

User Interface - Feature #1798

implement full frame family support (parent-child relationships between frames)

10/30/2012 10:11 AM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	100%
Category:		Estimated time:	120.00 hours
Target version:	GUI Support for a Complex ADM2 App	vendor_id:	GCD
billable:	No		
Description			

History

#1 - 10/31/2012 02:51 PM - Greg Shah

- Target version set to Milestone 12

#2 - 01/07/2015 03:00 PM - Greg Shah

I think the conversion already supports these features. If I understand correctly, the work is on the runtime side.

I think both PARENT and FIRST-CHILD/LAST-CHILD need some backing implementation code for widgets.

GenericWidget already extends HandleChain, but I'm not sure if all of our sibling linkages are setup yet, so NEXT-SIBLING/PREV-SIBLING are probably not functional either.

All of the above is needed for M12.

#3 - 01/07/2015 03:04 PM - Greg Shah

The above notes are really about the widget-level requirements, which perhaps should be placed in another task. I do know that we need the similar features implemented for frames too.

Window family support is needed as well. We can split that off into another task if that is worthwhile. Window family support is manifested in things like visibility control. For example, when a parent window is minimized, all child windows are hidden. On the other hand the docs say that the window parenting doesn't affect z-order.

#4 - 04/16/2015 05:58 AM - Constantin Asofiei

- Assignee set to Constantin Asofiei

- Status changed from New to WIP

Window family support is needed as well. We can split that off into another task if that is worthwhile. Window family support is manifested in things like visibility control. For example, when a parent window is minimized, all child windows are hidden. On the other hand the docs say that the window parenting doesn't affect z-order.

The runtime support for window family is in rev 10836 (update ca_upd201504315c.zip from [#2252](#)) and notes 896, 897 from [#2252](#).

#5 - 04/16/2015 03:18 PM - Constantin Asofiei

- File ca_upd20150416c.zip added

This update adds server-side support for FIRST/LAST-CHILD and finishes server-side support for PARENT.

Also, NEXT/PREV-SIBLING were fixed to work in with widget tree structures.

Runtime/conversion testing is in progress.

TODOs: continue testing the FRAME's FIRST/LAST-CHILD and PARENT (actually, when the frame is attached but not to a window, the parenting is done at the FIELD-GROUP, not FRAME...).

#6 - 04/17/2015 11:00 AM - Greg Shah

Code Review ca_upd20150416c.zip

It looks good. It may make sense to not check this in until after Hynek checks in his config changes. Please discuss with him before you check in.

#7 - 04/17/2015 02:14 PM - Constantin Asofiei

Greg Shah wrote:

Code Review ca_upd20150416c.zip

It looks good. It may make sense to not check this in until after Hynek checks in his config changes. Please discuss with him before you check in.

Yes, I'll postpone release until after Hynek finishes the config changes.

0416c.zip had a regression and I have some other fixes in places which need to be tested. What is missing currently is a config ID for the FIELD-GROUP widget: a frame (child of another frame) has the PARENT attribute set to the FIELD-GROUP, and we can't reference it without a FIELD-GROUP config ID.

#8 - 04/17/2015 02:39 PM - Constantin Asofiei

I was under the impression that we removed the "use a base ID for determining the frame's widget IDs", but we still use this, at least in TC.setScreenBuffer:

```
int  baseId = frameBuf.getBaseId();
int  size   = frameBuf.size();
int  id     = -1;
...
```

```
// iterate through all widgets and push new values and state as needed
for (int i = 0; i < size; i++)
{
    id = baseId + i + 1;
    state = frameBuf.getState(id);
    ...
}
```

This will cause problems when dynamic widgets are added to a frame, because this will no longer follow the "contiguous widget ID interval" approach we use now. At this moment, adding an ID to the FIELD-GROUP (only on server-side, as the client-side doesn't need them) will cause a NPE, because I'm breaking the "contiguous widget ID interval" approach.

#9 - 04/17/2015 02:50 PM - Greg Shah

I was under the impression that we removed the "use a base ID for determining the frame's widget IDs", but we still use this, at least in TC.setScreenBuffer

We had intended to have removed all of it. It looks like this case was missed.

Please do fix it.

#10 - 04/17/2015 03:57 PM - Constantin Asofiei

- File *ca_upd20150417e.zip* added

Replacement of 0416c.zip. Fixed a regression and some other issues.

#11 - 04/22/2015 02:54 AM - Constantin Asofiei

- File *ca_upd20150421a.zip* added

This version passed runtime testing (fixes problems in TC.setScreenBuffer).

