User Interface - Feature #3369

add support for BGCOLOR-RGB and FGCOLOR-RGB, FWD extension widget attributes

11/11/2017 08:20 PM - Constantin Asofiei

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		

Description

History

#1 - 11/11/2017 08:46 PM - Constantin Asofiei

3369a was created from trunk rev 11199. It contains the changes to add support for BGCOLOR-RGB and FGCOLOR-RGB FWD extension attributes. These will work like this:

- if the widget's BGCOLOR/FGCOLOR is already set by the business logic, use that. Otherwise, if its -RGB counterpart is set, use it.
- in ABL, it can receive hex values like 0xff0000 for color RED.

Support for WINDOW:BGCOLOR-RGB is not working yet - there is a problem in inheriting the GuiColorResolver from the WindowGuiImpl to WindowWorkspace.

#2 - 11/20/2017 10:30 AM - Constantin Asofiei

3369a rebased from trunk 11201. new rev is 11207.

#3 - 11/20/2017 12:46 PM - Constantin Asofiei

Greg, 3369a rev 11208 is ready for review.

Eric: Persistence.java needs a history entry.

#4 - 11/20/2017 12:58 PM - Constantin Asofiei

Eric, I couldn't reproduce the iframe issue you've reported with 3369a. If you have any other details I can use to duplicate it, please let me know.

#5 - 11/20/2017 01:02 PM - Eric Faulhaber

Constantin Asofiei wrote:

> Eric: Persistence.java needs a history entry.

Done.

#6 - 11/20/2017 01:08 PM - Eric Faulhaber

Constantin Asofiei wrote:

> Eric, I couldn't reproduce the iframe issue you've reported with 3369a. If you have any other details I can use to duplicate it, please let me know.

I'm not seeing this anymore with the latest revision. Tried both FF and Chrome.

#7 - 11/20/2017 01:28 PM - Greg Shah

Code Review Task Branch 3369a Revision 11209

I'm fine with the changes. I checked in some minor changes to make it easier to see these are extension attributes.

Please put this through regression testing.

#8 - 11/21/2017 04:03 PM - Sergey Ivanovskiy

3355a was merged into 3369a rev. 11213.

#9 - 11/22/2017 02:25 PM - Sergey Ivanovskiy

Committed revision 11218 removed waste code due to the web client profiling done by Eric.

#10 - 11/22/2017 02:43 PM - Hynek Cihlar

Revision 11219 contains a fix for unresponsive message boxes. Somebody please review.

#11 - 11/22/2017 03:57 PM - Greg Shah

Code Review Task Branch 3369a Revision 11219

I'm good with the change.

#12 - 11/22/2017 04:02 PM - Hynek Cihlar

3369a 11220 improves performance of coordinate values rounding.

#13 - 11/24/2017 07:25 AM - Constantin Asofiei

3369a rev 11225/11226 fixes class name collisions for converted external program names. Full conversion is required. If the configured OS in p2j.cfg.xml is a case-insensitive, then names will be checked case-insensitive. Otherwise, collisions are avoided on a case-sensitive basis.

Greg, please review.

#14 - 11/24/2017 08:55 AM - Greg Shah

Code Review Task Branch 3369a Revision 11226

I think the code is correct from a logical standpoint.

If we had time, I would prefer for checkClassName to be called disambiguateClassName and for the logic that follows the call to checkClassName to be moved into disambiguateClassName. Instead of editing the rule-set's state, I would prefer the logic and data access to be hidden inside disambiguateClassName. Moving the state inside there would make the result cleaner.

However, with our time constraints in mind, I'm OK with the code as is.

#15 - 11/24/2017 09:19 AM - Constantin Asofiei

Greg Shah wrote:

> Code Review Task Branch 3369a Revision 11226

I fixed a regression in 11227, I was not considering the package, only the class name... result was logically correct, but too broad.

#16 - 11/24/2017 04:28 PM - Sergey Ivanovskiy

Does it need to rebase 3369a?

#17 - 11/24/2017 04:29 PM - Constantin Asofiei

Yes, but I want to wait for Hynek's update first.

#18 - 11/24/2017 04:56 PM - Constantin Asofiei

Hynek Cihlar wrote:

> 3369a 11220 improves performance of coordinate values rounding.

