

User Interface - Bug #2745

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

fill-in sizing differences

10/08/2015 01:15 PM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	case_num:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to User Interface - Bug #2766: implement a text metrics server process			New

History

#1 - 10/08/2015 01:15 PM - Greg Shah

See [#2564-28](#)

#2 - 10/08/2015 01:17 PM - Greg Shah

The highlighted area issue will be worked in [#2727](#).

The vertical alignment differences will be worked in [#2730](#).

In this task, we only need to address the sizing differences noted.

#3 - 10/08/2015 01:19 PM - Greg Shah

Stanislav: Please post or point to the testcase being used for the problems reported in [#2564-28](#)

#4 - 10/08/2015 02:42 PM - Stanislav Lomany

```
DEF VAR v AS INTEGER FORMAT ">>>>>9" INIT 0.  
DEF VAR w AS INTEGER FORMAT ">>>>>9" INIT 0.
```

```
FORM v w WITH FRAME f1 SIZE 40 BY 10.  
v:BGCOLOR = 11.  
v:FONT = 10.
```

```
w:BGCOLOR = 11.  
w:FONT = 10.
```

```
DISPLAY v w WITH FRAME f1.  
ENABLE w WITH FRAME f1.
```

```
WAIT-FOR CLOSE OF CURRENT-WINDOW.
```

Font is Segoe UI, 14.

#5 - 10/19/2015 06:37 AM - Constantin Asofiei

- Status changed from New to WIP
- Assignee set to Constantin Asofiei

#6 - 10/19/2015 07:07 AM - Constantin Asofiei

This one is tricky: setting FONT=10 via attribute and FONT 10 as widget/frame option produce different fill-in widths... I thought this was something related to Segoe UI, but it is not. The height looks like is the same in both cases.

#7 - 10/19/2015 08:59 AM - Constantin Asofiei

I assumed that it was using the DEFAULT-FONT to compute the width, but this is not the case.

What looked more plausible was that it takes the format length, divides it by two and it uses this length to compute the width, but this stands only for Segoe UI,14.

Greg, do you know if the GUI project uses the FONT attribute? I have a feeling this is another headache to figure out.

#8 - 10/19/2015 09:13 AM - Greg Shah

do you know if the GUI project uses the FONT attribute? I have a feeling this is another headache to figure out.

Sadly, it is used 495 times.

#9 - 10/19/2015 09:13 AM - Constantin Asofiei

Greg Shah wrote:

do you know if the GUI project uses the FONT attribute? I have a feeling this is another headache to figure out.

Sadly, it is used 495 times.

Can you determine the widget types?

#10 - 10/19/2015 09:22 AM - Greg Shah

It is definitely used from FRAME. But it is also used with regular widgets accessed through a HANDLE or WIDGET-HANDLE. Some cases have the handle passed as a parameter to an internal proc that is not called directly from the same file. Other cases the handle is assigned via introspection (e.g. using FIRST-CHILD and NEXT-SIBLING to find contained widgets). It doesn't look easy to pull together a specific list.

Considering that they try to keep tight and consistent control over their font usage, I imagine they may use this technique with all used widget types.

#11 - 10/19/2015 09:25 AM - Constantin Asofiei

Greg Shah wrote:

Considering that they try to keep tight and consistent control over their font usage, I imagine they may use this technique with all used widget types.

OK, thanks for looking into it.

The theory at this time is that 4GL tries to not allow the widget to "leak" outside of its width computed at layout time; I'm trying to (dis)prove this.

#12 - 10/19/2015 10:12 AM - Constantin Asofiei

Constantin Asofiei wrote:

The theory at this time is that 4GL tries to not allow the widget to "leak" outside of its width computed at layout time; I'm trying to (dis)prove this.

This one fails too, setting the FONT attribute to the same font as the one set at the FONT option gets a smaller width.

And I'm getting into the twilight zone... when FONT attribute triggers new width calculation, it checks if the widget has been in a DISPLAY or not. Depending on this state, the new width is different. Also, for numeric fields, if they are DISPLAY'ed (at any point, before setting the FONT), the width remains "frozen"; otherwise, the WIDTH is re-computed.

