

User Interface - Bug #2923

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

./combo_box/combo_box9_1.p web combobox width is not enough to display its content with paddings

12/07/2015 10:54 AM - Sergey Ivanovskiy

Status:	New	Start date:	12/07/2015
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 12/07/2015 10:57 AM - Sergey Ivanovskiy

- File *combobox_content_padding.png* added

Please look at the attached picture, the web combobox width doesn't match its content with appropriate paddings.

#2 - 04/29/2016 01:26 AM - Sergey Ivanovskiy

Found that the legacy font width used to draw the combobox's entry field and its drop down list by `ComboBoxGuiImpl` is less than the font width we can get from the web driver.

The second issue is that if we create `ScrollableSelectionListGuiImpl` object from the combobox parent window, then all created sub widgets have this parent window as an owner but not the overlay window. It makes impossible to use their `GuiFontResolver` if we are in the batch drawing of the owner overlay window. I don't know if the window owner of `ScrollableSelectionListGuiImpl` should be the combobox overlay window but not the parent window of the combobox itself. Is it correct that the overlay window isn't an owner of widgets placed on it?

#3 - 04/29/2016 01:29 AM - Sergey Ivanovskiy

Don't know to whom address this question, who can help to answer it?

#4 - 04/29/2016 02:29 AM - Constantin Asofiei

Sergey, questions for the legacy font width ... is less than the font width we can get from the web driver.:

1. do you have the `simple/server/fonts` folder and the `custom-fonts` settings in `directory.xml` specified?
2. have you captured the legacy metrics for these strings, via the `get-text-metrics.p` program and placed the proper `text-metrics.xml` file in your project?

The second issue is that if we create `ScrollableSelectionListGuiImpl` object from the combobox parent window, then all created sub widgets have this parent window as an owner but not the overlay window.

Hynek, IIRC you made some changes here.

It makes impossible to use their `GuiFontResolver` if we are in the batch drawing of the owner overlay window

Do you mean that the window's environment (like font-table and so on) is not inherited from the parent window, in case of overlay windows (like the drop-down)?

#5 - 04/29/2016 02:50 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey, questions for the legacy font width ... is less than the font width we can get from the web driver.:

1. do you have the simple/server/fonts folder and the custom-fonts settings in directory.xml specified?
2. have you captured the legacy metrics for these strings, via the get-text-metrics.p program and placed the proper text-metrics.xml file in your project?

Thanks, no I didn't do this.

The second issue is that if we create ScrollableSelectionListGuiImpl object from the combobox parent window, then all created sub widgets have this parent window as an owner but not the overlay window.

Hynek, IIRC you made some changes here.

It makes impossible to use their GuiFontResolver if we are in the batch drawing of the owner overlay window

Do you mean that the window's environment (like font-table and so on) is not inherited from the parent window, in case of overlay windows (like the drop-down)?

I mean that ScrollableSelectionListGuiImpl if it is opened in the overlay window has GuiFontResolver that doesn't have the overlay window as its window instance.

#6 - 04/29/2016 02:53 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

It makes impossible to use their GuiFontResolver if we are in the batch drawing of the owner overlay window.

Please detail what problem you encountered here. You mean that, as the widgets have as parent window the overlay's parent (and not the overlay), you can access/select this window? Something else?

#7 - 04/29/2016 03:12 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

It makes impossible to use their GuiFontResolver if we are in the batch drawing of the owner overlay window.

Please detail what problem you encountered here. You mean that, as the widgets have as parent window the overlay's parent (and not the overlay), you can access/select this window? Something else?

This problem can appear if we select the parent window from the overlay window draw. But it is an artificial problem. For an example, if this code FontManager is invoked from the overlay window draw

```
int width = details.legacyMaxWidth;

if (width == -1)
{
    if (LOG.isLoggable(Level.WARNING))
    {
        final String msg = "Font [%d] with key [%s] has no legacy max width.";
        LOG.log(Level.WARNING, String.format(msg, font, key));
    }

    // let the driver measure the font
    GuiDriver gd = (GuiDriver) driver;
    gd.selectWindow(UiUtils.getWidgetIdAsInt(window));
    width = gd.getFontMaxWidth(details);
    gd.releaseWindow();
}
```

#8 - 04/29/2016 04:02 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Found that the legacy font width used to draw the combobox's entry field and its drop down list by `ComboBoxGuiImpl` is less than the font width we can get from the web driver.

The second issue is that if we create `ScrollableSelectionListGuiImpl` object from the combobox parent window, then all created sub widgets have this parent window as an owner but not the overlay window. It makes impossible to use their `GuiFontResolver` if we are in the batch drawing of the owner overlay window. I don't know if the window owner of `ScrollableSelectionListGuiImpl` should be the combobox overlay window but not the parent window of the combobox itself. Is it correct that the overlay window isn't an owner of widgets placed on it?

When you say owner, do you mean the return value of `Widget.parent()`?

#9 - 04/29/2016 05:15 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

Found that the legacy font width used to draw the combobox's entry field and its drop down list by `ComboBoxGuiImpl` is less than the font width we can get from the web driver.

The second issue is that if we create `ScrollableSelectionListGuiImpl` object from the combobox parent window, then all created sub widgets have this parent window as an owner but not the overlay window. It makes impossible to use their `GuiFontResolver` if we are in the batch drawing of the owner overlay window. I don't know if the window owner of `ScrollableSelectionListGuiImpl` should be the combobox overlay window but not the parent window of the combobox itself. Is it correct that the overlay window isn't an owner of widgets placed on it?

When you say owner, do you mean the return value of `Widget.parent()`?

Let \rightarrow denote parent relation and `ClassName` represent the instance of this `ClassName` class. This relation is between objects that are instances of their classes. The `ScrollableSelectionListGuiImpl` parent hierarchy looks normal. Please fix if it isn't true `ScrollableSelectionListGuiImpl` has this parent \rightarrow `ViewPort` \rightarrow `BorderedPanelGuiImpl` \rightarrow `ScrollPaneGuiImpl` \rightarrow `DropDownGuiImpl` \rightarrow `BorderPanelGuiImpl` \rightarrow `OverlayWindow` \rightarrow `combobox.topLevelWindow()`.

But the top level window hierarchy is confusing, because `ScrollableSelectionListGuiImpl.topLevelWindow() != DropDownGuiImpl.topLevelWindow()` and `ScrollableSelectionListGuiImpl.topLevelWindow() == combobox.topLevelWindow()` and `DropDownGuiImpl.topLevelWindow() == OverlayWindow`. I am talking about that the `GuiFontResolver` created by `ScrollableSelectionListGuiImpl` has a `window` field that is set to `combobox.topLevelWindow()`. Is it correct?

I am planning to fix it like it is implemented for ScrollPaneGuiImpl. The below code is from DropDown

```
@SuppressWarnings("unchecked")
public DropDown(ComboBox<O> parent)
{
    super("");
    this.comboBox = parent;
    list = screen().getFactory().createScrollableList(parent, parent.model());

    // enable scrolling, ensure there is no extra border or spacing
    scrollpane = screen().getFactory().createScrollPane(list, this::topLevelWindow);
    super.add(scrollpane);
}
```

and the below code is from GuiWidgetFactory

```
/**
 * Create UI-specific {@link ScrollableList} implementation.
 *
 * @param parent
 *         Parent {@link ComboBox} widget.
 * @param model
 *         Scrollable list model.
 *
 * @return instance of UI-specific implementation of {@link ScrollableList}.
 */
@Override
public ScrollableSelectionListGuiImpl createScrollableList(ComboBox<GuiOutputManager> parent,
                                                          ListModel<String> model)
{
    return new ScrollableSelectionListGuiImpl(parent, model);
}
.....
/**
 * Create GUI-specific {@link ScrollPane} instance.
 *
 * @param widget
 *         Widget which will be scrolled.
 * @param wnd
 *         TopLevelWindow supplier
 *
 * @return instance of GUI-specific implementation of {@link ScrollPaneGuiImpl}.
 */
@Override
public ScrollPaneGuiImpl createScrollPane(ScrollableWidget<GuiOutputManager> widget,
                                         Supplier<TopLevelWindow<GuiOutputManager>> wnd)
{
    return new ScrollPaneGuiImpl(widget, wnd);
}
```

#11 - 04/29/2016 05:55 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

Found that the legacy font width used to draw the combobox's entry field and its drop down list by `ComboBoxGuiImpl` is less than the font width we can get from the web driver.

The second issue is that if we create `ScrollableSelectionListGuiImpl` object from the combobox parent window, then all created sub widgets have this parent window as an owner but not the overlay window. It makes impossible to use their `GuiFontResolver` if we are in the batch drawing of the owner overlay window. I don't know if the window owner of `ScrollableSelectionListGuiImpl` should be the combobox overlay window but not the parent window of the combobox itself. Is it correct that the overlay window isn't an owner of widgets placed on it?

When you say owner, do you mean the return value of `Widget.parent()`?

Let -> denote parent relation and `ClassName` represent the instance of this `ClassName` class. This relation is between objects that are instances of their classes. The `ScrollableSelectionListGuiImpl` parent hierarchy looks normal. Please fix if it isn't true `ScrollableSelectionListGuiImpl` has this parent -> `ViewPort` -> `BorderedPanelGuiImpl` -> `ScrollPaneGuiImpl` -> `DropDownGuiImpl` -> `BorderPanelGuiImpl` -> `OverlayWindow` -> `combobox.topLevelWindow()`.

But the top level window hierarchy is confusing, because `ScrollableSelectionListGuiImpl.topLevelWindow() != DropDownGuiImpl.topLevelWindow()` and `ScrollableSelectionListGuiImpl.topLevelWindow() == combobox.topLevelWindow()` and `DropDownGuiImpl.topLevelWindow() == OverlayWindow`. I am talking about that the `GuiFontResolver` created by `ScrollableSelectionListGuiImpl` has a window field that is set to `combobox.topLevelWindow()`. Is it correct?

`ScrollableSelectionListGuiImpl.topLevelWindow() == combobox.topLevelWindow()` is certainly not correct. It should really follow the standard child-parent relationship protocol.

#12 - 04/29/2016 05:55 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

I am planning to fix it like it is implemented for ScrollPaneGuiImpl.

Sounds good.

#13 - 04/29/2016 06:01 AM - Hynek Cihlar

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

I am planning to fix it like it is implemented for ScrollPaneGuiImpl.

Sounds good.

Actually why does createScrollPane() need the Supplier to get the top-level window? If it receives the correct parent widget, then getting the top-level window is just a matter of calling Widget.topLevelWindow(). This could similarly work for the ScrollableSelectionListGuiImpl case.

ScrollableSelectionListGuiImpl is used in two cases, one in overlay window as combo drop-down and the other in SIMPLE COMBO-BOX where the list is really part of the COMBO-BOX hierarchy (no overlay window involved). So I believe GuiWidgetFactory.createScrollableList() should just receive the correct parent widget.