There are still these cases which use BigDecimal:

```
Coordinate.chuiRound(double) (2 matches)
Coordinate.guiRound(double) (2 matches)
CoordinatesConversion.toChar(int, int, RoundingMode) (5 matches)
CoordinatesConversion.toPixels(double, int) (2 matches)
```

#19 - 11/24/2017 05:10 PM - Constantin Asofiei

3369a rev 11231 contains some performance fixes:

1. avoid logging overhead in the EmulatedWindowState and SwingEmulatedWindow
2. avoid unnecessary repaints

Eric, please check the impact.

#20 - 11/24/2017 06:34 PM - Constantin Asofiei

Constantin Asofiei wrote:

> 3369a rev 11231 contains some performance fixes:
>
> 1. avoid unnecessary repaints

There is a regression here in the large GUI app... repaints are avoided too aggressively, some cases do not repaint (i.e. tabs or button 'bold' in the main menu window). Although it might be an issue more of 'repaints are not triggered when they should be', which is just uncovered because of my changes.

#21 - 11/24/2017 06:52 PM - Greg Shah

Can I rebase 3369a? I know the 3375a change is not yet available, but I think it is best to go ahead and pick up the trunk changes so that we can get a conversion run in tonight.

I will rebase in 15 minutes if there are no objections.

#22 - 11/24/2017 07:13 PM - Greg Shah

I'm rebasing now.

#23 - 11/24/2017 07:38 PM - Greg Shah

Branch 3369a rebased from trunk rev 11204. The latest revision is 11236.

This will require a full reconversion of the large GUI application.

#24 - 11/24/2017 07:50 PM - Constantin Asofiei

Constantin Asofiei wrote:

> Constantin Asofiei wrote:
>
>> 3369a rev 11231 contains some performance fixes:
>>
>> 1. avoid unnecessary repaints
>
> There is a regression here in the large GUI app... repaints are avoided too aggressively, some cases do not repaint (i.e. tabs or button 'bold' in the main menu window). Although it might be an issue more of 'repaints are not triggered when they should be', which is just uncovered because of my changes.

Fixed in 3369a 11237.

#25 - 11/24/2017 09:56 PM - Eric Faulhaber

Constantin Asofiei wrote:

> Constantin Asofiei wrote:
>
> > Constantin Asofiei wrote:
> >
> > > 3369a rev 11231 contains some performance fixes:
> > >
> > > 1. avoid unnecessary repaints
> >
> > There is a regression here in the large GUI app... repaints are avoided too aggressively, some cases do not repaint (i.e. tabs or button 'bold' in the main menu window). Although it might be an issue more of 'repaints are not triggered when they should be', which is just uncovered because of my changes.
>
> Fixed in 3369a 11237.

I haven't yet tested with the larger app, as it's reconverting currently. I did test with the Hotel GUI demo and I can confirm that the calendar control is much faster (2x or better). Nice work!

I did find one regression which I assume is from this optimization, but hopefully it is an edge case:

1. bring up the Hotel GUI app and go to the "Rooms" item
2. click "Add Rooms..." to bring up the dialog with the tree control
3. expand the "Double" branch in the tree control
4. expand the "Single" branch

Notice how the "Single" branch expands in height and overpaints/corrupts the top of the "Double" branch. I tested with trunk revision 11204 and it does not do this (the "Double" branch part of the image moves down). I guess that whatever optimization you may be doing with clip regions must be adjusted to repaint more of the buffer when the clip region being repainted changes in size/position. Hopefully, this edge case can be accommodated without regressing the performance gains you've achieved.

Incidentally, I'm seeing the iframe scroll bars again the first time I enter the Hotel GUI embedded mode with 3369a/11237 (and not with trunk 11204), though that seems unrelated to the repaint changes.

#26 - 11/25/2017 07:20 AM - Eric Faulhaber

I did some testing with the large app in Swing mode and in virtual desktop mode (converted with 3369a rev. 11237). It is definitely faster (particularly noticeable in the main menu).

There are some issues with repainting in the ADM screens: e.g., an old tab remains "selected" after selecting a new one; sometimes buttons from old screens are still visible when displaying a new one. I'm not sure if the button issue is a new regression or was already there -- I had seen some issues like this with previous versions. The ADM tab issue is new.

#27 - 11/25/2017 07:20 AM - Constantin Asofiei

Eric Faulhaber wrote:

> There are some issues with repainting in the ADM screens: e.g., an old tab remains "selected" after selecting a new one; sometimes buttons from old screens are still visible when displaying a new one. I'm not sure if the button issue is a new regression or was already there -- I had seen some issues like this with previous versions. The ADM tab issue is new.

Yes, I'm working on these now.

