# User Interface - Bug #2520

## Dynamic  list controls are not visible

02/20/2015 07:47 AM - Igor Skornyakov

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 02/20/2015 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Igor Skornyakov | **% 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**

**Related issues:**

| | |
|---|---|
| Related to User Interface - Feature #2522: add attribute/method support for r... | **Closed** |

### History

**#1 - 02/20/2015 01:37 PM - Greg Shah**

*- Description updated*

The dynamic RADIO-SET and COMBO-BOX are not visible, but are "working" - one can make a "blind" selection with the RADIO-SET and the corresponding trigger will be activated. With the COMBO-BOX the selection drop-down list is activated (but at a wrong position). I haven't tried SELECTION-LIST.

**#2 - 02/20/2015 01:38 PM - Greg Shah**

Is this still a problem on bzr rev 10766?

**#3 - 02/20/2015 01:47 PM - Igor Skornyakov**

Greg Shah wrote:

> Is this still a problem on bzr rev 10766?

Yes, it is.

**#4 - 02/20/2015 05:18 PM - Greg Shah**

*- Assignee set to Igor Skornyakov*

*- Target version set to Milestone 12*

**#5 - 03/06/2015 08:46 AM - Igor Skornyakov**

It seems that I've resolved the issue (with one-line fix). Will start regression testing after the one for #2508 will be finished. As far as I understand my next task is #2522 but it is not formally assigned.

**#6 - 03/07/2015 03:51 AM - Igor Skornyakov**

*- File ias_upd20150307a.zip added*

Fixed. Started regression test

**#7 - 03/08/2015 05:20 AM - Igor Skornyakov**

Has passed conversion regression testing and is committed as bzr rev. 10797.

**#8 - 03/09/2015 09:08 AM - Greg Shah**

Igor: in the future, please do not check anything in before you are explicitly approved to do so. Such approval is also preconditioned upon a code review.

Constantin: please review the change (bzr rev 10797). Is the modification safe?

**#9 - 03/09/2015 09:12 AM - Igor Skornyakov**

Greg Shah wrote:

> Igor: in the future, please do not check anything in before you are explicitly approved to do so. Such approval is also preconditioned upon a code review.

OK, sorry about that.

**#10 - 03/09/2015 10:35 AM - Constantin Asofiei**

Greg Shah wrote:

> Constantin: please review the change (bzr rev 10797). Is the modification safe?

It's hard to tell. The Frame.savedWidgets is used to create a snapshot of the frame's widgets, so that the frame dimension is not recomputed each time the frame definition is pushed to the client.

Setting the savedWidgets to null will invalidate the saved frame size state and re-compute it. This may result in a different layout next time the frame is shown.

In terms of dynamic widget/frames, I think savedWidgets may need some rework, as this will not get updated if new widgets are added to the frame, after it was made visible.

So, Igor: what issue were you trying to solve? Do you have a standalone test for it?

**#11 - 03/09/2015 10:52 AM - Igor Skornyakov**

*- File w3x.p added*

Constantin Asofiei wrote:

> So, Igor: what issue were you trying to solve? Do you have a standalone test for it?

Constantin,
The problem was that dynamic widgets where not visible (the sample program is attached). I've discovered that this happens because the 'origin' and 'dimension' of such a widget where not properly initialized. My fix is the least intrusive change I've found to deal with that.

**#12 - 03/09/2015 02:25 PM - Constantin Asofiei**

Igor, I think this is part of a larger defect in P2J, related to dynamic widgets/frames.

Something I think should be common to both static and dynamic frames is their size: this can't change if the frame's content gets changed (i.e. widgets are added, removed, etc).

Look for a case like this:
1. create a static frame with some widgets
2. view the frame (so its dimension is computed)
3. create a dynamic widget
4. set its width to be greater than the frame's width (via the WIDTH-CHARS attribute)
5. add this widget to the frame
6. view the frame again - see if it's width changes

In 4GL, the frame dimension should remain the same.  This I think is broken in P2J with your change: when adding a new widget, you force the frame to recompute its width/height.