#14 - 04/29/2016 07:32 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

I am planning to fix it like it is implemented for ScrollPaneGuiImpl.

Sounds good.

Actually why does createScrollPane() need the Supplier to get the top-level window? If it receives the correct parent widget, then getting the top-level window is just a matter of calling Widget.topLevelWindow(). This could similarly work for the ScrollableSelectionListGuiImpl case.

ScrollableSelectionListGuiImpl is used in two cases, one in overlay window as combo drop-down and the other in SIMPLE COMBO-BOX where the list is really part of the COMBO-BOX hierarchy (no overlay window involved). So I believe GuiWidgetFactory.createScrollableList() should just receive the correct parent widget.

Yes, agree. It doesn't need. GuiFontResolver can use topLevelWindow().

#15 - 04/29/2016 11:22 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey, questions for the legacy font width ... is less than the font width we can get from the web driver.:

1. do you have the simple/server/fonts folder and the custom-fonts settings in directory.xml specified?
2. have you captured the legacy metrics for these strings, via the get-text-metrics.p program and placed the proper text-metrics.xml file in your project?

It seems that the combobox with the given format string doesn't use text metrics to calculate its width. Its width is calculated by its chars width multiplying the given font width.

The font width is taken from the font-metrics.xml and is known as a legacy width. If we substitute the font legacy width with the driver font width than the combobox width is changed. Is font legacy width required for the compatibility with native 4GL application? Please explain if the font legacy can't be substituted by the driver's returned font width, then how we can change the combobox width?

```
define variable Combo as character format "x(13)" initial "Item number 0"
  view-as combo-box list-items
    "Item number 0",
    "Item number 1",
    "Item number 2",
    "Item number 3",
    "Item number 4",
    "Item number 5",
    "Item number 6",
    "Item number 7",
    "Item number 8",
    "Item number 9",
    "Item number A",
    "Item number B",
    "Item number C",
    "Item number D",
    "Item number E",
    "Item number F"
  inner-lines 9 sort.
```


#16 - 04/29/2016 11:35 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Is font legacy width required for the compatibility with native 4GL application?

Yes

Please explain if the font legacy can't be substituted by the driver's returned font width, then how we can change the combobox width?

The combobox width must remain computed using the legacy font metrics; is needed because the widget's width/height (whatever that is) must be 100% compatible with how GUI 4GL computes it; otherwise, we will not be able to perform the layout correctly, as the widgets might no longer fit into the specified frame, on the same line, etc.

If the line text for the combo's drop-down can't match 100% to how 4GL draws it, then that will be a limitation. When drawing text, currently we are doing a "best of" scenario to try to fit that text into the required space - this is what `GuiDriver.drawStringScaled` does. Try changing the `ScrollableSelectionListGuiImpl.drawItem` to use `gd.drawStringScaled` instead of `gd.drawString`. But for this to work, you will need to have the legacy text metrics captured in `text-metrics.xml`.

#17 - 04/29/2016 11:55 AM - Sergey Ivanovskiy

Thanks, is it an automate process to collect all strings used by application in `lines-list.txt` and then to run `./fonts/system.metrics/get-text-metrics.p?`
What is the content of `./fonts/system.metrics/font-list.txt`?

```
MS Sans Serif,10,true  
Courier New,5  
Segoe UI,6  
Microsoft Sans Serif,7  
system,8  
FixedSys,9
```

Found this document `testcases/uast/fonts/font_details.odt`. Do you know additional documents for reading?

#18 - 04/29/2016 12:04 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Thanks, is it an automate process to collect all strings used by application in lines-list.txt and then to run
./fonts/system.metrics/get-text-metrics.p?

Yes, they are collected during conversion, in a ui_strings.txt file (just rename it to lines-list.txt or enter ui_strings.txt at the prompt). Check if the combo's content (i.e. lines with Item number.. is in this file - if not, add it by hand (for now).

What is the content of ./fonts/system.metrics/font-list.txt?

Is a list of fonts (with their size and style) for which you want to capture the text metrics. Just leave it like that.

Found this document testcases/uast/fonts/font_details.odt. Do you know additional documents for reading?

I know I posted some details to some task, but I can't find it right now.

Just run the get-text-metrics.p on windev01, via prowin32 -p get-text-metrics.p from the command line. Enter the details and let it run - it might take a few minutes. After that, get the generated text-metrics.xml to your system.

#19 - 05/02/2016 09:16 AM - Sergey Ivanovskiy

- File *WebClientBoxAlignedText.png* added

- File *2923_1.txt* added

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

Thanks, is it an automate process to collect all strings used by application in lines-list.txt and then to run
./fonts/system.metrics/get-text-metrics.p?

Yes, they are collected during conversion, in a ui_strings.txt file (just rename it to lines-list.txt or enter ui_strings.txt at the prompt). Check if the combo's content (i.e. lines with Item number.. is in this file - if not, add it by hand (for now).

What is the content of ./fonts/system.metrics/font-list.txt?

Is a list of fonts (with their size and style) for which you want to capture the text metrics. Just leave it like that.

Found this document testcases/uast/fonts/font_details.odt. Do you know additional documents for reading?

I know I posted some details to some task, but I can't find it right now.

Just run the get-text-metrics.p on windev01, via prowinc32 -p get-text-metrics.p from the command line. Enter the details and let it run - it might take a few minutes. After that, get the generated text-metrics.xml to your system.

Found another solution to place text entries for web combo boxes centered within their boxes. It doesn't change the Swing client and doesn't change combo boxes dimensions. Please look at the picture. Committed revision 11018 into task branch 3084a is ready for the review.

#20 - 05/02/2016 09:40 AM - Sergey Ivanovskiy

- File 2923_2.txt added

Please review the committed revision 11019. It fixed drawTextBoxAligned.

#21 - 05/05/2016 12:14 PM - Sergey Ivanovskiy

Greg, Constantin please review the proposed fix (rev 11028) branch 3084a

#22 - 05/11/2016 04:23 PM - Greg Shah

Code Review Task Branch 3084a Revision 11025 (was 11019)

One thing that seems wrong is the direct reference to ComboBoxGuiImpl in ScrollableSelectionListGuiImpl. That class should be generic. It is used in selection-list too, right?

I can't properly evaluate the Combo-Box changes. Eugenie: please review 11025 in 3084a.

#23 - 05/11/2016 05:23 PM - Eugenie Lyzenko

Greg Shah wrote:

I can't properly evaluate the Combo-Box changes. Eugenie: please review 11025 in 3084a.

Need some clarification.

Now the entry field(DROP-DOWN mode) text is centered in both X and Y directions within the entry field rectangle, right?

If so the text position might have differences from 4GL for small text, large text, text with length not exactly matching the entry field visible area, so this will cause to regressions.

Was this update tested in GUI mode to compare with 4GL screens? I think there will be regressions in GUI.

#24 - 05/11/2016 05:34 PM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

Greg Shah wrote:

I can't properly evaluate the Combo-Box changes. Eugenie: please review 11025 in 3084a.

Need some clarification.

Now the entry field(DROP-DOWN mode) text is centered in both X and Y directions within the entry field rectangle, right?

If so the text position might have differences from 4GL for small text, large text, text with length not exactly matching the entry field visible area, so this will cause to regressions.

Was this update tested in GUI mode to compare with 4GL screens? I think there will be regressions in GUI.

No, it adds additional parameters to the gui client, I need to deliver additional parameters to the client side, but the GUI use case doesn't use them, otherwise the web client uses these new parameters to center the items in the exceptional cases if these items are near the right bound or over it

```
public void drawString(String text, int boxWidth, int boxHeight, int x, int y,  
                       int xOffset, int yOffset)
```

#25 - 05/11/2016 06:13 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

No, it adds additional parameters to the gui client, I need to deliver additional parameters to the client side, but the GUI use case doesn't use them, otherwise the web client uses these new parameters to center the items in the exceptional cases if these items are near the right bound or over it
[...]

I see. In swing mode the change does not work and combo-box works as before.

But in web client mode this will center the string, right? And this is the moment to check if the P2J code will works as 4GL. Why do you think the text must be centered in this case? Can you provide the 4GL pictures for these cases?

#26 - 05/12/2016 12:47 AM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

Sergey Ivanovskiy wrote:

No, it adds additional parameters to the gui client, I need to deliver additional parameters to the client side, but the GUI use case doesn't use them, otherwise the web client uses these new parameters to center the items in the exceptional cases if these items are near the right bound or over it
[...]

I see. In swing mode the change does not work and combo-box works as before.

But in web client mode this will center the string, right? And this is the moment to check if the P2J code will works as 4GL. Why do you think the text must be centered in this case? Can you provide the 4GL pictures for these cases?

The pictures should be different, the changes are here to work around the issue. Please read note 16 to note 19. We can't change the box dimensions, but should squeeze text in the provided boxes (combo box's entry and drop down list). I don't know why the web client displays items differently from the gui and 4GL cases if all dimensions for all widgets and font sizes are the same. I tried the way proposed by Constantin for both clients and got very divergent results for these clients and decided to center the items in the exceptional cases if these items are near the right bound or over it.

#27 - 05/12/2016 01:06 AM - Sergey Ivanovskiy

Have found that the current solution isn't acceptable because different items are placed with different offsets. Thus I should apply text scaling for the web client only. If my understanding of this issue is correct, then the issue appears if the items maximal char length equals to the size specified by the format string.

#28 - 05/12/2016 01:15 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

Have found that the current solution isn't acceptable because different items are placed with different offsets. Thus I should apply text scaling for the web client only. If my understanding of this issue is correct, then the issue appears if the items maximal char length equals to the size specified by the format string.

Planning to test different cases with fill-in and combo-boxes that are filled according to their maximal formats and then to compare 4GL and Web client pictures.

#29 - 05/12/2016 01:32 AM - Sergey Ivanovskiy

Removed note 29. The test ./combo_box/combo_box9_1.p is correct.

#30 - 05/12/2016 05:56 AM - Sergey Ivanovskiy

I have 3 pictures for ./combo_box/combo_box9_1.p : 4GL from windev01, P2J swing and web clients. Eugenie, could you help how to compare these images by pixels? Visually 4GL and P2J swing are the same, but the web client has a font with the antialiased effect.

#31 - 05/12/2016 06:01 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

I have 3 pictures for ./combo_box/combo_box9_1.p : 4GL from windev01, P2J swing and web clients. Eugenie, could you help how to compare these images by pixels? Visually 4GL and P2J swing are the same, but the web client has a font with the antialiased effect.

The antialiasing issue for the web client is still open - I didn't find a way to turn off antialiasing only for the P2J page, only globally. OTOH, I tested this only on Firefox, at that time.

