User Interface - Bug #3069

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

The maximize/restore action for the web client doesn't work properly

04/19/2016 02:53 AM - Sergey Ivanovskiy

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Sergey Ivanovskiy	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Cleanup and Stabilization for GUI		
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 04/19/2016 03:01 AM - Sergey Ivanovskiy

If we press the maximize/restore caption button several times, then the window canvas displays two images one above the other. The top most image is the restored window image and the normal window image is below it.
For this case I found that the widget regions aren't correct because the regions for the caption buttons are shifted.

#2 - 04/19/2016 03:17 AM - Sergey Ivanovskiy

The WebGuiDriver: public synchronized void registerMouseWidgets() was changed to use InstrumentedMap<K,V> and the code became more complicated.

#3 - 04/19/2016 05:47 AM - Constantin Asofiei

Please post a screenshot which shows the issue.

#4 - 04/19/2016 05:50 AM - Sergey Ivanovskiy

- File restore.png added

This picture displays the issue.

#5 - 04/21/2016 04:34 AM - Sergey Ivanovskiy

- File resize&cache.xcf added

The sequence to reproduce:
1) First turn of the p2j.screen module cache to draw without caching, only the p2j.canvas_renderer module cache is active.
2) Run ./hello.p
3) Click restore/maximize button. The result is the window is squeezed. setResize OK
4) Click restore/maximize button again. The result is the window is restored. setResize OK
5) Click restore/maximize button again. The result is the window is squeezed, but the canvas remains the same. Failed. The resize is not send to the client!
Working on why the resize isn't send.

#6 - 04/21/2016 04:34 AM - Sergey Ivanovskiy

- File deleted (resize&cache.xcf)

#7 - 04/21/2016 04:35 AM - Sergey Ivanovskiy

- File resize&cache.png added

#8 - 04/21/2016 10:24 AM - Sergey Ivanovskiy

The problem with the drawing cache system is that resize messages are reached GuiWebDriver correctly, but due to the cache the resize messages are not sent to the JS web client. Since we don't cache the current canvas size with the cached drawing, we can watch this effect. We must save on the client side the current canvas size with the cached image and restore the image and the canvas size.

#9 - 04/21/2016 10:31 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> The problem with cache system is that resize messages are reached GuiWebDriver correctly, but we don't cache the current canvas size with the cached drawing. Because of this we can watch this effect. We must save on the client side the current canvas size with the cached image and restore the image and the canvas size.

Please try this: change GuiWebSocket.md5Hash to include the window's size. This should be enough to distinguish between draw caches with different window sizes.

#10 - 04/21/2016 06:07 PM - Sergey Ivanovskiy

Checked that if we send the batch of resize drawing operations in case of there is a resize drawing operation, then the resize effect is gone. I think the problem is not the two drawing batch operations can't be separated by md5 hash sum but that the canvas can change its size and we don't track its size if the cached drawing is used. It is obvious that we can draw the small cached image on the large canvas but the canvas of the cached image can be less than the current canvas size. Thus we can watch different resize effects.

#11 - 04/22/2016 01:06 PM - Sergey Ivanovskiy

- File 3069_1.txt added

Please review the committed revision 11024. Utilizes the root cause that we mustn't cache the resize operations.

#12 - 04/22/2016 01:28 PM - Greg Shah

Code Review Task Branch 1811u Revision 11024

I'm fine with the changes.

Constantin: do you have any concerns?

#13 - 04/22/2016 02:00 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Please review the committed revision 11024. Utilizes the root cause that we mustn't cache the resize operations.

The idea is interesting, but what I don't understand is why are you sending the resize operations twice... for example, in GuiWebSocket.resizeWindow() you are still calling allocateAndSend(PaintPrimitives.RESIZE_WINDOW, width, height); - which will be batched with other drawing operations.

Is your idea that the resize operations are sent on their own, and the result is not cached?

#14 - 04/22/2016 02:35 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

> Sergey Ivanovskiy wrote:
>
>> Please review the committed revision 11024. Utilizes the root cause that we mustn't cache the resize operations.
>
> The idea is interesting, but what I don't understand is why are you sending the resize operations twice... for example, in GuiWebSocket.resizeWindow() you are still calling allocateAndSend(PaintPrimitives.RESIZE_WINDOW, width, height); - which will be batched with other drawing operations.
>
> Is your idea that the resize operations are sent on their own, and the result is not cached?