**#13 - 03/09/2015 02:39 PM - Igor Skornyakov**

Constantin Asofiei wrote:

> Igor, I think this is part of a larger defect in P2J, related to dynamic widgets/frames.
>
> Something I think should be common to both static and dynamic frames is their size: this can't change if the frame's content gets changed (i.e. widgets are added, removed, etc).
>
> Look for a case like this:
> 1. create a static frame with some widgets
> 2. view the frame (so its dimension is computed)
> 3. create a dynamic widget
> 4. set its width to be greater than the frame's width (via the WIDTH-CHARS attribute)
> 5. add this widget to the frame
> 6. view the frame again - see if it's width changes

> In 4GL, the frame dimension should remain the same.  This I think is broken in P2J with your change: when adding a new widget, you force the frame to recompute its width/height.

I see. Thank you Constantin. I will rework my fix. However as far as I understand the frame **can be** resized after adding/removing a widget if its AUTO-RESIZE attribute is TRUE. Am I right?

#### #14 - 03/09/2015 02:44 PM - Constantin Asofiei

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
>> Igor, I think this is part of a larger defect in P2J, related to dynamic widgets/frames.
>>
>> Something I think should be common to both static and dynamic frames is their size: this can't change if the frame's content gets changed (i.e. widgets are added, removed, etc).
>>
>> Look for a case like this:
>> 1. create a static frame with some widgets
>> 2. view the frame (so its dimension is computed)
>> 3. create a dynamic widget
>> 4. set its width to be greater than the frame's width (via the WIDTH-CHARS attribute)
>> 5. add this widget to the frame
>> 6. view the frame again - see if it's width changes
>>
>> In 4GL, the frame dimension should remain the same.  This I think is broken in P2J with your change: when adding a new widget, you force the frame to recompute its width/height.
>
> I see. Thank you Constantin. I will rework my fix. However as far as I understand the frame **can be** resized after adding/removing a widget if its AUTO-RESIZE attribute is TRUE. Am I right?

As I see from dvref.pdf, AUTO-RESIZE is for GUI only.

**#15 - 03/09/2015 02:48 PM - Igor Skornyakov**

Constantin Asofiei wrote:

> As I see from dvref.pdf, AUTO-RESIZE is for GUI only.

I know. However in my tests this attribute had an effect in ChUI as well. I will double check.

**#16 - 03/09/2015 02:57 PM - Constantin Asofiei**

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
>> As I see from dvref.pdf, AUTO-RESIZE is for GUI only.
>
> I know. However in my tests this attribute had an effect in ChUI as well. I will double check.

In any case, this is not currently supported by P2J.  The APIs are a no-op now.

**#17 - 03/09/2015 03:10 PM - Igor Skornyakov**

Constantin Asofiei wrote:

> In any case, this is not currently supported by P2J.  The APIs are a no-op now.

I see. Should I leave it as-is if the real behavior of 4GL is different?

**#18 - 03/09/2015 03:14 PM - Constantin Asofiei**

Igor Skornyakov wrote:

> Constantin Asofiei wrote:

> > In any case, this is not currently supported by P2J.  The APIs are a no-op now.
>
> I see. Should I leave it as-is if the real behavior of 4GL is different?

Yes, don't complicate things for now.  Just confirm the implications of your change and how 4GL behaves when the frame content no longer fits in the expected frame dimension.

#### #19 - 03/09/2015 03:21 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Yes, don't complicate things for now.  Just confirm the implications of your change and how 4GL behaves when the frame content no longer fits in the expected frame dimension.

Sorry, I do not understand. What if the 4GL behavior is different for different values of the AUTO-RESIZE attribute as it seems to be? Adding support for the attribute is not a big deal.

#### #20 - 03/09/2015 03:30 PM - Constantin Asofiei

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
> > Yes, don't complicate things for now.  Just confirm the implications of your change and how 4GL behaves when the frame content no longer fits in the expected frame dimension.
>
> Sorry, I do not understand. What if the 4GL behavior is different for different values of the AUTO-RESIZE attribute as it seems to be? Adding support for the attribute is not a big deal.

The urgency now is to determine if your Frame.addWidget change does not break any existing P2J functionalities.  As the runtime testing doesn't cover all possible scenarios, there may be existing, working features, which are now broken by this change.  Focus on what I mentioned above: check how frame dimension reacts after a new widget is added, and this widget does not fit the current frame size.

As a side-note, the frame size can be computed in two modes:
- explicitly, via the SIZE phrase or sizing attributes
- implicitly, when the frame size is computed from the widget contents.