#32 - 05/12/2016 06:10 AM - Sergey Ivanovskiy

- File 4GL_windev01_combobox_fill_in.png added

- File P2J_web_combobox_fill_in.png added

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

I have 3 pictures for ./combo_box/combo_box9_1.p : 4GL from windev01, P2J swing and web clients. Eugenie, could you help how to compare these images by pixels? Visually 4GL and P2J swing are the same, but the web client has a font with the antialiased effect.

The antialiasing issue for the web client is still open - I didn't find a way to turn off antialiasing only for the P2J page, only globally. OTOH, I tested this only on Firefox, at that time.

It seems that the web item text occupies the larger space than the same text in 4GL windev01 or in P2J swing client snapshots due to antialiasing.

#33 - 05/12/2016 06:11 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

It seems that the web item text occupies the larger space than the same text in 4GL windev01 or in P2J swing client snapshots due to antialiasing.

Yes, this is a side effect of antialiasing, and another reason why we use legacy metrics instead of relying on the driver to compute the text metrics.

#34 - 05/12/2016 06:30 AM - Sergey Ivanovskiy

- File combo_box9_1.p added

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

It seems that the web item text occupies the larger space than the same text in 4GL windev01 or in P2J swing client snapshots due to antialiasing.

Yes, this is a side effect of antialiasing, and another reason why we use legacy metrics instead of relying on the driver to compute the text metrics.

It seems that [#2923](#) has this root cause because the compared fonts looks the same, dimensions looks equal and antialiased letters are composed by 3 lines. The combo box test is modified to check the fill-in field.

#35 - 05/12/2016 07:41 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

It seems that the web item text occupies the larger space than the same text in 4GL windev01 or in P2J swing client snapshots due to antialiasing.

Yes, this is a side effect of antialiasing, and another reason why we use legacy metrics instead of relying on the driver to compute the text metrics.

The 4GL in your pictures shows left edge alignment, while P2J web client picture is X centered. And this is not anti-alias issue. To see more clear - try to use the items shorter than in combo_box9_1.p, for example "ltn 1"...

#36 - 05/12/2016 08:21 AM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

It seems that the web item text occupies the larger space than the same text in 4GL windev01 or in P2J swing client snapshots due to antialiasing.

Yes, this is a side effect of antialiasing, and another reason why we use legacy metrics instead of relying on the driver to compute the text metrics.

The 4GL in your pictures shows left edge alignment, while P2J web client picture is X centered. And this is not anti-alias issue. To see more clear - try to use the items shorter than in combo_box9_1.p, for example "ltn 1"...

Eugenie, I think you are confused by my 3084a code, but my changes aren't correct. The attached picture for the P2J web doesn't display items centered.

#37 - 05/12/2016 08:27 AM - Sergey Ivanovskiy

Planning to roll back combo box changes and to postpone this task. It is due to the font's antialiased effect. Greg, do you agree?

#38 - 05/12/2016 11:51 AM - Sergey Ivanovskiy

- File 2923_3.txt added

I have checked that scaling text items doesn't help. It produces artifacts of double letters and looks bad. But this function added to GuiDriver can be usefull

```
/**
 * Draw a string using the provided text box and text offset position.
 *
 * @param text
 *       text string to draw.
 * @param x
 *       The left text position.
 * @param y
 *       The top text position.
 * @param boxWidth
 *       The text box width, which needs to be matched when drawing.
 * @param boxHeight
 *       The text box height, which needs to be matched when drawing.
 * @param xBoxOffset
 *       The horizontal box position.
 * @param yBoxOffset
 *       The vertical box position.
 */
public void drawString(String text, int x, int y, int boxWidth, int boxHeight,
                      int xBoxOffset, int yBoxOffset);
```

#39 - 05/12/2016 12:28 PM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

I have checked that scaling text items doesn't help. It produces artifacts of double letters and looks bad. But this function added to GuiDriver can be usefull
[...]

Found a mistake in my code that was the cause of double letters (located in p2j.screen.js). It has been fixed and the result is acceptable. Working on the code cleanup and the review note 22.

#40 - 05/12/2016 09:54 PM - Sergey Ivanovskiy

- File 2923_4.txt added

Greg Shah wrote:

Code Review Task Branch 3084a Revision 11025 (was 11019)

One thing that seems wrong is the direct reference to `ComboBoxGuiImpl` in `ScrollableSelectionListGuiImpl`. That class should be generic. It is used in `selection-list` too, right?

I can't properly evaluate the Combo-Box changes. Eugenie: please review 11025 in 3084a.

Committed revision 11030 (3084a) fixed this issue by scaling the combobox items and tried to fix the direct reference to `ComboBoxGuiImpl` in `ScrollableSelectionListGuiImpl`, but there were difficulties due to `WidgetFactory`. Attached the diff of changes.

#41 - 05/17/2016 02:44 PM - Sergey Ivanovskiy

The proposed solution doesn't work because it applies the stretch to the content uniformly to all combo boxes. There are cases, for an example, `./demo/demo_widgets.p`, that have combo boxes large enough to squeeze all items with free right spaces. Thus these two cases are very different. Please look at

1) have free spaces at the right side

```
DEF VAR chcb AS CHAR VIEW-AS COMBO-BOX LIST-ITEMS "item1", "item2", "item3", "item4", "item5", "item6", "item7", "item8", "item9", "item10" INNER-LINES 5.
```

2) doesn't have free spaces at the right side, the format string and inner-lines are given

```
define variable Combo as character format "x(13)" initial "Item number 0"  
  view-as combo-box list-items  
    "Item number 0",  
    "Item number 1",  
    "Item number 2",  
    "Item number 3",  
    "Item number 4",  
    "Item number 5",  
    "Item number 6",  
    "Item number 7",  
    "Item number 8",  
    "Item number 9",  
    "Item number A",  
    "Item number B",  
    "Item number C",  
    "Item number D",  
    "Item number E",  
    "Item number F"  
  inner-lines 9 sort.
```

#42 - 05/17/2016 03:19 PM - Sergey Ivanovskiy

- File `combo_box_test.png` added

Committed revision 11043 fixes this issue. The screen looks suitable.

#43 - 06/01/2016 03:16 PM - Constantin Asofiei

Sergey, for this task, have you checked combo items with length less than the combo's width?

Also, Eugenie: please review the comb-related changes in 3084a.

#44 - 06/01/2016 03:18 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey, for this task, have you checked combo items with length less than the combo's width?

Also, Eugenie: please review the comb-related changes in 3084a.

Yes, checked, there is a picture from note 42 in this task.

#45 - 06/01/2016 04:29 PM - Eugenie Lyzenko

Constantin Asofiei wrote:

Also, Eugenie: please review the comb-related changes in 3084a.

I'm OK with changes. My only concern is modified string drawing primitive for entry field drawing in `ComboBoxGuiImpl.java`:

```
...
    gd.drawString(display, xt, yt, w, h, orig.x, orig.y);
}
else
{
    gd.setColor(lightShadow);
    gd.drawString(display, xt + 1, yt + 1, w, h, orig.x, orig.y);
    gd.setColor(darkShadow);
    gd.drawString(display, xt, yt, w, h, orig.x, orig.y);
}
...
```

It is important to test the string is drawing in correct location with pixel-to-pixel match precision. Was this check performed?

#46 - 06/01/2016 06:45 PM - Sergey Ivanovskiy

Thanks, planning to compare items locations for the entry field and combobox list pixels with Progress 4GL. This method uses scaling and it leads to that the items positions can be equal, but their lengths can be different and can't be the same as they are displayed by Progress 4GL.

#47 - 06/02/2016 09:03 AM - Sergey Ivanovskiy

- File *combo_box9_1.p* added

- File *4GL_combo_box_zoomed_4.xcf* added

- File *web_combo_box_zoomed_4.xcf* added

Eugenie Lyzenko wrote:

Constantin Asofiei wrote:

Also, Eugenie: please review the comb-related changes in 3084a.

I'm OK with changes. My only concern is modified string drawing primitive for entry field drawing in *ComboBoxGuiImpl.java*:

[...]

It is important to test the string is drawing in correct location with pixel-to-pixel match precision. Was this check performed?

The right paddings for web combobox entries are different from the Progress 4GL rendering. Also the window layout for this example is not the same as the original one.

The items that are not scaled have larger right paddings than scaled items paddings.

#48 - 06/02/2016 09:14 AM - Sergey Ivanovskiy

For this example the following issue can be observed. The P2J combobox entry item is always selected and its background is colored if this widget has the input focus. Otherwise the original Progress 4GL combobox entry item can be unselected if its items list is displayed and a different item from this list is selected and the widget is focused.

#49 - 06/02/2016 09:38 AM - Sergey Ivanovskiy

Greg, do you have any objection if I will create 2923a branch for this combobox issue and remove combobox changes from 3084a branch, because this branch contains CLIPBOARD changes and has new template parameter, *graphicsCashed*? 3084a is almost ready waiting Safari support.

#50 - 06/02/2016 10:00 AM - Greg Shah

For this example the following issue can be observed. The P2J combobox entry item is always selected and its background is colored if this widget has the input focus. Otherwise the original Progress 4GL combobox entry item can be unselected if its items list is displayed and a different item from this list is selected and the widget is focused.

Please review the combo-box issues in [#2677](#). If this is not represented, create a new sub-task.

#51 - 06/02/2016 10:00 AM - Greg Shah

Greg, do you have any objection if I will create 2923a branch for this combobox issue and remove combobox changes from 3084a branch, because this branch contains CLIPBOARD changes and has new template parameter, graphicsCached? 3084a is almost ready waiting Safari support.

No objection.

#52 - 06/02/2016 02:14 PM - Sergey Ivanovskiy

New branch 2923a is created.

#53 - 06/02/2016 02:15 PM - Sergey Ivanovskiy

- File 2923a_1.txt added

I think this way can be successful to fix this issue.

#54 - 06/17/2016 01:50 AM - Sergey Ivanovskiy

The note 47 describes that if an item of the combo box widget is scaled depending on the item native width and the required box width, then items are placed with different offsets from the owner's border. It is due to the font anti-aliasing and scaling. But if these transformations are applied to the ideal logical space, then the all items must have the same offset.

The 2923 case has this root case that the font legacy width is less than the web font width. Planning to apply the common scaling factor to the all combo box's items. The legacy font width is taken into account by the required box width, since the required box width is the legacy font width multiplied by the format number. Thus the chars width is required by the Web client to calculate the common scaling factor.

#55 - 06/17/2016 08:33 AM - Sergey Ivanovskiy

Tested these "mmm...", "MMM..." and "WWW..." 13-length strings with Swing and Web GUI clients

```
define variable Combo as character format "x(13)" initial "Item 0"  
  view-as combo-box list-items  
    "Item 0",  
    "Item 1",  
    "Item number 2",  
    "Item number 3",  
    "Item number 4",  
    "Item number 5",  
    "Item number 6",  
    "Item number 7",  
    "Item number 8",
```

```
"Item number 9",  
"Item number A",  
"Item number B",  
"Item number C",  
"Item number D",  
"Item number E",  
"Item number F",  
"mmmmmmmmmmmmmmmm",  
"wwwwwwwwwwwwwwww",  
"MMMMMMMMMMMMMM"  
inner-lines 9 sort.
```

and observed that the Swing and web clients can't display these items fitted to the combo box width. For the web client the difference between the actual text width and the provided box space is almost twice greater than the provided space, thus if we apply the common scaling factor to all items, then these items are displayed disproportional.