#28 - 11/25/2017 07:33 AM - Eric Faulhaber

An observation: I've noticed that the embedded mode drawing seems much more "crisp" than virtual desktop mode or the Swing client.

By that, I mean, with Swing and virtual desktop mode, I tend to see a screen with multiple widgets painting progressively: the frame is drawn and you can see it visibly being populated with controls from top to bottom and left to right. A browse, then a column or row of buttons, etc. It doesn't all happen at once.

In embedded mode, the entire screen seems to be block transferred at once from an off-screen buffer; you don't see the widgets being drawn individually. It is much more pleasing to the eye. Also, I don't notice the same issue with buttons from one screen being painted incorrectly in another, as I described above for virtual desktop and Swing modes.

#29 - 11/25/2017 07:40 AM - Eric Faulhaber

Eric Faulhaber wrote:

> Also, I don't notice the same issue with buttons from one screen being painted incorrectly in another, as I described above for virtual desktop and Swing modes.

Actually, I did manage to hit this in embedded mode just now, but not as frequently as with the other modes. Haven't figured out a reliable way to recreate it yet.

#30 - 11/25/2017 09:07 AM - Constantin Asofiei

Eric Faulhaber wrote:

> There are some issues with repainting in the ADM screens: e.g., an old tab remains "selected" after selecting a new one;

This one is fixed in 3369a 11238.

> sometimes buttons from old screens are still visible when displaying a new one. I'm not sure if the button issue is a new regression or was already there -- I had seen some issues like this with previous versions.

Can you send me an email with this one? Some recreate/screen-shots?

#31 - 11/27/2017 03:29 AM - Sergey Ivanovskiy

I am using 3369a to work on #3385 and it seems that the Login screen (Swing client) doesn't accept keys input.

#32 - 11/27/2017 03:33 AM - Sergey Ivanovskiy

Don't mind it works now, after external jconsole has been accessed to the client process. It is not a bug probably.

#33 - 11/27/2017 08:56 AM - Constantin Asofiei

Another regression fixed, 3369a rev 11247 is in testing.

#34 - 11/27/2017 01:15 PM - Constantin Asofiei

Constantin Asofiei wrote:

> Another regression fixed, 3369a rev 11247 is in testing.

Rev 11247 passed main part of testing; ctrl-c is in progress.

There is a new #3377 rev 11248 change, which looks risky. Do we continue with 3369a as the cumulative branch?

#35 - 11/27/2017 01:30 PM - Greg Shah

No, I want to get the 3369a into the trunk.

Ovidiu: Would you please move the changes 11248 into a separate branch?

#36 - 11/27/2017 02:29 PM - Ovidiu Maxiniuc

No problem. I am un-committing 3369a/ right now.

#37 - 11/27/2017 03:11 PM - Constantin Asofiei

- % Done changed from 0 to 100

- Status changed from New to WIP

3369a Was merged to trunk rev 11206 and archived.

#38 - 11/27/2017 03:44 PM - Greg Shah

- Status changed from WIP to Closed

#39 - 12/21/2017 12:59 PM - Hynek Cihlar

3369b replaces all UI coordinate roundings implemented by BigDecimal with more primitive constructs, like java.math, hopefully saving some CPU cycles. The changes passed GUI and ChUI regression tests. Please review.

#40 - 12/21/2017 02:20 PM - Greg Shah

Code Review Task Branch 3369b Revision 11211

It is not clear why we need both Coordinate.guiRound() and Coordinate.scale() when they essentially have the same algorithm. There is a missing test for infinity and NaN, but otherwise it is the same code.

Other than this, I'm fine with the changes.

#41 - 12/28/2017 01:35 PM - Greg Shah

This branch needs a rebase. Do you still believe it is safe to merge to trunk? Another option is to merge it into 3435a.

Also, please reply back to the questions in #3369-40.

#42 - 01/04/2018 06:57 PM - Hynek Cihlar

I wasn't in the issue watch list and I missed the last two notes.

I resolved the point in note 40, rebased 3369b and merged it to 3435a as revision 11233. 3369b is archived.

#43 - 01/05/2018 06:40 AM - Greg Shah

Hynek Cihlar wrote:

> I wasn't in the issue watch list and I missed the last two notes.
>
> I resolved the point in note 40, rebased 3369b and merged it to 3435a as revision 11233. 3369b is archived.

I'm fine with the changes.

#44 - 02/05/2018 08:38 AM - Greg Shah

The fix for #3369-18 was put into 3435a. Branch 3435a was merged to trunk as revision 11217.