#### #21 - 03/09/2015 03:33 PM - Igor Skornyakov

Constantin Asofiei wrote:

> The urgency now is to determine if your Frame.addWidget change does not break any existing P2J functionalities. As the runtime testing doesn't cover all possible scenarios, there may be existing, working features, which are now broken by this change. Focus on what I mentioned above: check how frame dimension reacts after a new widget is added, and this widget does not fit the current frame size.
>
> As a side-note, the frame size can be computed in two modes:
> - explicitly, via the SIZE phrase or sizing attributes
> - implicitly, when the frame size is computed from the widget contents.

OK. I will do this. Thank you for the clarification.

#### #22 - 03/09/2015 04:43 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, I think this is part of a larger defect in P2J, related to dynamic widgets/frames.
>
> Something I think should be common to both static and dynamic frames is their size: this can't change if the frame's content gets changed (i.e. widgets are added, removed, etc).
>
> Look for a case like this:
> 1. create a static frame with some widgets
> 2. view the frame (so its dimension is computed)
> 3. create a dynamic widget
> 4. set its width to be greater than the frame's width (via the WIDTH-CHARS attribute)
> 5. add this widget to the frame
> 6. view the frame again - see if it's width changes
>
> In 4GL, the frame dimension should remain the same. This I think is broken in P2J with your change: when adding a new widget, you force the frame to recompute its width/height.

Constantin,
You're right. In your scenario the frame size will not change and the added widget is invisible. I will think how to change my fix to retain compatibility.

**#23 - 03/10/2015 05:56 AM - Igor Skornyakov**

*- File w3x.p added*

Constantin,
If I correctly understand your scenario my change retains the 4GL compatibility: after the frame is enabled adding the dynamic widget doesn't change its size both in 4GL and Java runtime. Moreover in both environments the added widget is invisible regardless of its size. Please take a look at the attached 4GL program. Is it what you had in mind?
Thank you.

**#24 - 03/10/2015 06:21 AM - Constantin Asofiei**

Igor, see this test:

```
def var hf as handle.

def var i as int.

form i with frame f1.
display i with frame f1.

create fill-in hf.
hf:row = 1.
hf:column = 1.
hf:width-chars = 2.
hf:visible = true.
hf:frame = frame f1:handle.

view frame f1.
```

With your change, P2J abends; without your change, P2J does not abend.

I think the savedWidgets need to be updated with the new widget, when addWidget is called for a dynamic widget.

**#25 - 03/10/2015 06:26 AM - Igor Skornyakov**

I see. Thank you Constantin.

**#26 - 03/10/2015 05:10 PM - Igor Skornyakov**

The updating of savedWidgets doesn't work. Continue the investigation.

**#27 - 03/11/2015 11:55 AM - Igor Skornyakov**

With the following update in Frame

```
   public void addWidget(Widget<O> widget)
   {
      contentPane.add(widget);
      if (savedWidgets != null)
      {
         WidgetItem<O>[] widgets = Arrays.copyOf(savedWidgets, savedWidgets.length+1);
         widgets[savedWidgets.length] = new WidgetItem<O>(widget);
         savedWidgets = widgets;
      }
      if (widget instanceof Editor)
      {
         withEditors = true;
      }
   }
```

the dynamic list widgets become visible and Constantin's sample doesn't crash. However the look of this example at runtime is different from 4GL: an additional empty line appears after the "i" label. It seems however that this is a separate issue.

### #28 - 03/11/2015 12:35 PM - Greg Shah

This looks reasonable. Please add this to the changes for #2522 to reduce testing effort. Also, it will give Constantin some time to finalize his update that also changes the Frame class. This means you should expect to merge your code before you run testing.

### #29 - 03/13/2015 12:33 PM - Igor Skornyakov

Final version has passed regression testing and is committed as bzr rev. 10811

### #30 - 03/16/2015 09:00 AM - Greg Shah

*- Status changed from New to Closed*

### #31 - 11/16/2016 12:13 PM - Greg Shah

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

## Files

| | | | |
|---|---|---|---|
| ias_upd20150307a.zip | 43.5 KB | 03/07/2015 | Igor Skornyakov |
| w3x.p | 4.76 KB | 03/09/2015 | Igor Skornyakov |
| w3x.p | 4.76 KB | 03/10/2015 | Igor Skornyakov |