The same problem appears for the Swing client too. It seems that the font is not monospaced. Don't know what can be done here if we don't want to use horizontal scrolling for the combo box items.

#56 - 06/17/2016 08:42 AM - Constantin Asofiei

Sergey, first question is: what font are you using for your test in the previous note? If you are using default font (System), then that is not yet mapped to a ttf font in P2J, so you will have different font rendering when compared to 4GL; you need to use an explicit font in the directory's font-table, via the font option, to test these accurately. Also, have you captured text metrics for the test?

Second, before going further into fixing this, and as this is related to the anti-aliasing being ON, please look into different ways we can turn anti-aliasing OFF. Which browsers allow this? Can it be done only for P2J document/web page? Can it be turned off globally, otherwise? If anti-aliasing is turned off, is your scaling issue still an issue, for the web browser?

#57 - 06/17/2016 08:59 AM - Sergey Ivanovskiy

Thanks, I will check it. Tested the example from the note 55 on windev01 and observed the same issue that the items are not fitted the combo box width.

#58 - 06/17/2016 09:08 AM - Sergey Ivanovskiy

- File *combobox_4GL.png* added

#59 - 06/17/2016 09:17 AM - Greg Shah

Please post images of the P2J results for the same case (swing and web).

#60 - 06/18/2016 06:52 AM - Sergey Ivanovskiy

- File *combobox_P2J_Swing.png* added

- File *combobox_P2J_Web.png* added

- File `directory.xml` added

In my environment the `directory.xml` provides custom ttf fonts in `testcases/simple/server/fonts` and `text-metrics.xml` is build following the note 18, `font-metrics.xml` is located with `text-metrics.xml`. The web client uses 8pt "MS Sans Serif" to draw combo box items, but the Swing client uses `java.awt.Font[family=Microsoft Sans Serif,name=Microsoft Sans Serif,style=plain,size=11]`.

#61 - 06/18/2016 07:35 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey, first question is: what font are you using for your test in the previous note? If you are using default font (System), then that is not yet mapped to a ttf font in P2J, so you will have different font rendering when compared to 4GL; you need to use an explicit font in the directory's font-table, via the font option, to test these accurately. Also, have you captured text metrics for the test?

Constantin, please explain the font option in the directory's font-table.

#62 - 06/18/2016 07:54 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Constantin, please explain the font option in the directory's font-table.

To use the FONT option, the syntax is something like this:

```
display ch font 8 with frame f1.
```

where 8 is a the font8 entry in the directory's font-table.

On windev01, you need to edit a `progress.ini` file and update the `[fonts]` section so that the i.e. font8 matches what is in your directory.

Search on windev01 to find a 'base' `progress.ini` and copy it in one of your folders; after that, to start the program so that it uses `progress.ini`, use this command:

```
prowin32 -ininame progress.ini -basekey INI -p your-program.p
```

#63 - 06/18/2016 08:08 AM - Sergey Ivanovskiy

Thank you, the font option directs Progress 4GL to use the given font 8 to draw ch, defined in "[fonts]" section of progress.ini file, doesn't it? What font is used by Progress 4GL default-font or default-fixed-font if we don't provide the FONT option?

I can't explain why the Swing client uses this font `java.awt.Font[family=Microsoft Sans Serif,name=Microsoft Sans Serif,style=plain,size=11]`, otherwise the Web client uses this one 8pt "MS Sans Serif" and it seems according to FontTable module

```
// default fonts
standardFontTable.put("default-font", "MS Sans Serif, size=8");
standardFontTable.put("default-fixed-font", "Courier New, size=8");
```

#64 - 06/18/2016 02:58 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Thank you, the font option directs Progress 4GL to use the given font 8 to draw ch, defined in "[fonts]" section of progress.ini file, doesn't it?

Correct.

What font is used by Progress 4GL default-font or default-fixed-font if we don't provide the FONT option?

It uses the default-font or default-fixed-font (in case of numeric/date FILL-INS, as I recall).

I can't explain why the Swing client uses this font `java.awt.Font[family=Microsoft Sans Serif,name=Microsoft Sans Serif,style=plain,size=11]`, otherwise the Web client uses this one 8pt "MS Sans Serif" and it seems according to FontTable module [...]

The FontTable config is used as a fallback if there is no setting in the directory font-table. So, what is your font-table defined in the directory? Note that a default-font set to the empty string will default to System font, as I recall. And web/swing don't have a mapping for this.

Anyway, please try to test with an explicit font set at the combo-box and see how 4GL/Swing behave.

#65 - 06/19/2016 04:09 AM - Sergey Ivanovskiy

Constantin, I have no additional font settings except custom-font settings in the directory.xml (Please look at the attached file in the note 60). What are the target settings? Please see readDefaultFonts() of FontTable and FontTable's static initializations

```
// load the standard font table
// default fonts
standardFontTable.put("default-font", "MS Sans Serif, size=8");
standardFontTable.put("default-fixed-font", "Courier New, size=8");
```

it follows that the default font is "MS Sans Serif, size=8".

#66 - 06/19/2016 04:52 AM - Sergey Ivanovskiy

We have only this font fonts/micros.ttf and it is the Microsoft Sans Serif font, thus MS Sans Serif is mapped to Microsoft Sans Serif.

#67 - 06/19/2016 05:23 AM - Sergey Ivanovskiy

Ok, the next transformation from deriveFont of SwingGuiDriver gives java.awt.Font[family=Microsoft Sans Serif,name=Microsoft Sans Serif,style=plain,size=11] the point size 11.

#68 - 06/19/2016 09:03 AM - Constantin Asofie

As I recall, MS SansSerif is a bitmap font, and we don't support bitmap fonts - thus you can see differences in the final resolved font. When testing, don't use bitmap fonts, use true-type fonts like Tahoma, Segoe, Courier New, etc

This is a sample on how to configure the font-table in the directory - add the node on the same level as custom-fonts, under server/default:

```
<node class="container" name="font-table">
  <node class="string" name="default-font">
    <node-attribute name="value" value=""/>
  </node>
  <node class="string" name="default-fixed-font">
    <node-attribute name="value" value="Courier New, size=8"/>
  </node>
  <node class="string" name="font0">
    <node-attribute name="value" value="FixedSys"/>
  </node>
  <node class="string" name="font1">
    <node-attribute name="value" value="System"/>
  </node>
  <node class="string" name="font2">
    <node-attribute name="value" value="Courier New, size 8"/>
  </node>
  <node class="string" name="font3">
    <node-attribute name="value" value="Courier New, size=8"/>
  </node>
  <node class="string" name="font4">
    <node-attribute name="value" value="Microsoft Sans Serif, size=8"/>
  </node>
  <node class="string" name="font5">
    <node-attribute name="value" value="Microsoft Sans Serif, size=10"/>
  </node>
  <node class="string" name="font6">
    <node-attribute name="value" value="Microsoft Sans Serif, size=8, bold"/>
  </node>
  <node class="string" name="font7">
    <node-attribute name="value" value="System"/>
  </node>
```

```
</node>
<node class="string" name="font8">
  <node-attribute name="value" value="Microsoft Sans Serif, size=8"/>
</node>
<node class="string" name="font9">
  <node-attribute name="value" value="Microsoft Sans Serif, size 8"/>
</node>
<node class="string" name="font10">
  <node-attribute name="value" value="Microsoft Sans Serif, size 8"/>
</node>
<node class="string" name="font11">
  <node-attribute name="value" value="Microsoft Sans Serif, size 8"/>
</node>
<node class="string" name="font12">
  <node-attribute name="value" value="Courier New, size 8"/>
</node>
<node class="string" name="font13">
  <node-attribute name="value" value="system"/>
</node>
<node class="string" name="font14">
  <node-attribute name="value" value="system"/>
</node>
<node class="string" name="font15">
  <node-attribute name="value" value="system"/>
</node>
<node class="string" name="font16">
  <node-attribute name="value" value="Courier New, size=8"/>
</node>
</node>
```

Also, when testing progress, make sure the explicit font used at the widget is the same in progress.ini as in the directory.

#69 - 06/19/2016 09:14 AM - Constantin Asofiei

There was a problem in previous note, the node name was font-1table instead of font-table.

#70 - 06/19/2016 04:06 PM - Sergey Ivanovskiy

Ok, thank you.

#71 - 06/20/2016 01:43 AM - Sergey Ivanovskiy

I didn't succeeded to build a personal environment on windev01. The first I copied progress.ini from C:\dlc91d\bin\ to c:\users\sbi and tried to start prowin32 -ininame progress.ini -basekey INI -p combo_box9_1.p. These messages appeared one after the other

```
Read version 3 from convmap.cp. Expected version is 5. (6107)
Unable to read header information from file convmap.cp. (6061)
```

According to *Chapter 4: Maintaining User Environments* of dp4gl.pdf tried to rename progress.ini to follow another progress.ini search use case and it did not help. The same messages appeared. Also tried to use C:\dlc91d\bin\83dbutils\progress.ini file (copied it to my personal folder and promsgs file) with this result

```
msgOpen: unable to open message file: PROMSGS
```

Did I miss something to run tests using the provided environment progress.ini? Do you have a valid ini progress file to test programs against the different font settings?

#72 - 06/20/2016 07:33 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

I didn't succeeded to build a personal environment on windev01. The first I copied progress.ini from C:\dlc91d\bin\ to c:\users\sbi ...

That file is for 4GL 9.1D, not 10.2B. Sorry about that, I should have been more explicit; please use the file from d:\oe102b\bin\.

#73 - 06/20/2016 05:36 PM - Sergey Ivanovskiy

- File P2J_Web_Swing_AntiAliasing_Off.png added

Constantin Asofiei wrote:

Sergey, first question is: what font are you using for your test in the previous note? If you are using default font (System), then that is not yet mapped to a ttf font in P2J, so you will have different font rendering when compared to 4GL; you need to use an explicit font in the directory's font-table, via the font option, to test these accurately. Also, have you captured text metrics for the test?

Second, before going further into fixing this, and as this is related to the anti-aliasing being ON, please look into different ways we can turn anti-aliasing OFF. Which browsers allow this? Can it be done only for P2J document/web page? Can it be turned off globally, otherwise? If anti-aliasing is turned off, is your scaling issue still an issue, for the web browser?

If the font anti-aliasing is disabled globally for Ubuntu using ~/.fonts.conf, then P2J Swing and P2J Web looks very similar. Please look at this picture, where the Swing client is located over the Web client. I don't know how to proceed with this task. The paddings looks the same. If we line the P2J

Web client combo box's items, then the extent of this line meets the P2J Swing client combo box's items.