Yes, it was sent twice because the order of drawing operations is important. Let drawings be directed by GUI classes. Thus we can't remove the resize operations from this sequence. The second resize is only to place and to resize a canvas. Do you think that we can extract the resize operations to send the batch message without resize, next resize and then without resize in general no more than three batch messages. But it needs more complicated processing.

#15 - 04/22/2016 02:47 PM - Constantin Asofiei

What about my idea in note 9? Why wouldn't including into the md5 hash the window's size (as reported, in pixels, by i.e. WindowGuiImpl APIs) work? My understanding is that the problem is only in the global/full-window cache, not in the per-widget cache, correct?

#16 - 04/22/2016 02:47 PM - Constantin Asofiei

Constantin Asofiei wrote:

> Why wouldn't including into the md5 hash the window's size (as reported, in pixels, by i.e. WindowGuiImpl APIs) work?

I mean only in the md5 hash for the entire drawing op batch, not for the per-widget drawing.

#17 - 04/22/2016 02:51 PM - Sergey Ivanovskiy

Constantin, I don't try it, because the resize operations are performed on canvas object, but all cached drawings are performed on the context, they don't change the canvas size.

#18 - 04/22/2016 02:55 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Constantin, I don't try it, because the resize operations are performed on canvas object, but all cached drawings are performed on the context, they don't change the canvas size.

OK, so what you are saying is the canvas is not resized because the JS side "sees" the image already in the cache and draws it (thus the resize operations are never executed), right? Well, lets try it differently: if a drawing batch contains drawing operations, then just set the global md5 cache to zero (and don't cache the result).

#19 - 04/22/2016 03:00 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

> Sergey Ivanovskiy wrote:
>
>> Constantin, I don't try it, because the resize operations are performed on canvas object, but all cached drawings are performed on the context, they don't change the canvas size.
>
> OK, so what you are saying is the canvas is not resized because the JS side "sees" the image already in the cache and draws it (thus the resize operations are never executed), right? Well, lets try it differently: if a drawing batch contains drawing operations, then just set the global md5 cache to zero (and don't cache the result).

Ok, understand. Do you mean if the resize operation is inside the batch message, then don't cache it. I think it is the best solution.

#20 - 04/22/2016 03:11 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Do you mean if the resize operation is inside the batch message, then don't cache it. I think it is the best solution.

Yes, correctly - sendPendingDrawingOps should not set the global drawing cache.

#21 - 04/22/2016 03:32 PM - Sergey Ivanovskiy

- File 3069_2.txt added

Greg, Constantin please review the committed revision 11025. This fix is improved. Now the batch of drawing operations is not cached if it contains at least one of set location, resize and set bounds.

#22 - 04/22/2016 03:39 PM - Sergey Ivanovskiy

I would like to cleanup this fix, because found that the magic byte array can be static and resizeOps can have type int.

#23 - 04/22/2016 03:54 PM - Sergey Ivanovskiy

- File 3069_3.txt added

Committed rev 11026. Added the summary diff of the effective changes from 11024 to 11026.

#24 - 04/25/2016 03:31 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Committed rev 11026. Added the summary diff of the effective changes from 11024 to 11026.

The changes are OK.

#25 - 04/25/2016 06:33 AM - Greg Shah

- Start date deleted (04/19/2016)

- Assignee set to Sergey Ivanovskiy

Can I close this task?

#26 - 04/25/2016 06:36 AM - Sergey Ivanovskiy

Yes, the task is done.

#27 - 04/25/2016 06:37 AM - Greg Shah

- % Done changed from 0 to 100

- Status changed from New to Closed

- Target version set to Milestone 16

#28 - 11/16/2016 12:22 PM - Greg Shah

- Target version changed from Milestone 16 to Cleanup and Stabilization for GUI

Files

restore.png	46.7 KB	04/19/2016	Sergey Ivanovskiy
resize&cache.png	14.7 KB	04/21/2016	Sergey Ivanovskiy
3069_1.txt	5.25 KB	04/22/2016	Sergey Ivanovskiy
3069_2.txt	3.3 KB	04/22/2016	Sergey Ivanovskiy
3069_3.txt	5.21 KB	04/22/2016	Sergey Ivanovskiy