User Interface - Bug #3093

widget-pool resources need to be pre-ordered when deleted (to enforce parent/child relationshinps for widgets)

04/29/2016 01:35 PM - Constantin Asofiei

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 04/29/2016 01:40 PM - Constantin Asofiei

In case of widgets added to widget-pools, the order needs to be deterministic: if the entire UI tree is added in a widget pool (i.e. window, frame, menu, widgets, etc), then they need to be destroyed in a pre-determined order: window/frame/menu last, everything else sooner.

What is unknown is if some other resources might have the same problem (x-document, x-noderef).

See this test for widget case:

```
create widget-pool "a".

def var hw as handle.
def var hb as handle.
def var hf as handle.

create window hw in widget-pool "a".
hw:width-pixels = 200.
hw:height-pixels = 200.

create toggle-box hb in widget-pool "a".
hb:WIDTH-CHARS = 2.
hb:HEIGHT-CHARS = 2.

create frame hf in widget-pool "a".
hf:width-chars = 20.
hf:height-chars = 5.

hb:frame = hf.
hf:parent = hw.

hf:visible = true.
hb:VISIBLE = TRUE.
hw:visible = true.
PAUSE IN WINDOW hw.

hf:visible = false.
hw:visible = false.

delete widget-pool "a".
```

#2 - 04/29/2016 01:42 PM - Constantin Asofiei

- Assignee set to Constantin Asofiei

- Status changed from New to WIP

I think a similar issue exists if the hw and hf handles are deleted before the delete widget-pool statement - hb widget, when trying to hide it, is seen as an orphan with no window.

#3 - 05/03/2016 10:06 AM - Constantin Asofiei

In the end, the issue is not related to widget-pools. Is related to how dynamic windows or frames get deleted:

1. if the dynamic window gets deleted, all frames are detached from the window
2. if the dynamic frame gets deleted, then all its widgets (dynamic or not) get deleted
3. if the dynamic window gets not-visible (i.e. a HIDE), the VISIBLE attribute is reported as FALSE for all widgets in the window's UI tree (but this is more for the #2809 task)

The most issues/abends arise when any of the above is done from a trigger: as the event processing had a valid window before the trigger was invoked, that window might be invalid/non-existent after the trigger has finished.

Also, another found issue: CLOSE and WINDOW-CLOSE events are equivalent for an WAIT-FOR (need to test for triggers, too).

#4 - 05/05/2016 08:32 AM - Constantin Asofiei

I've created branch 3093a from trunk rev 11022. Rev 11023 fixes the close button issue in task #2898, but the fix is not yet complete - there are other cases (found in uast/res_delete/widget-order-delete tests) which still fail.

Also, there are other areas to investigate, like what happens if a frame which was in a deleted window is attached to another window: if the frame was visible in the deleted window, will it still be visible in the new window? The reason to investigate this is that after the frame's window gets deleted, FRAME:VISIBLE is reported as false. If this attribute is somehow linked to the frame's realized state (i.e. the frame becomes 'unrealized' after its window gets deleted), then changing the actual VISIBLE attribute at the frame will not be correct...

#5 - 09/15/2016 07:29 AM - Constantin Asofiei

Greg, please review 3093b rev 11096: it fixes the Close button issue and also some other issues reported by Sergey/Eugenie. 3093b passed MAJIC testing and 454 test functionality is OK.

Also, 3093a was archived as dead (as it contained some changes which were incorrect, but are left there for historically reasons).

#6 - 09/15/2016 09:57 AM - Greg Shah

Code Review Task Branch 3093b Revision 11096

These changes look long overdue.

1. The removal of the dynamic check in GenericFrame.getStaticWidgetIds() is counterintuitive. If it is correct/intended, then please add a comment to explain why it is not needed.

2. For this code from ThinClient.invokeTriggers():

```
            if (frame != null)
            {
                frame = getWidget(frame.getId().asInt());
            }
```

I think a comment should be placed before frame = getWidget(frame.getId().asInt()); to explain that the value may now be null if the frame is invalid.

The inFocus = getWidget(inFocus.getId().asInt()); needs the same comment.

3. On frame delete, does the 4GL really re-parent static child frames to the default window?

4. This re-parenting on delete only happens on the client-side. Is the server-side state kept fully consistent with these changes?

