. No more scaling code inside.

#260 - 08/17/2015 01:22 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Task branch 2546c for review updated to revision 10926.
>
> This is the final cleanup/adjustment candidate. Please review and if there are no objection I would like to insert this update into current testing plan. No more scaling code inside.

I'm OK with the code. You can put it into testing.

BTW, I assume that the tests I've posted in note 241 and 130 are OK now, correct?

#261 - 08/17/2015 01:34 PM - Eugenie Lyzenko

> BTW, I assume that the tests I've posted in note 241 and 130 are OK now, correct?

Yes, the widget size in pixels are the same in 4GL and P2J.

There is a painting issue with #241 test(very small font case) but I guess this is related to mouse move repaint generation approach generates to much painting when it is not necessary.

#262 - 08/18/2015 11:53 AM - Eugenie Lyzenko

The main part for 2546c has been completed without regressions. Waiting for CTRL-C one to be done.

#263 - 08/18/2015 01:43 PM - Eugenie Lyzenko

The testing completed, no regressions. The results file is 2546c_10926_16cd2a2_20150818_evl.zip.

So the 2546c is ready to be merged into trunk. Waiting for confirmation.

#264 - 08/18/2015 02:03 PM - Greg Shah

Please merge 2546c to trunk.

#265 - 08/18/2015 02:13 PM - Eugenie Lyzenko

The 2646c has been merged into trunk as revision 10924. Then archived.

#266 - 08/18/2015 02:43 PM - Greg Shah

If I understand correctly, all work defined for this task is complete. Correct?

#267 - 08/18/2015 03:09 PM - Eugenie Lyzenko

Greg Shah wrote:

> If I understand correctly, all work defined for this task is complete. Correct?

Yes, I've finished all I've planned to do for this task.

#268 - 08/18/2015 03:47 PM - Greg Shah

- Status changed from WIP to Closed

#269 - 09/08/2015 02:03 PM - Constantin Asofiei

Eugenie, isn't the combo's drop-down supposed to close when a item is chosen via mouse click? Because currently it doesn't.

#270 - 09/08/2015 02:18 PM - Eugenie Lyzenko

Constantin Asofiei wrote:

> Eugenie, isn't the combo's drop-down supposed to close when a item is chosen via mouse click?

Yes, the drop-down should be closed on mouse selection by click.

> Because currently it doesn't.

In all my tests it does. So provide the testcase with wrong behavior.

#271 - 09/08/2015 02:30 PM - Constantin Asofiei

Eugenie Lyzenko wrote:

> Constantin Asofiei wrote:
>
>> Eugenie, isn't the combo's drop-down supposed to close when a item is chosen via mouse click?
>
>
>> Yes, the drop-down should be closed on mouse selection by click.
>
>> Because currently it doesn't.
>
>
>> In all my tests it does. So provide the testcase with wrong behavior.

I was testing in 1811q, looks like this is a regression in this branch. I'll fix it.

#272 - 11/16/2016 12:13 PM - Greg Shah

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

#273 - 12/01/2020 07:16 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

> 3. The visible rows calculation is now reflects the 4GL approach. The INNER-LINES attribute is not changing but the visible rows adjusted to the current list size. If INNER-LINES > list size, the visible rows become equal to list size. Moreover in GUI it can not be less than 3 lines, in ChUI there is no such adjustment.

I can't duplicate the minimum 3 inner lines in GUI using OE rev 11.6.3.

Is something I'm missing? I'm using a test like this:

```
def var ch as char.

form ch view-as combo-box
      inner-lines 4
      list-items "a", "b"
      with frame f1 side-labels.

update ch with frame f1.
```

And the dropdown shows only the 2 specified list-items. FWD shows a blank line after them.