#74 - 06/20/2016 05:38 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

If the font anti-aliasing is disabled globally for Ubuntu using `~/.fonts.conf`, then P2J Swing and P2J Web looks very similar.

Does this work for all browsers? Is this the only way to do it?

Please look at this picture the Swing client is located over the Web client.

How is it compared to the 4GL?

#75 - 06/20/2016 05:44 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

If the font anti-aliasing is disabled globally for Ubuntu using `~/.fonts.conf`, then P2J Swing and P2J Web looks very similar.

Does this work for all browsers? Is this the only way to do it?

Constantin, I am using this task [#2701](#). The task to disable fonts anti-aliasing for browsers is a different issue, isn't it?

#76 - 06/20/2016 05:51 PM - Sergey Ivanovskiy

- File `.fonts.conf` added

If this attached `~/.fonts.conf` is added, then all applications change their fonts look&feel after restart.

#77 - 06/20/2016 05:54 PM - Sergey Ivanovskiy

Also I have tested this <https://github.com/lazyhacker/chromecss> for Chrome and it doesn't work for me.

#78 - 06/20/2016 05:58 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Constantin, I am using this task [#2701](#). The task to disable fonts anti-aliasing for browsers is a different issue, isn't it?

I think is a different task, but we should know the extent on which anti-aliasing disabling is possible or not, for all browsers, before starting hacking our way into making the anti-aliased fonts match the 'disabled anti-aliasing' way. The idea is, if turning off anti-aliasing is possible on a per-page/document, then the original work for this task is not needed. Otherwise, is needed.

#79 - 06/20/2016 06:23 PM - Sergey Ivanovskiy

The web search to explore this way didn't give me a hope that it would be possible uniformly for browsers. It seems that the css way to use `body { -webkit-font-smoothing: none; }` (<https://developer.mozilla.org/en-US/docs/Web/CSS/font-smooth>) doesn't work because according to my Chrome test the font's appearance is changed, but the fonts anti-aliasing is still present. Planning to spend more time to the css way, but I am not sure that it works properly. (<http://blog.lazyhacker.com/2014/03/disabling-anti-aliasing-on-chrome.html>, <https://github.com/h5bp/html5-boilerplate/issues/598>)

#80 - 06/20/2016 06:32 PM - Sergey Ivanovskiy

And <http://usabilitypost.com/2012/11/05/stop-fixing-font-smoothing/> explains the font's smoothing and font's anti-aliasing are different methods.

#81 - 06/21/2016 04:31 AM - Greg Shah

we should know the extent on which anti-aliasing disabling is possible or not, for all browsers, before starting hacking our way into making the anti-aliased fonts match the 'disabled anti-aliasing' way. The idea is, if turning off anti-aliasing is possible on a per-page/document, then the original work for this task is not needed. Otherwise, is needed.

Yes, this is correct.

#82 - 06/21/2016 06:34 AM - Sergey Ivanovskiy

I think it is not possible to disable font smoothing using css. We only can detect if the given web client uses font smoothing to follow this way <http://www.useragentman.com/blog/2009/11/29/how-to-detect-font-smoothing-using-javascript/>, then we can do a horizontal shift to 1 px left depending on the detecting smoothing edge. If font anti-aliasing is enabled, then it should use pixels to smooth. But if sub-pixels are used, then the smoothing area can be more finer.

#83 - 06/21/2016 07:51 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

I think it is not possible to disable font smoothing using css. We only can detect if the given web client uses font smoothing to follow this way <http://www.useragentman.com/blog/2009/11/29/how-to-detect-font-smoothing-using-javascript/>, then we can do a horizontal shift to 1 px left depending on the detecting smoothing edge. If font anti-aliasing is enabled, then it should use pixels to smooth. But if sub-pixels are used, then the smoothing area can be more finer.

Do you think is worth (it makes things easier) disabling the sub-pixels when drawing text? I mean, to determine that we need to shift 1px left?

#84 - 06/21/2016 07:57 AM - Greg Shah

In note 73, you mentioned that if font anti-aliasing is disabled at the Ubuntu OS level, then the issue is not present.

1. Please investigate how anti-aliasing can be disabled (or not) for each browser.
2. Please help us understand if font smoothing also contributes to the issue.
3. Is the issue present in all browsers already?
4. If OS level disabling of anti-aliasing is needed, how can this be done in MacOS and Windows (you've already posted how to do this in Linux)?

#85 - 06/21/2016 08:17 AM - Sergey Ivanovskiy

Greg Shah wrote:

In note 73, you mentioned that if font anti-aliasing is disabled at the Ubuntu OS level, then the issue is not present.

1. Please investigate how anti-aliasing can be disabled (or not) for each browser.

It can depend on the pair of OS and browser. I have only these systems Ubuntu, Fedore, Win 7 32-bit and Mac OS X El Capitan available.

2. Please help us understand if font smoothing also contributes to the issue.

Font anti-aliasing is a font smoothing.

3. Is the issue present in all browsers already?

For Chrome and Firefox on Ubuntu if fonts anti-aliasing is enabled globally, then the issue is present.

4. If OS level disabling of anti-aliasing is needed, how can this be done in MacOS and Windows (you've already posted how to do this in Linux)?

Disabling font anti-aliasing on the OS level results the bad look and feel for other applications. Do you persist to continue this way?

#86 - 06/21/2016 08:26 AM - Sergey Ivanovskiy

Sub-pixels depends on the hardware capabilities.

#87 - 06/21/2016 08:32 AM - Sergey Ivanovskiy

Greg, do you need a report how to disable fonts anti-aliasing for these systems: Ubuntu, Fedore, Win 7 32-bit and Mac OS X El Capitan and available Firefox, Chrome, Safari, IE11?

#88 - 06/21/2016 08:35 AM - Sergey Ivanovskiy

This is a valuable report <http://szafranek.net/works/articles/font-smoothing-explained/>

#89 - 06/21/2016 09:08 AM - Greg Shah

Please help us understand if font smoothing also contributes to the issue.

Font anti-aliasing is a font smoothing.

In note 80 you stated "the font's smoothing and font's anti-aliasing are different methods". Please clarify this. Does disabling the font smoothing disable all anti-aliasing for fonts (and thus "resolve" the issue)?

Is the issue present in all browsers already?

For Chrome and Firefox on Ubuntu if fonts anti-aliasing is enabled globally, then the issue is present.

What about IE and Safari?

If OS level disabling of anti-aliasing is needed, how can this be done in MacOS and Windows (you've already posted how to do this in Linux)?

Disabling font anti-aliasing on the OS level results the bad look and feel for other applications. Do you persist to continue this way?

I understand the limitations. I just want to know what is possible.

I also want you to determine if there is a browser-specific method for disabling anti-aliasing for each browser.

do you need a report how to disable fonts anti-aliasing for these systems: Ubuntu, Fedore, Win 7 32-bit and Mac OS X El Capitan and available Firefox, Chrome, Safari, IE11?

Yes.

#90 - 06/21/2016 01:22 PM - Sergey Ivanovskiy

- File KDE.zip added

I tested how to disable fonts anti-aliasing on Fedore 20 using KDE to change fonts settings globally. The actual configuration file is located `~/.config/fontconfig/fonts.conf` and it is supposed to be a correct location for this file. The same location and the fonts configuration file is valid for Ubuntu 15.10. <http://manpages.ubuntu.com/manpages/wily/man5/fonts-conf.5.html> is for the reference. The situation is more complicated. The fonts anti-aliasing is not an issue for my notebook's 1366x768 screen (window.devicePixelRatio = 1) because the P2J default font doesn't change its straight parts of letters depending on fonts anti-aliasing is enabled or not. Please look at the provided zipped pictures.

#91 - 06/21/2016 01:51 PM - Sergey Ivanovskiy

Greg Shah wrote:

In note 80 you stated "the font's smoothing and font's anti-aliasing are different methods". Please clarify this. Does disabling the font smoothing disable all anti-aliasing for fonts (and thus "resolve" the issue)?

From these sources <https://www.grc.com/ctwhat.htm> , <http://szafranek.net/works/articles/font-smoothing-explained/> and <http://usabilitypost.com/2012/11/05/stop-fixing-font-smoothing/> it follows that fonts anti-aliasing uses the whole pixel to smooth, but the other smoothing method uses sub-pixels. Using font.conf sub-pixels smoothing is controlled by

```
<match target="font">
  <edit mode="assign" name="rgba">
    <const>rgb</const>
  </edit>
</match>
```

the valid values for "rgba" sub-pixels smoothing are

unknown	rgba	0
rgb	rgba	1
bgr	rgba	2
vrgb	rgba	3
vbgr	rgba	4
none	rgba	5

and anti-aliasing is controlled by

```
<match target="font">
  <edit mode="assign" name="hinting">
    <bool>>true</bool>
  </edit>
</match>
<match target="font">
  <edit mode="assign" name="hintstyle">
    <const>hintfull</const>
  </edit>
</match>
<match target="font">
  <edit mode="assign" name="antialias">
    <bool>>false</bool>
  </edit>
</match>
```

but this setting

```
<match target="font">
```



```
<edit mode="assign" name="antialias">
  <bool>false</bool>
</edit>
</match>
```

disables fonts anti-aliasing globally.

#92 - 06/21/2016 01:56 PM - Sergey Ivanovskiy

Watching KDE.zip please don't take into account the Firefox artifacts, the new Firefox version displays items correctly.

#93 - 06/21/2016 02:01 PM - Sergey Ivanovskiy

I think if we estimate how the vertical char '|' is rendered, then it helps to evaluate if the left pixel shift is required or not.

#94 - 06/21/2016 02:44 PM - Greg Shah

I can't tell anything from the firefox png in your KDE zip. From note 92, I guess you mean for us to ignore it completely.

Looking at the screen captures, I don't see a left shift issue. The issue I see is that the text in the web is vertically shifted.

Before we go further, please just get complete answers for the questions in note 84. I want to have a better understanding of anti-aliasing before we decide on this task.

#95 - 06/21/2016 03:38 PM - Sergey Ivanovskiy

Greg Shah wrote:

I can't tell anything from the firefox png in your KDE zip. From note 92, I guess you mean for us to ignore it completely.

No, the Firefox image is a valid image. The image has these artifacts. The first visible and the last visible drop down items are cut at the beginning.

Looking at the screen captures, I don't see a left shift issue. The issue I see is that the text in the web is vertically shifted.

Yes, for this test the horizontal shift doesn't need and it doesn't depend on the fonts anti-aliasing settings. But the different computer Ubuntu 15.10 having this screen 1920x1080 (window.devicePixelRatio = 1) requires the horizontal shift if the fonts anti-aliasing is enabled.

Before we go further, please just get complete answers for the questions in note 84. I want to have a better understanding of anti-aliasing before we decide on this task.