#7 - 09/16/2016 05:46 AM - Constantin Asofiei

Greg Shah wrote:
3093b was rebased from trunk 11094 and rev 11098 contains the comment changes.

> 1. The removal of the dynamic check in GenericFrame.getStaticWidgetIds() is counterintuitive. If it is correct/intended, then please add a comment to explain why it is not needed.

Done

> 2. For this code from ThinClient.invokeTriggers():

Done

> 3. On frame delete, does the 4GL really re-parent static child frames to the default window?

Actually, no. This is part of another issue in P2J: frames can't exist without being added to a window, and when frame.openScope() is called, it adds it to a current or default window. There is lots of code which relies on a frame always being attached to a window, so it can't be easily reworked.

> 4. This re-parenting on delete only happens on the client-side. Is the server-side state kept fully consistent with these changes?

There are two cases here:

1. on client-side, when a frame gets deleted, all static child frames are attached to the current/default-window. When a window gets deleted, all its frames get attached to the current/default-window. All these parent/window related configs set by the client are later transferred to server-side, so its kept in sync.
2. on server-side, P2J support for WINDOW:FIRST-CHILD currently works only with WINDOW children. But, a root frame can iterate over its child frames too, via FIRST-CHILD/NEXT-SIBLING.

The conclusion is, with current trunk/3093b the reparenting is OK; only issue is when we make WINDOW:FIRST-CHILD to iterate over frames too, we will need some changes (especially related to the 'P2J keeps all frames in a window, regardless if they should not be attached, on client-side' issue.

#8 - 09/16/2016 07:30 AM - Greg Shah

> > 3. On frame delete, does the 4GL really re-parent static child frames to the default window?
>
> Actually, no. This is part of another issue in P2J: frames can't exist without being added to a window, and when frame.openScope() is called, it adds it to a current or default window. There is lots of code which relies on a frame always being attached to a window, so it can't be easily reworked.

This seems likely to cause problems. Any application code that uses the default-window or current-window after an unexpected frame has been added will potentially break because the state is different. I would expect problems both from application code (iterating next/prev handles for siblings/parent/children) and from all the places where we internally iterate over frames and/or children of the window. There are many such places and more be added or edited all the time. It seems very difficult to enforce restrictions on/make safe all these cases.

It would be safer to create a "fake" window (one that cannot be traversed as a next/prev handle) as a container for these so that the application has less of a chance to break. Even that would not be quite right because the window attribute could be examined. We might want to mask out the fake window from the value returned by that attribute so converted code can't "see" the window.

#9 - 09/16/2016 02:30 PM - Constantin Asofiei

Greg, 3093b can be committed as is, as it fixes known issues in 454 and 274 (#3110) tests.

The issue in previous note, I'm not aware of affecting it the scenarios, and can be postponed.

#10 - 09/16/2016 03:14 PM - Greg Shah

OK, please merge 3093b to trunk.

Please create a new sub-task of #2677 for deferred working of the issue in note 8. Then we will close this task, unless there are some other issues to resolve.

#11 - 09/16/2016 04:37 PM - Constantin Asofiei

Greg Shah wrote:

> OK, please merge 3093b to trunk.

Merged to trunk rev 11095 and archived.

> Please create a new sub-task of #2677 for deferred working of the issue in note 8.

See #3179

> Then we will close this task, unless there are some other issues to resolve.

I have some tests which have trigger-related issues (see testcases/uast/res_delete/widget-delete-order3.p, 4 and 5 tests). Not sure yet if they are related to the trigger event combination (CLOSE of THIS-PROCEDURE, APPLY) or widget deletion.

#12 - 09/21/2016 09:35 AM - Greg Shah

> I have some tests which have trigger-related issues (see testcases/uast/res_delete/widget-delete-order3.p, 4 and 5 tests). Not sure yet if they are related to the trigger event combination (CLOSE of THIS-PROCEDURE, APPLY) or widget deletion.

Is this still something being investigated?

#13 - 09/21/2016 09:45 AM - Constantin Asofiei

Greg Shah wrote:

> > I have some tests which have trigger-related issues (see testcases/uast/res_delete/widget-delete-order3.p, 4 and 5 tests). Not sure yet if they are related to the trigger event combination (CLOSE of THIS-PROCEDURE, APPLY) or widget deletion.
>
> Is this still something being investigated?

Yes, I think is related to trigger/APPLY.