

User Interface - Bug #4643

Wrong text width calculation in Swing driver

05/14/2020 09:24 AM - Vladimir Tsichevski

Status:	New	Start date:	05/14/2020
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#1 - 05/14/2020 09:34 AM - Vladimir Tsichevski

- File text-width-measurement-with-gimp.png added

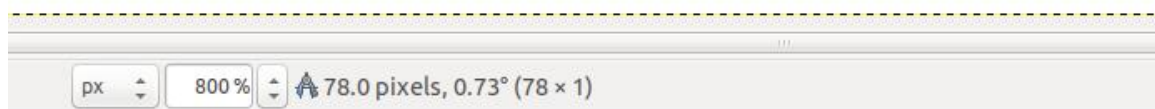
To calculate widget optimal sizes, editor cursor position etc., we need to know the text width for a given string on display.

To get this, we use:

1. `gd.getTextWidth(text, font)` to get the width of the text, or
2. `gd.getTextWidths(text, font)` to get all partial widths.

For Swing driver, the values returned by these methods are incorrect for non-fixed fonts (always lesser than expected).

For example, for the `FontManager.DEFAULT_FONT`, and the string "does not equal", the `gd.getTextWidth()` returns the value 71, and the real value is something around 78:



In AWT, I can do correct measurements with the following code:

```
FontRenderContext frc = new FontRenderContext(null, true, false);
Font font = new Font("Tahoma", Font.PLAIN, 11);
Rectangle2D bounds = font.getStringBounds(TEXT, frc);
double textWidth = bounds.getWidth();
...
```

Note the third argument `usesFractionalMetrics` value for `FontRenderContext` constructor must be `false` here, otherwise the value returned will not match the real text width.

#2 - 05/14/2020 09:37 AM - Vladimir Tsichevski

Comments from GregShah:

Vladimir,

I suspect that the reason is due to the idea that we are trying to size things as closely to the original 4GL text width as we can. The 4GL widget/frame layout is completely based on the specific fonts used in the original system. The Windows font engine calculates metrics for those fonts which are different from the Java font engine and different from how browsers render the same font (even on Windows). If we don't maintain the legacy sizing, then the screens are laid out the wrong way which cannot be accepted.

Of course, you have no such limitations. So my recommendation is to add methods to the drivers/API to provide the results that you > need, bypassing the legacy compatibility versions.

Thanks,
Greg

#3 - 05/14/2020 10:21 AM - Greg Shah

If I get it right, in SwingGuiDriver first AWT internals are used to calculate individual character glyph sizes, but the total text width is calculated naively just as plain sum of character widths, which is obviously not the case if font kerning and/or ligatures are used for text rendering.

Perhaps, but you may also be mistaking the 4GL compatibility behavior here. The 4GL implements much of its behavior by calculating the height and width of a single "character". Internally, much of the layout processing is done using this single character size, even though in variable width fonts each character can have a different size.

Of course, you have no such limitations. So my recommendation is to add methods to the drivers/API to provide the results that you need, bypassing the legacy compatibility versions.

In this case we will probably need to rename old methods

I'd prefer to just use a new name for the new APIs since it will be used much less than the current APIs.

#4 - 05/14/2020 10:24 AM - Vladimir Tsichevski

Greg Shah wrote:

I'd prefer to just use a new name for the new APIs since it will be used much less than the current APIs.

Ok. Which names would you suggest as the alternative for `getTextWidth()` and `getTextWidths()`?

#5 - 05/14/2020 10:26 AM - Greg Shah

I'd prefer to just use a new name for the new APIs since it will be used much less than the current APIs.

Ok. Which names would you suggest as the alternative for `getTextWidth()` and `getTextWidths()`?

`getTextWidthNative()` and `getTextWidthsNative()`

#6 - 05/14/2020 10:28 AM - Constantin Asofiei

The `getStringBounds` approach is different than the `awt.FontMetrics` used by `SwingFontMetrics`. The Native suffix works, but we need comments on how this is done. Also, we need to add a version for the JS driver, too.

#7 - 05/14/2020 02:55 PM - Vladimir Tsichevski

Constantin Asofiei wrote:

The `getStringBounds` approach is different than the `awt.FontMetrics` used by `SwingFontMetrics`. The Native suffix works, but we need comments on how this is done.

Ok

Also, we need to add a version for the JS driver, too.

Does the JS version wrong results as the `SwingGuiDriver` does? If not, then we can just redirect the Native implementation to the original non-Native ones.

#8 - 05/14/2020 03:14 PM - Constantin Asofiei

Vladimir Tsichevski wrote:

Does the JS version wrong results as the SwingGuiDriver does? If not, then we can just redirect the Native implementation to the original non-Native ones.

I don't if they are OK for you. If they are OK, just redirect these APIs in GuiWebDriver to the existing ones.

Files

text-width-measurement-with-gimp.png	12.9 KB	05/14/2020	Vladimir Tsichevski
--------------------------------------	---------	------------	---------------------