Continue to investigate the note 84 questions.

#96 - 06/22/2016 02:34 AM - Sergey Ivanovskiy

The situation with fonts anti-aliasing is complicated because the fonts transformation takes into account the device screen. For the web client `window.devicePixelRatio == 1` is true for each of the tested environment, but the screen size and the current font size influence the performed fonts transformation. If the screen is 1366x768 and the web embedded font is Microsoft Sans Serif, size=8, then fonts-antialiasing doesn't widen the visual presentation of characters. Otherwise the screen is 1920x1080 with the same font, then fonts-antialiasing widens the visual presentation of characters so the left 1 pixel shift is required.

#97 - 06/22/2016 04:20 AM - Sergey Ivanovskiy

How to change font smoothing in Windows is described by <https://www.winhelp.us/change-font-smoothing-in-windows.html> and <http://www.thewindowsclub.com/disable-font-smoothing-windows>.

#98 - 06/22/2016 06:48 AM - Sergey Ivanovskiy

Some interesting finding for the web client is that if the web embedded font, "Microsoft Sans Serif", is not detected, then its font size isn't transformed by the Canvas API. It means that if we set `ctx.font="8pt 'Microsoft Sans Serif'"`, then `console.debug(ctx.font)` returns the same string `8pt 'Microsoft Sans Serif'`. But if this font is detected and isn't loaded as the embedded font, then `console.debug(ctx.font)` returns this string `11px 'Microsoft Sans Serif'`. We see that the 8pt unit is converted to 11px unit.

#99 - 06/22/2016 08:03 AM - Greg Shah

Interesting result.

On the one hand, we could use unique font names (e.g. "GCD Font 1") for our embedded fonts and always force the loading of embedded fonts, even on platforms where the same font already exists. Would this yield the proper results?

On the other hand, it is crazy to do this when the font is already there. So it would be preferred to just disable this "help" from the browser. This seems like it is more than just anti-aliasing. Perhaps this is some kind of automatic scaling for device resolution.

#100 - 06/22/2016 08:38 AM - Sergey Ivanovskiy

Fonts detecting is a useful feature that tries to solve [#2930](#) the embedded font license issue. The described fonts transformation is normal because the Browser can use its own fonts.

#101 - 06/22/2016 08:57 AM - Sergey Ivanovskiy

I have some troubles with IE11 to disable anti-aliasing first and to detect the web embedded fonts. Please look at <http://stackoverflow.com/questions/761778/forcing-anti-aliasing-using-css-is-this-a-myth>. This statement "You can force font anti-alias in css by using pt instead of px." is true or false, don't know but IE11 uses anti-aliasing for the canvas drawing if the fonts anti-aliasing is disabled globally. Planning to add screen shots, but it works for the Chrome and it seems all other applications and IE11 don't use anti-aliasing for menus.

#102 - 06/22/2016 09:22 AM - Sergey Ivanovskiy

Some interesting finding for the web client is that if the web embedded font, "Microsoft Sans Serif", is not detected, then its font size isn't transformed by the Canvas API. It means that if we set `ctx.font="8pt 'Microsoft Sans Serif'"`, then `console.debug(ctx.font)` returns the same string `8pt 'Microsoft Sans Serif'`.

It can be observed for Ubuntu with the Firefox web client.

But if this font is detected and isn't loaded as the embedded font, then `console.debug(ctx.font)` returns this string `11px "Microsoft Sans Serif"`. We see that the 8pt unit is converted to 11px unit.

It can be observed for Windows 7 32-bit with the IE11 web client or the Google Chrome web client.

#103 - 06/22/2016 09:29 AM - Sergey Ivanovskiy

It seems that the note 98 and 100 is related to the Firefox web client, because the Google Chrome web client on Ubuntu transforms 8pt to 11px too.

#104 - 06/22/2016 09:32 AM - Sergey Ivanovskiy

And this issue with incorrect left paddings for combo box items is observed for the Firefox and it isn't reproduced for the Google Chrome.

#105 - 06/22/2016 09:37 AM - Greg Shah

This is confusing to me. Are you saying that we want the scaling of the 8px to 11px to occur?

Can you show the issue in screen shots that are captured as already zoomed? Please post them directly embedded in the history so we don't have to download and unpack a zip. I still don't know exactly what the problem is for this task.

#106 - 06/22/2016 09:43 AM - Sergey Ivanovskiy

Greg Shah wrote:

This is confusing to me. Are you saying that we want the scaling of the 8px to 11px to occur?

We can observe this transformation for the Chrome on Ubuntu, but not for the Firefox.

Can you show the issue in screen shots that are captured as already zoomed? Please post them directly embedded in the history so we don't have to download and unpack a zip. I still don't know exactly what the problem is for this task.

You can watch https://proj.goldencode.com/redmine/attachments/5062/combobox_P2J_Web.png that displays the Firefox web client on Ubuntu with the issue. I will add 4 times zoomed pictures for the Firefox and for the Google Chrome respectively.

#107 - 06/22/2016 09:59 AM - Sergey Ivanovskiy

- File *Firefox_left_padding_issue.xcf* added
- File *Firefox_lpi_4_times_zoomed.xcf* added
- File *Chrome_to_compare.xcf* added
- File *Chrome_to_compare_4_times_zoomed.xcf* added

Please look at these pictures. Please be patient due to it is very easy to do a mistake if we estimate pictures visually.

#108 - 06/22/2016 10:20 AM - Greg Shah

Please post these as png or jpg.

#109 - 06/22/2016 10:24 AM - Sergey Ivanovskiy

- File deleted (*Chrome_to_compare_4_times_zoomed.xcf*)

#110 - 06/22/2016 10:24 AM - Sergey Ivanovskiy

- File deleted (*Chrome_to_compare.xcf*)

#111 - 06/22/2016 10:24 AM - Sergey Ivanovskiy

- File deleted (*Firefox_lpi_4_times_zoomed.xcf*)

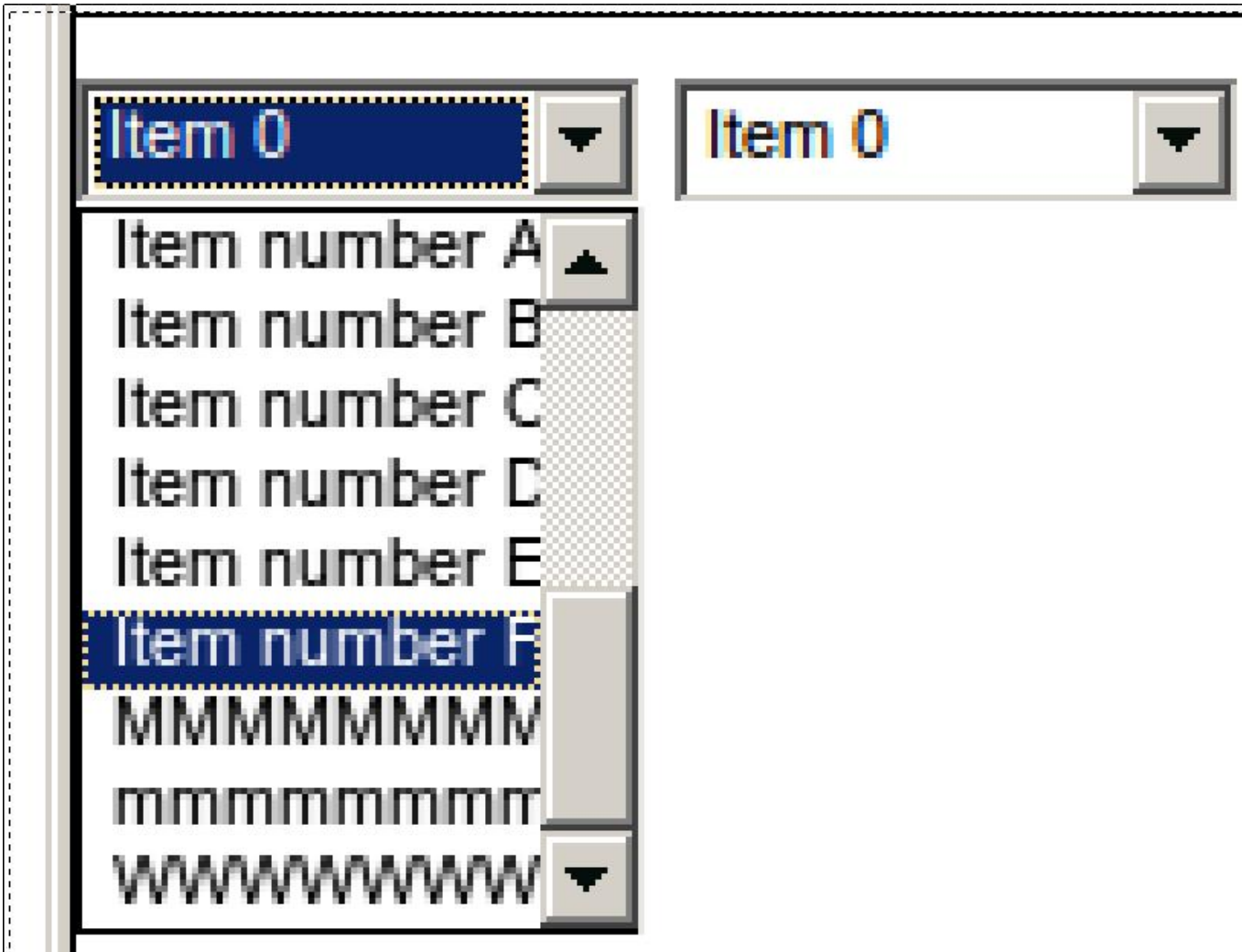
#112 - 06/22/2016 10:24 AM - Sergey Ivanovskiy

- File deleted (*Firefox_left_padding_issue.xcf*)