Will retest/release after HC's config rework update.

#12 - 04/22/2015 11:07 AM - Greg Shah

Code Review *ca_upd20150421a.zip*

I'm good with the changes.

#13 - 05/01/2015 05:44 AM - Constantin Asofiei

- File `ca_upd20150501a.zip` added

Merged version of 0421a.zip with rev 10844. Conversion testing passed, runtime testing is starting (it should pass too, will be finished in ~6 hours).

#14 - 05/01/2015 10:35 AM - Greg Shah

Code Review `ca_upd20150501a.zip`

It looks good. Check it in when it passes.

#15 - 05/01/2015 11:14 AM - Constantin Asofiei

Greg Shah wrote:

Code Review `ca_upd20150501a.zip`

It looks good. Check it in when it passes.

Committed revision 10845.

#16 - 11/23/2015 03:46 PM - Constantin Asofiei

2677b rev 10953 contains second phase of frame family support (plus some other fixes). Please review.

Runtime testing is in progress.

#17 - 11/24/2015 06:47 AM - Constantin Asofiei

2677b rev 10954 fixes a bug in `isRootFrame()`. main part is in progress.

#18 - 11/24/2015 07:36 AM - Hynek Cihlar

Constantin Asofiei wrote:

2677b rev 10953 contains second phase of frame family support (plus some other fixes). Please review.

In `ThinClient.placeFrameInt()` there is a comment "only root frames are placed on window" but the conditional checking whether frame is root was removed.

Not part of the change set, but the line in `ThinClient.processEventsWorker()` looks suspicious:

```
boolean repaint = src.parent() != null && srcFrame == src.window().getTopVisibleFrame();
```

The `window()` call should be probably replaced with `topLevelWindow()` otherwise wrong frame will be used when in DIALOG.

Similar issue as above in `UiUtils.getFramesAbove()`. In case of a DIALOG the method will collect widgets of a wrong window.

#19 - 11/24/2015 07:45 AM - Greg Shah

Code Review Task Branch 2677b Revision 10954

My only question is about the safety of `bounds.intersects(outer)` in `AbstractContainer.draw()` as compared to the previous use of `getClippings()` to determine if we should draw. I can see this reducing drawing in some cases and increasing drawing in other cases. It is not intuitively clear to me if this is safe.

#20 - 11/24/2015 08:43 AM - Constantin Asofiei

Greg Shah wrote:

Code Review Task Branch 2677b Revision 10954

My only question is about the safety of `bounds.intersects(outer)` in `AbstractContainer.draw()` as compared to the previous use of `getClippings()` to determine if we should draw. I can see this reducing drawing in some cases and increasing drawing in other cases. It is not intuitively clear to me if this is safe.

In GUI, all drawing (driver-level) is made using as clipping rectangle the outer rectangle (minimum rectangle which contains all posted rectangles, from `PaintEvent`). This is required to be consistent, otherwise, if a lower (in z-order) frame partially intersects the clipped regions (think of two buttons, in opposite corners, with another frame below them) then it will be drawn using the outer rectangle; we don't have a mechanism to re-draw the frame (and content) for each and every posted rectangle (and not sure how fast this will be, as it will require re-drawing for each rectangle). So, as the lower frame draws using as clip the outer rectangle, anything above it (widgets, frames, etc) needs to re-draw, even if they do not intersect the individual rectangles (from `PaintEvent`)... this is what the changes in `AbstractContainer.draw()` does: it enforces consistencies between the clip rectangle set at drawing time (driver level) and the checks to determine the widgets which need to be drawn.

#21 - 11/24/2015 08:47 AM - Constantin Asofiei

Hynek Cihlar wrote:

Constantin Asofiei wrote:

2677b rev 10953 contains second phase of frame family support (plus some other fixes). Please review.

In `ThinClient.placeFrameInt()` there is a comment "only root frames are placed on window" but the conditional checking whether frame is root was removed.

Thanks, I'll remove it.

Not part of the change set, but the line in `ThinClient.processEventsWorker()` looks suspicious:
`boolean repaint = src.parent() != null && srcFrame == src.window().getTopVisibleFrame();`
The `window()` call should be probably replaced with `topLevelWindow()` otherwise wrong frame will be used when in `DIALOG`.

Similar issue as above in `UiUtils.getFramesAbove()`. In case of a `DIALOG` the method will collect widgets of a wrong window.