#13 - 10/19/2015 10:15 AM - Greg Shah

Guy: would you mind reviewing the history of this task and sharing any insight you may have?

Constantin is finding very unusual behavior when using the FONT attribute. More to the point, the behavior is different from that implemented when using the FONT widget option for sizing/layout (statically defined at compile time).

#14 - 10/19/2015 10:49 AM - Guy M

This update includes 2 parts:

- 1) My investigation of OpenEdge FONT attribute
- 2) My organization's usage of the FONT attribute

1) My investigation of OpenEdge FONT attribute

Having looked at the Help on FONT, I think the thing you've maybe overlooked is the AUTO-RESIZE attribute. To quote:

If the AUTO-RESIZE attribute is TRUE, the widget automatically resizes when the LABEL, FONT or FORMAT attributes of the widget change. If AUTO-RESIZE is FALSE, the widget retains its original size.

The default value for this attribute is TRUE for widgets that are not explicitly sized when they are defined, and FALSE for explicitly sized widgets.

When the AUTO-RESIZE attribute is set to TRUE, the AVM resizes button and toggle-box widgets with run-time changes to the LABEL attribute, and combo-box and fill-in field widgets with run-time changes to the FORMAT attribute.

This attribute resizes the following widgets with run-time changes to the FONT attribute:

- Browse columns
- Buttons
- Combo boxes
- Editors
- Fill-ins
- Radio sets
- Selection lists
- Sliders
- Texts
- Toggle boxes

For browse columns, if you set the browse column's AUTO-RESIZE attribute to TRUE, the AVM resizes the browse column when a change occurs in the browse column's font or in the font or text of the browse column's label.

If the font of a browse column grows such that the height needs to be increased, the AVM increases the height of all cells in the browse column, which increases the row height of the browse (because all rows have the same height). This might affect the DOWN, ROW-HEIGHT-CHARS, and ROW-HEIGHT-PIXELS attributes of the browse column.

If the font of a browse column decreases, the AVM does not decrease the height of the rows, because the decrease might clip text in other columns.

Note: If the developer changes the size of the widget at run time by using the HEIGHT-CHARS, HEIGHT-PIXELS, WIDTH-CHARS, or WIDTH-PIXELS attribute, the AVM resets AUTO-RESIZE to FALSE.

2) My organization's usage of the FONT attribute

I've had a brief look at the usage of the FONT attribute, and the cases I've seen fall into 3 categories:

- a) Sausage machine code - this is (very) old character code (pre-1996), that was converted to GUI via our "sausage-machine", which automatically turned character code written using our standard include files into "GUI" code.
- b) Multi-layouts: The Progress AppBuilder allows the design of objects with multiple layouts - which can be chosen and switched between at runtime. It appears that a result of using the multi-layouts is for Progress to insert statements like FONT = ? into the code. Perhaps the developer changed the font of an object on a specific layout, or maybe Progress just does that (I haven't investigated).
- c) Dynamic objects created - particularly for buttons (e.g. in the application panel aorchsp1.w), assign the FONT attribute.

In all these cases, I believe the font that is being assigned is a standard font, and we possibly could re-code to remove the references - e.g. to statically define the font in the frame, and then let the object inherit the font from the frame.

#15 - 10/19/2015 11:37 AM - Constantin Asofiei

Guy Mills wrote:

This update includes 2 parts:

- 1) My investigation of OpenEdge FONT attribute
- 2) My organization's usage of the FONT attribute

Although this is interesting (we are aware about AUTO-RESIZE), the specific issue I'm trying to understand and solve is this:

```
def var ch as char format "x(20)".  
  
form ch font 1 with frame f1 size 40 by 10.  
  
message ch:width-pixels in frame f1.  
ch:font in frame f1 = 1.  
message ch:width-pixels in frame f1.  
  
wait-for close of current-window.
```

As you can see, the font does not change, is just assigned again, but 4GL returns two widths: 110 pixels and 70 pixels. I'm trying to figure why/how just changing the FONT attribute (even to the same font!) resizes the FILL-IN. If you have any ideas about this, please let me know.