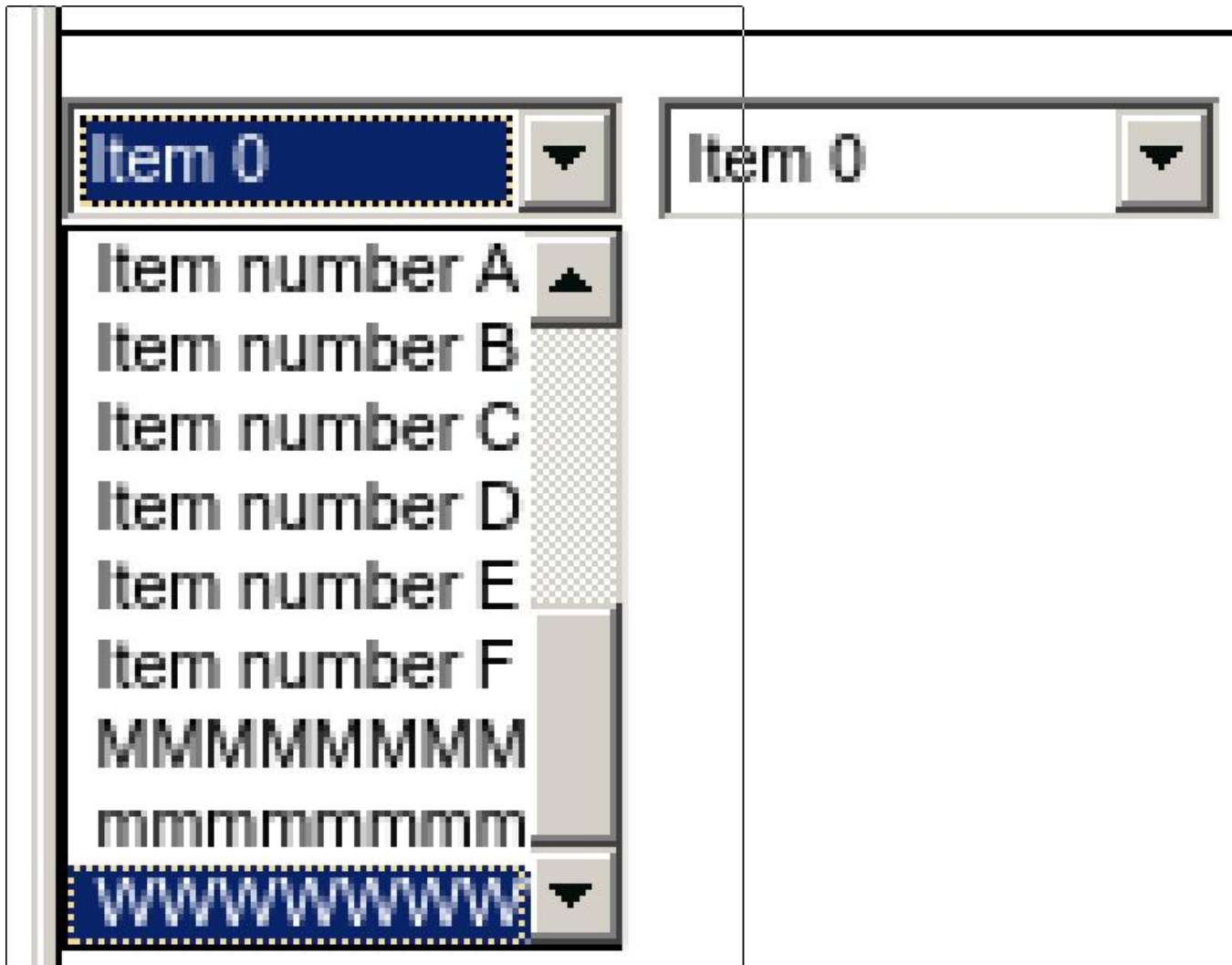
#113 - 06/22/2016 10:28 AM - Sergey Ivanovskiy

- File *Firefox_left_padding_issue.png* added
- File *Firefox_lpi_4_times_zoomed.png* added
- File *Chrome_to_compare.png* added
- File *Chrome_to_compare_4_times_zoomed.png* added

For the Firefox web client



For the Chrome web client



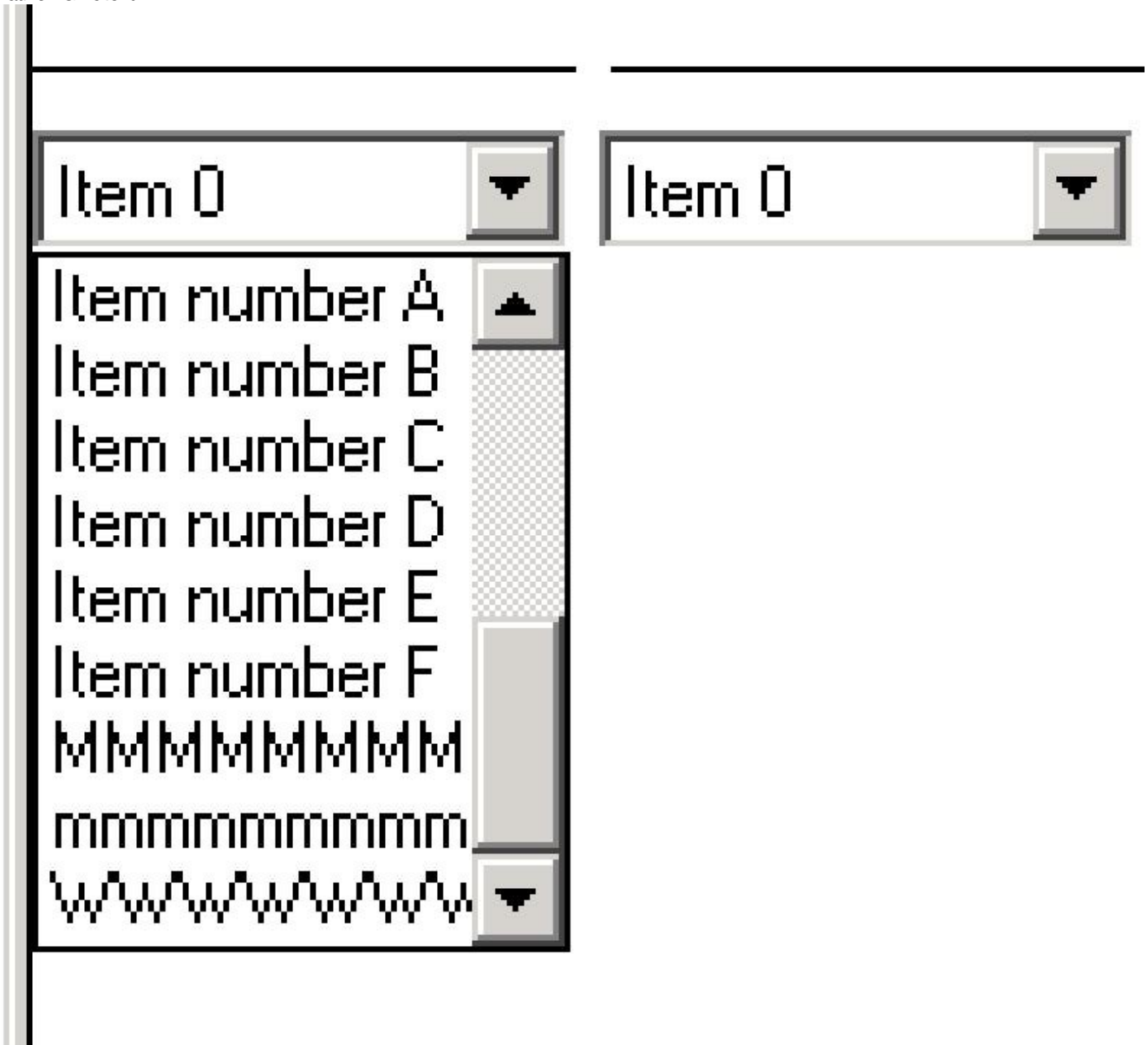
#114 - 06/22/2016 10:38 AM - Greg Shah

Can you post the 4GL version that is also magnified?

#115 - 06/22/2016 10:42 AM - Sergey Ivanovskiy

- File 4GL_combo_box_items_4_times_zoomed.png added

The native 4GL client



#116 - 06/22/2016 10:47 AM - Greg Shah

How does the change from "8pt to 11px" affect this?

#117 - 06/22/2016 10:52 AM - Sergey Ivanovskiy

I don't know but the logs can be confusing if we expect to see "8 pt Microsoft Sans Serif" but observe "11px Microsoft Sans Serif".

#118 - 06/22/2016 10:56 AM - Greg Shah

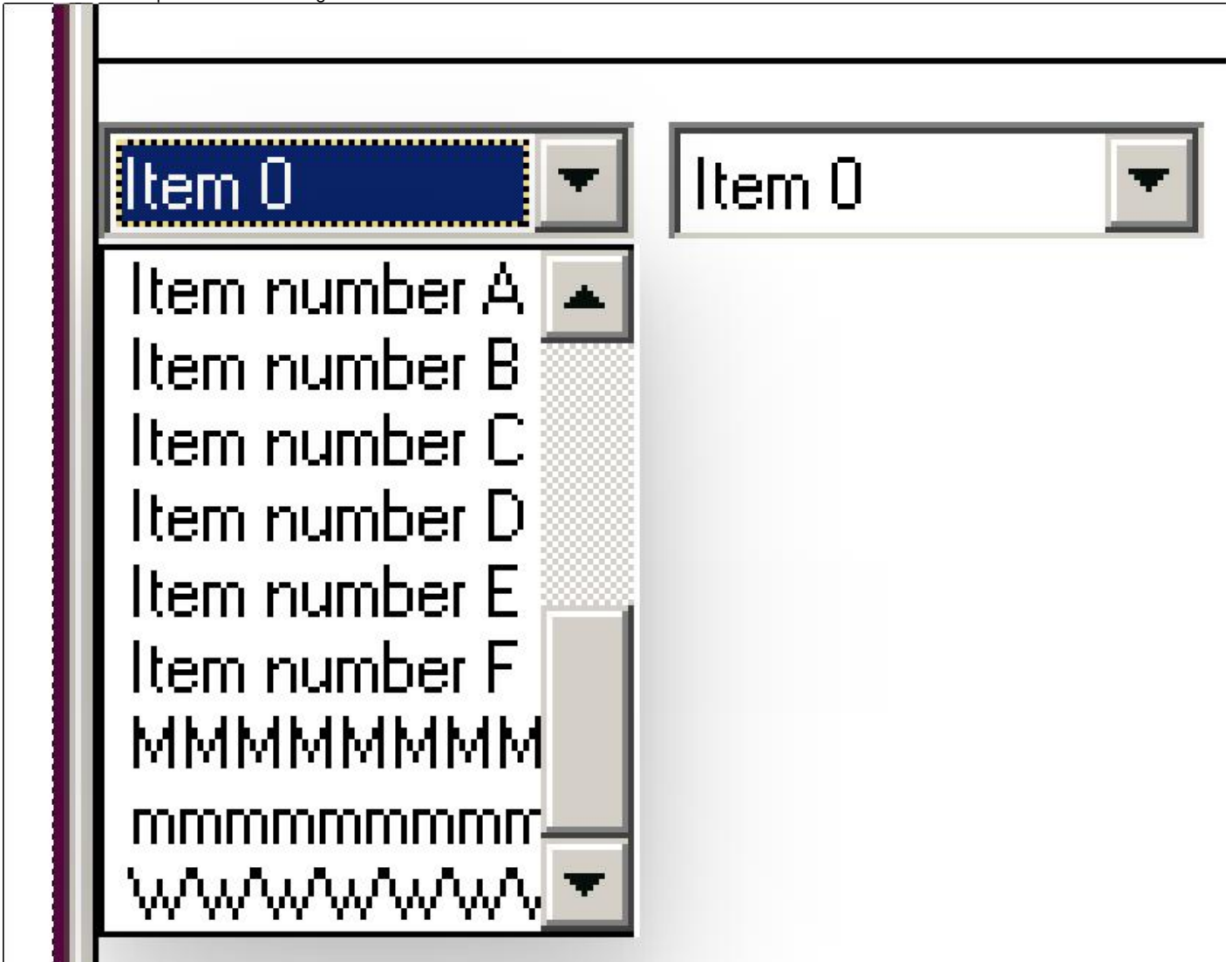
OK, it is much more clear now.

