User Interface - Bug #2621

wait-for only works for current-window

07/25/2015 09:20 AM - Greg Shah

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Constantin Asofiei		**% Done:**	0%
Category:			**Estimated time:**	0.00 hour
Target version:	GUI Support for a Complex ADM2 App			
billable:	No		**case_num:**	
vendor_id:	GCD		**version:**	
Description				

History

#1 - 07/25/2015 09:20 AM - Greg Shah

WAIT-FOR needs to be fully generalized to any window handle.

#2 - 07/26/2015 01:14 PM - Constantin Asofiei

- Status changed from New to WIP

Created branch 2621a from trunk rev 10900.

#3 - 07/26/2015 01:49 PM - Constantin Asofiei

The wait-for window-close of h or close of h where h is a window seems to work, but there some issues:

- solved - there are some artifacts on h when a widget is enabled and accepts focus on the default-window: caret is being drawn on h and at some point (not a reliable recreate yet) the enabled widget is displayed on h too.
- when switching windows, the fill-in gets selected (I guess a focus-gained event is raised)
- when moving frame between windows (via view ... in window), the fill-in content gets reset.
- solved - also, window:title no longer works (is not displayed in the GUI window).

I'll work on all these issues as part of this task.

#4 - 07/30/2015 03:50 PM - Constantin Asofiei

Please review branch 2621a, revision 10905. It contains these fixes:

- FramePlacementManager must be window specific
- fixed window title drawing
- fixed caret drawing

The other issues - fill-in gets selected and also reset - is related to focus processing when there are multiple windows. I think this requires a task of its own; my feeling at this time is: what if each 4GL window has its own focus management? This is indicated if there are two windows, each one with two or more enabled frames: when switching focus between windows, the window-level focused widget does not changed, when focus is re-gained by the window.

#5 - 07/31/2015 06:19 AM - Constantin Asofiei

Rebased branch 2621a, from trunk rev 10905 (new branch rev 10906)

#6 - 07/31/2015 02:54 PM - Greg Shah

Code Review Task Branch 2621a Revision 10906

I like the changes.

My only question: Should the gd.selectWindow(window.config().id.asInt()); that you added to EditorGuiImpl and FillInGuiImpl actually be a standard feature of AbstractGuiDriver.draw() and AbstractGuiDriver.drawFromOrigin()? This seems like it will be a common requirement that is better done in common code than in each widget's drawing method.

#7 - 07/31/2015 03:04 PM - Constantin Asofiei

Greg Shah wrote:

> Code Review Task Branch 2621a Revision 10906
>
> I like the changes.
>
> My only question: Should the gd.selectWindow(window.config().id.asInt()); that you added to EditorGuiImpl and FillInGuiImpl actually be a standard feature of AbstractGuiDriver.draw() and AbstractGuiDriver.drawFromOrigin()? This seems like it will be a common requirement that is better done in common code than in each widget's drawing method.

At some point the assumption behind draw() was that OutputManager.setInvalidate will take care of selecting the window before the window drawing code is executed. But there are cases when drawing is initiated outside of drawing brackets.

Forcing window selection at the driver level is a solution, but this will mean that the current EmulatedWindowState needs to be interrogated each time draw() is called to get the current window ID. And for the web driver this will require a trip to the browser.

Also, the cases you noted at [#1811](#) about the GuiDriver.translate calls in ThinClient APIs need to be fixed so that the window is explicitly selected before drawing is done. At some point, they will become problematic.

Explicit window selection should be the exception, not the norm - the norm should be that when draw is called, the window is already selected. The draw caret code is just a part of the APIs which can be called outside of our normal setInvalidate path (the drawing bracket).

#8 - 07/31/2015 03:14 PM - Greg Shah

but this will mean that the current EmulatedWindowState needs to be interrogated each time draw() is called to get the current window ID. And for the web driver this will require a trip to the browser.

I'm tracking the windowId on the Java side so there is no need for a trip down. In fact it comes originally from the Java side and I just reuse that same ID on the javascript side to map to the javascript Window object that contains the canvas and backs all window functions.

Still, since it is not a universal requirement, it seems best to avoid the approach.

Also, the cases you noted at [#1811](#1811) about the GuiDriver.translate calls in ThinClient APIs need to be fixed so that the window is explicitly selected before drawing is done. At some point, they will become problematic.

I guess you may as well fix them now as part of this task.

#9 - 08/03/2015 10:43 AM - Constantin Asofiei

Rebased branch 2621a, from trunk rev 10906 (new branch rev 10907)

#10 - 08/04/2015 03:54 AM - Constantin Asofiei

Rebased branch 2621a, from trunk rev 10907 (new branch rev 10909)

#11 - 08/04/2015 09:42 AM - Greg Shah

Code Review Task Branch 2621a Revision 10909

It looks good. I think this can go into testing unless there is anything else to do.

#12 - 08/04/2015 01:34 PM - Constantin Asofiei

Rebased branch 2621a, from trunk rev 10908 (new branch rev 10910)

#13 - 08/05/2015 03:24 AM - Constantin Asofiei

Rebased branch 2621a, from trunk rev 10909 (new branch rev 10911)

Also, it has passed runtime testing.

#14 - 08/05/2015 12:36 PM - Greg Shah

Please merge to trunk now.

#15 - 08/05/2015 12:41 PM - Constantin Asofiei

- Status changed from WIP to Review

Branch 2621a was merged to trunk rev 10910 and archived.

#16 - 08/05/2015 01:01 PM - Greg Shah

- Status changed from Review to Closed

#17 - 11/16/2016 12:13 PM - Greg Shah

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