OK, I'm testing this to see how GUI reacts.

#22 - 11/24/2015 08:52 AM - Greg Shah

What else is left to do for this task (besides regression testing)?

#23 - 11/24/2015 09:14 AM - Constantin Asofiei

Hynek Cihlar wrote:

Similar issue as above in `UiUtils.getFramesAbove()`. In case of a `DIALOG` the method will collect widgets of a wrong window.

This one can't be switched to `topLeveWindow()` as `UiUtils.getTopLevelComponents()` requires a `Window`. Also, how should a `ModalWindow` return its widgets - shouldn't this have only one widget, for `DIALOG` (i.e. the underlying frame)?

#24 - 11/24/2015 09:23 AM - Hynek Cihlar

Constantin Asofiei wrote:

Hynek Cihlar wrote:

Similar issue as above in `UiUtils.getFramesAbove()`. In case of a `DIALOG` the method will collect widgets of a wrong window.

This one can't be switched to `topLeveWindow()` as `UiUtils.getTopLevelComponents()` requires a `Window`.

Yes, this doesn't seem to be related to your case. The method is used in other cases valid for a `DIALOG`. For example from `ThinClient.waitForWorker()`. I will check this.

Also, how should a `ModalWindow` return its widgets - shouldn't this have only one widget, for `DIALOG` (i.e. the underlying frame)?

Yes, `DIALOG` window should only host one frame. There is an abstract method `TopLevelWindow.getFrames()` that does frame enumeration in `ModalWindow` (and other `TopLevelWindow` subtypes as well).

#25 - 11/24/2015 09:39 AM - Constantin Asofiei

Hynek Cihlar wrote:

Constantin Asofiei wrote:

Hynek Cihlar wrote:

Similar issue as above in `UiUtils.getFramesAbove()`. In case of a `DIALOG` the method will collect widgets of a wrong window.

This one can't be switched to `topLeveWindow()` as `UiUtils.getTopLevelComponents()` requires a `Window`.

Yes, this doesn't seem to be related to your case. The method is used in other cases valid for a `DIALOG`. For example from `ThinClient.waitForWorker()`. I will check this.

Also, how should a `ModalWindow` return its widgets - shouldn't this have only one widget, for `DIALOG` (i.e. the underlying frame)?

Yes, `DIALOG` window should only host one frame. There is an abstract method `TopLevelWindow.getFrames()` that does frame enumeration in `ModalWindow` (and other `TopLevelWindow` subtypes as well).

OK, then I'm leaving both of the issues you mentioned in note 18 to you.

#26 - 11/24/2015 09:39 AM - Constantin Asofiei

Greg Shah wrote:

What else is left to do for this task (besides regression testing)?

I can't think of anything else to test.

#27 - 11/24/2015 10:03 AM - Hynek Cihlar

Constantin Asofiei wrote:

Yes, DIALOG window should only host one frame. There is an abstract method `TopLevelWindow.getFrames()` that does frame enumeration in `ModalWindow` (and other `TopLevelWindow` subtypes as well).

OK, then I'm leaving both of the issues you mentioned in note 18 to you.

Yes, no problem. I have moved this out to #2875.

#28 - 11/25/2015 03:47 AM - Constantin Asofiei

2677b rev 10960 fixes another regression (looks like is last one). runtime testing is in progress.

#29 - 11/25/2015 08:39 AM - Constantin Asofiei

Constantin Asofiei wrote:

2677b rev 10960 fixes another regression (looks like is last one). runtime testing is in progress.

rev 10960 passed runtime testing.

#30 - 11/25/2015 08:42 AM - Greg Shah

- % Done changed from 0 to 100
- Status changed from WIP to Closed

Well done!

#31 - 03/07/2016 02:41 PM - Greg Shah

Constantin: what testcases can be used to explore our frame family support (especially in regard to focus processing)? These will be used by Sergey in working [#2954](#).

#32 - 03/08/2016 07:05 AM - Constantin Asofiei

See the testcases/uast/frame-z-order/frame-parenting.p, nested-frames.p and frame-parenting2.p tests.

#33 - 11/16/2016 12:12 PM - Greg Shah

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

Files

ca_upd20150416c.zip	250 KB	04/16/2015	Constantin Asofiei
ca_upd20150417e.zip	386 KB	04/17/2015	Constantin Asofiei
ca_upd20150421a.zip	388 KB	04/22/2015	Constantin Asofiei
ca_upd20150501a.zip	389 KB	05/01/2015	Constantin Asofiei