Files

cbb_p2j_sample_0_gui.jpg	31.2 KB	04/22/2015	Eugenie Lyzenko
cbb_test0_4GL_0_gui.jpg	44.2 KB	04/22/2015	Eugenie Lyzenko
evl_upd20150422b.zip	7.58 KB	04/23/2015	Eugenie Lyzenko
cbb_p2j_sample_1_gui.jpg	35.7 KB	04/23/2015	Eugenie Lyzenko
cbb_test9_1_4GL_0_gui.jpg	57.8 KB	04/23/2015	Eugenie Lyzenko
cbb_p2j_sample_2_gui.jpg	43.2 KB	04/28/2015	Eugenie Lyzenko
cbb_p2j_sample_0428_0_gui.jpg	53.6 KB	04/29/2015	Eugenie Lyzenko
cbb_p2j_sample_0428_1_gui.jpg	52.2 KB	04/29/2015	Eugenie Lyzenko
cbb_p2j_sample_0428_2_gui.jpg	47.9 KB	04/29/2015	Eugenie Lyzenko
evl_upd20150428a.zip	42 KB	04/29/2015	Eugenie Lyzenko
evl_upd20150501b.zip	42.8 KB	05/02/2015	Eugenie Lyzenko
evl_upd20150506a.zip	52.8 KB	05/07/2015	Eugenie Lyzenko
evl_upd20150507a.zip	52.4 KB	05/07/2015	Eugenie Lyzenko
evl_upd20150507b.zip	52.6 KB	05/07/2015	Eugenie Lyzenko
evl_upd20150507c.zip	52.7 KB	05/08/2015	Eugenie Lyzenko
evl_upd20150508a.zip	53 KB	05/08/2015	Eugenie Lyzenko
cbb_p2j_sample_0508_0_distorted_gui.jpg	40.3 KB	05/08/2015	Eugenie Lyzenko
evl_upd20150508b.zip	53.1 KB	05/08/2015	Eugenie Lyzenko
evl_upd20150508c.zip	53.1 KB	05/08/2015	Eugenie Lyzenko
evl_upd20150508d.zip	53.3 KB	05/08/2015	Eugenie Lyzenko
evl_upd20150512a.zip	12.9 KB	05/13/2015	Eugenie Lyzenko
cbb_p2j_sample_0512_0_gui.jpg	57.8 KB	05/13/2015	Eugenie Lyzenko
evl_upd20150514a.zip	36 KB	05/14/2015	Eugenie Lyzenko
evl_upd20150514b.zip	39.6 KB	05/14/2015	Eugenie Lyzenko
evl_upd20150515a.zip	40 KB	05/16/2015	Eugenie Lyzenko
evl_upd20150518a.zip	178 KB	05/18/2015	Eugenie Lyzenko
evl_upd20150518b.zip	179 KB	05/18/2015	Eugenie Lyzenko
evl_upd20150519a.zip	179 KB	05/19/2015	Eugenie Lyzenko
evl_upd20150526a.zip	181 KB	05/27/2015	Eugenie Lyzenko
evl_upd20150527a.zip	181 KB	05/27/2015	Eugenie Lyzenko
evl_upd20150529a.zip	187 KB	05/30/2015	Eugenie Lyzenko
evl_upd20150601a.zip	195 KB	06/02/2015	Eugenie Lyzenko
cbb_p2j_sample_0601_0_gui.jpg	40.4 KB	06/02/2015	Eugenie Lyzenko
cbb_p2j_sample_0601_1_gui.jpg	54 KB	06/02/2015	Eugenie Lyzenko
evl_upd20150602a.zip	196 KB	06/03/2015	Eugenie Lyzenko
evl_upd20150609a.zip	208 KB	06/10/2015	Eugenie Lyzenko
evl_upd20150610a.zip	210 KB	06/11/2015	Eugenie Lyzenko
cbb_p2j_simple_mode_0610_0_gui.jpg	55 KB	06/11/2015	Eugenie Lyzenko
cbb_p2j_simple_mode_0610_1_gui.jpg	60.3 KB	06/11/2015	Eugenie Lyzenko
cbb_p2j_simple_mode_0610_2_gui.jpg	62.1 KB	06/11/2015	Eugenie Lyzenko
evl_upd20150611a.zip	213 KB	06/12/2015	Eugenie Lyzenko
evl_upd20150613a.zip	226 KB	06/13/2015	Eugenie Lyzenko
cbb_p2j_drop_down_0613_0_gui.jpg	51.3 KB	06/13/2015	Eugenie Lyzenko
cbb_p2j_drop_down_0613_1_gui.jpg	43.2 KB	06/13/2015	Eugenie Lyzenko

evl_upd20150614a.zip	227 KB	06/14/2015	Eugenie Lyzenko
evl_upd20150615a.zip	227 KB	06/15/2015	Eugenie Lyzenko
evl_upd20150616a.zip	226 KB	06/17/2015	Eugenie Lyzenko
evl_upd20150617a.zip	15.1 KB	06/17/2015	Eugenie Lyzenko
evl_upd20150617b.zip	16.4 KB	06/17/2015	Eugenie Lyzenko
evl_upd20150618a.zip	210 KB	06/19/2015	Eugenie Lyzenko
evl_upd20150619a.zip	213 KB	06/19/2015	Eugenie Lyzenko
evl_upd20150622a.zip	210 KB	06/23/2015	Eugenie Lyzenko
evl_upd20150626a.zip	221 KB	06/27/2015	Eugenie Lyzenko
evl_upd20150628a.zip	230 KB	06/28/2015	Eugenie Lyzenko
evl_upd20150630a.zip	222 KB	06/30/2015	Eugenie Lyzenko
evl_upd20150701a.zip	235 KB	07/02/2015	Eugenie Lyzenko
cbb_p2j_sample_0701_0_gui.jpg	65.5 KB	07/02/2015	Eugenie Lyzenko
cbb_p2j_sample_0701_1_gui.jpg	50.1 KB	07/02/2015	Eugenie Lyzenko
cbb_p2j_sample_0701_2_gui.jpg	56.6 KB	07/02/2015	Eugenie Lyzenko
cbb_test15_4GL_0_gui.jpg	42.6 KB	07/09/2015	Eugenie Lyzenko
cbb_test15_p2j_0_gui.jpg	30.7 KB	07/09/2015	Eugenie Lyzenko
cbb_test15_1_4GL_0_gui.jpg	45.1 KB	07/10/2015	Eugenie Lyzenko
combo_tahoma_14.png	7.91 KB	07/17/2015	Constantin Asofiei
cbb_test14_p2j_20150722_gui.jpg	38.9 KB	07/21/2015	Eugenie Lyzenko
movie-ratings-demo.jpg	98.3 KB	07/22/2015	Eugenie Lyzenko