I still want you to finish answering the items in note 84. Then I think we will put this task on hold. I just want the anti-aliasing details to be well known and documented before we stop work on this.

#119 - 06/22/2016 11:00 AM - Sergey Ivanovskiy

- File P2J_Swing_client_combo_box_items_4_times_zoomed.png added

This is the missed picture for the Swing client



#120 - 06/22/2016 11:21 AM - Greg Shah

The swing client seems left-shifted too as compared with the 4GL. It looks just like the firefox case. Both of those seem to have a 2-pixel left shift. The Chrome case seems to have a 1 pixel left-shift.

#121 - 06/22/2016 11:52 AM - Greg Shah

Actually, I should say this differently. The 4GL text starts 2 pixels to the left of the drop-down text from the Swing client. The 4GL text appears to be 1 pixel to the left of the Chrome case (due to anti-aliasing). If you look closely, the Firefox case has a very faint light grey shadow which is drawn in the same place as Chrome. So it really has a similar anti-aliasing effect that makes it actually draw 1 pixel further right than the 4GL text. BUT since the shadow is so hard to see, it appears to draw 2 pixels to the right of where the 4GL is drawing.

#122 - 06/22/2016 04:49 PM - Sergey Ivanovskiy

Yes, agree. To solve this anti-aliasing effect for the Firefox we can try note 93.

#123 - 06/22/2016 07:54 PM - Sergey Ivanovskiy

- File Win7_AniAliasing_test.zip added

To disable fonts anti-aliasing effects for Windows 7 we can follow these rules from

http://answers.microsoft.com/en-us/windows/forum/windows_7-desktop/disable-all-font-smoothing-in-windows-7-ie/f180e803-3317-4433-8fd2-63aadaecc2d2

1. Disable ClearType by opening "Control Panel->Fonts->Adjust ClearType text" and unchecking "Turn on ClearType". Alternatively you can change the value of the following registry key from 2 to 1:

```
[HKEY_CURRENT_USER\Control Panel\Desktop] "FontSmoothingType"=dword:00000001
```

2. Disable font smoothing by unchecking "Smooth edges of screen fonts" in "Control Panel->System->Advanced system settings -> Performance Settings->Visual Effects.

Alternatively you can change the value of the following registry key from 2 to 1:

```
[HKEY_CURRENT_USER\Control Panel\Desktop] "FontSmoothing"="1"
```

3. Restart the computer.

All applications are rendered without anti-aliasing but IE11 continues to render letters on the canvas using fonts smoothing.

I tried to use pixel units for the font size converting points to pixels by this formula $1 \text{ pixel} = (1 \text{ pt} / 0.75 + 0.5) \lfloor 0$, but the result is the same.

#124 - 06/22/2016 08:23 PM - Sergey Ivanovskiy

For Mac OS X El Capitan the Safari web client renders letters on the canvas using fonts smoothing. It seems that we can change only the smoothing effects, but can't disable anti-aliasing for Safari.

I tested these options

1. MAC open System Preferences>General and disable "Use LCD font smoothing when available".
2. Create antialiasing.css file with the following content and apply it to Safari>Preferences...>Advanced>Style sheet: antialiasing.css
3. Test the Safari web client with this antialiasing.css:

```
html {  
  -webkit-font-smoothing: none;  
}
```

or

```
html {  
  -webkit-font-smoothing: subpixel-antialiased;  
}
```

or

```
html {  
  -webkit-font-smoothing: antialiased;  
}
```

4. Observe the differences. In all these cases the font anti-aliasing is applied.
Also I tried to apply this command option:

```
defaults write com.apple.Safari AppleFontSmoothing -int 0
```

and to read Safari options we can execute this command line `defaults read -app Safari`.

#125 - 06/22/2016 08:33 PM - Sergey Ivanovskiy

Also I tried this advised command `defaults write CoreGraphics CGFontDisableAntialiasing YES`, but it doesn't disable fonts anti-aliasing for the Safari web client. I can't find the documentation for this domain CoreGraphics. May be the domain parameter or its key or its value is changed. Refs: <http://apple.stackexchange.com/questions/82700/how-to-switch-off-subpixel-antialiasing-in-safari>,

<http://apple.stackexchange.com/questions/8994/turn-off-anti-aliasing-on-mac>, <https://github.com/zhenglaizhang/mac-setup>,
<http://www.macworld.com/article/1145157/smoothsnow.html>

#126 - 06/23/2016 07:39 AM - Greg Shah

Although the results are not what we hoped for, the details meet the objectives of note 84.

The last thing to do, before we pause this task, is to please post zoomed in images of the problem on both IE and Safari (like you did in note 119).

#127 - 06/23/2016 07:42 AM - Greg Shah

Yes, agree. To solve this anti-aliasing effect for the Firefox we can try note 93.

Understood. It is a good idea.

The thing that I was surprised by was that we need to be shifting the drop-down text drawing left in all cases (even the Swing case is 2 pixels too far to the right). The tricky part is that for the web client we need to adjust this based on your idea (so that when this effect occurs the shift is just 1 pixel instead of 2).

#128 - 06/23/2016 10:18 AM - Sergey Ivanovskiy

- *File ComboBox_Safari_Antialiasing_Enabled_4_times_zoomed.png added*

- *File SafariAntiAliasingEnabled.zip added*

This picture displays the Safari web client with fonts anti-aliasing

Item 0

Item 0

- Item number A
- Item number B
- Item number C
- Item number D
- Item number E
- Item number F
- MMMMMMMN
- mmmmmmmm
- wwwwwwww

#129 - 06/23/2016 10:30 AM - Sergey Ivanovskiy

Anti-aliasing or font smoothing for Mac OS X El Capitan, this command defaults -currentHost write -globalDomain AppleFontSmoothing -int 0 disables "Use LCD font smoothing when available" option that has a short description in the note 124. And this command defaults -currentHost write -globalDomain AppleFontSmoothing -int 2 doesn't enable it. The check box becomes looking like '-' instead of checked 'v' or unchecked ' '.

#130 - 06/23/2016 11:04 AM - Sergey Ivanovskiy

- File *SafariAppleFontSmoothingBlocked.zip* added

Disabling AppleFontSmoothing

```
defaults -currentHost write -globalDomain AppleFontSmoothing -int 0
defaults write com.apple.Safari AppleFontSmoothing -int 0
```

applying Safari>Preferences...>Advanced>Style sheet: antialiasing.css

```
html {
  -webkit-font-smoothing: none;
}
```

and restarting Mac OS doesn't change the canvas fonts look and feel.

#131 - 06/23/2016 11:23 AM - Sergey Ivanovskiy

- File *IE11ClearTypeTurnedOff.zip* added

- File *ComboBoxTest_IE11_ClearTypeTurnedOff_4_times_zoomed.png* added

This picture displays the IE11 web client with ClearType turned off.

Item 0

Item number A

Item number B

Item number C

Item number D

Item number E

Item number F

MMMMMMMMMMMM

mmmmmmmmmm

wwwwwwwwww

Item 0

Files

combobox_content_padding.png	16.9 KB	12/07/2015	Sergey Ivanovskiy
WebClientBoxAllignedText.png	20.8 KB	05/02/2016	Sergey Ivanovskiy
2923_1.txt	19 KB	05/02/2016	Sergey Ivanovskiy
2923_2.txt	2.47 KB	05/02/2016	Sergey Ivanovskiy
4GL_windev01_combobox_fill_in.png	44.4 KB	05/12/2016	Sergey Ivanovskiy
P2J_web_combobox_fill_in.png	39.8 KB	05/12/2016	Sergey Ivanovskiy
combo_box9_1.p	1.51 KB	05/12/2016	Sergey Ivanovskiy
2923_3.txt	14.1 KB	05/12/2016	Sergey Ivanovskiy
2923_4.txt	31.5 KB	05/13/2016	Sergey Ivanovskiy
combo_box_test.png	20.4 KB	05/17/2016	Sergey Ivanovskiy
combo_box9_1.p	2.29 KB	06/02/2016	Sergey Ivanovskiy
4GL_combo_box_zoomed_4.xcf	352 KB	06/02/2016	Sergey Ivanovskiy
web_combo_box_zoomed_4.xcf	364 KB	06/02/2016	Sergey Ivanovskiy
2923a_1.txt	32.7 KB	06/02/2016	Sergey Ivanovskiy
combobox_4GL.png	5.42 KB	06/17/2016	Sergey Ivanovskiy
combobox_P2J_Web.png	17.1 KB	06/18/2016	Sergey Ivanovskiy
combobox_P2J_Swing.png	9.58 KB	06/18/2016	Sergey Ivanovskiy
directory.xml	38.6 KB	06/18/2016	Sergey Ivanovskiy
P2J_Web_Swing_AntiAliasing_Off.png	31.6 KB	06/20/2016	Sergey Ivanovskiy
.fonts.conf	268 Bytes	06/20/2016	Sergey Ivanovskiy
KDE.zip	199 KB	06/21/2016	Sergey Ivanovskiy
Firefox_left_padding_issue.png	169 KB	06/22/2016	Sergey Ivanovskiy
Firefox_lpi_4_times_zoomed.png	33.8 KB	06/22/2016	Sergey Ivanovskiy
Chrome_to_compare.png	204 KB	06/22/2016	Sergey Ivanovskiy
Chrome_to_compare_4_times_zoomed.png	35.4 KB	06/22/2016	Sergey Ivanovskiy
4GL_combo_box_items_4_times_zoomed.png	11.5 KB	06/22/2016	Sergey Ivanovskiy
P2J_Swing_client_combo_box_items_4_times_zoomed.png	25.8 KB	06/22/2016	Sergey Ivanovskiy
Win7_AniAliasing_test.zip	385 KB	06/22/2016	Sergey Ivanovskiy
ComboBox_Safari_Antialiasing_Enabled_4_times_zoomed.png	47.6 KB	06/23/2016	Sergey Ivanovskiy
SafariAntiAliasingEnabled.zip	511 KB	06/23/2016	Sergey Ivanovskiy
SafariAppleFontSmoothingBlocked.zip	514 KB	06/23/2016	Sergey Ivanovskiy
ComboBoxTest_IE11_ClearTypeTurnedOff_4_times_zoomed.png	32.4 KB	06/23/2016	Sergey Ivanovskiy
IE11ClearTypeTurnedOff.zip	71.2 KB	06/23/2016	Sergey Ivanovskiy