#16 - 10/19/2015 12:12 PM - Guy M

Constantin Asofiei wrote:

If you have any ideas about this, please let me know.

...well, if you set AUTO-RESIZE to FALSE in your example, then the size doesn't change, so I guess it might have something to do with the AUTO-RESIZE behaviour (even if it's a bit odd)...

```
def var ch as char format "x(20)".  
  
form ch font 1 with frame f1 size 40 by 10.  
  
message ch:width-pixels in frame f1.  
ch:AUTO-RESIZE = FALSE.  
ch:font in frame f1 = 1.  
message ch:width-pixels in frame f1.  
  
wait-for close of current-window.
```

My guess is that the widget is originally being sized using the font of the frame - which in your example is ? (which I guess defaults to whatever your DEFAULT-FONT is). Then when you resize explicitly, it uses the AUTO-RESIZE functionality to resize to FONT 1.

With my tests, if I set the initial font of ch to ?, I get the same results as your original example.

#17 - 10/19/2015 12:18 PM - Guy M

Scrap that last theory.

I did my test wrong. If I explicitly set the font to ? it also resizes.

Now I see your issue.

#18 - 10/19/2015 12:28 PM - Guy M

Check this out:

```
def var ch as char format "x(20)" INITIAL 12345678901234567890.
form ch font 1 with frame f1 size 40 by 10.
DISPLAY ch WITH FRAME f1.
message ch:width-pixels in frame f1 VIEW-AS ALERT-BOX.
ch:font in frame f1 = ?. /* get the same result as re-assigning to font 1 - disproves my original theory */
message ch:width-pixels in frame f1 VIEW-AS ALERT-BOX.
ch:SCREEN-VALUE = FILL(" ",20).
ch:font in frame f1 = 1.
message ch:width-pixels in frame f1 VIEW-AS ALERT-BOX.
wait-for close of current-window.
```

The results for this are:

150 (my original)
170!!! - and the fill-in resizes to fit in the data
90 - which was what your code on my system originally resized to.

So it seems that the size changes to accommodate the contents. And the default contents are maybe blanks?

#19 - 10/19/2015 12:36 PM - Constantin Asofiei

Guy Mills wrote:

So it seems that the size changes to accommodate the contents. And the default contents are maybe blanks?

Wow, FILL-IN's content was nowhere on my list of probable causes.

I think this is it, if AUTO-RESIZE is ON, changing the FONT will adjust the width so that content fits.

Thanks, I appreciate your help.

#20 - 10/19/2015 12:43 PM - Guy M

I hate to admit it, but it wasn't on my list either.

I was trying to test another idea, hence the long string, but I noticed it resized to fit the contents!

Oh well, sometimes you get lucky :-)

#21 - 10/19/2015 01:36 PM - Greg Shah

Thank you Guy, well done.

I think this is it, if AUTO-RESIZE is ON, changing the FONT will adjust the width so that content fits.

Constantin: do we now have a dynamic text metrics issue?

#22 - 10/19/2015 01:49 PM - Constantin Asofiei

Greg Shah wrote:

Thank you Guy, well done.

I think this is it, if AUTO-RESIZE is ON, changing the FONT will adjust the width so that content fits.

Constantin: do we now have a dynamic text metrics issue?

Yes, the fill-in's width follows the text's real width, and we can't capture metrics for all possible texts.

What we can do is capture metrics for space-filled strings for all formats - this will solve all cases when the font is set before content is placed into the widget.

#23 - 10/20/2015 07:11 AM - Constantin Asofiei

1811s rev 10999 contains the fix for fill-in auto-resize.

I'm putting this rev through runtime testing, there is a change which is sensible (ConfigHelper.setModified).

#24 - 10/22/2015 06:16 AM - Constantin Asofiei

1811s rev 11007 and 11013 fix two regression in reports, related to ConfigHelper.setModified. runtime testing is in progress.

#25 - 10/23/2015 01:40 PM - Constantin Asofiei

- % Done changed from 0 to 100

As 1811s was merged to trunk, this can be closed.

#26 - 10/23/2015 02:03 PM - Greg Shah

- Status changed from WIP to Closed

#27 - 11/16/2016 12:13 PM - Greg Shah

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