

## User Interface - Feature #4164

### implement misc UI features (frame options, misc attributes)

08/05/2019 08:17 PM - Greg Shah

<b>Status:</b>	WIP	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Stanislav Lomany	<b>% Done:</b>	70%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			
<b>Related issues:</b>			
Related to User Interface - Feature #4394: add some frame options and attributes		<b>Closed</b>	
Related to User Interface - Bug #4625: Frame configuration not shared or sync...		<b>New</b>	

## History

### #1 - 08/05/2019 08:20 PM - Greg Shah

- test the FRAME\_COL function to ensure it is compatible
- frame options
  - SCROLL (runtime stubs)
  - USE\_TEXT (runtime stubs)
  - NO-ATTR-SPACE (partial runtime)
- attributes
  - browse:num\_entries
  - editor:auto\_indent
  - widget:next\_tab\_item
  - frame:prev\_tab\_item
- event support
  - SCROLL-NOTIFY - high level events for browse widget
    - "A mouse action in the scrollbar area of a browse."
    - "This event allows the developer to track physical movement of the focused row in the browse viewport."
    - "This event is supported only for backward compatibility. Use SCROLL-HORIZONTAL and SCROLL-VERTICAL instead."
  - SCROLL-HORIZONTAL and SCROLL-VERTICAL - high level events for browse widget
  - OFF-END - high level event for browse widget, "A keyboard or mouse action that tries to move after the last row of a browse."
  - OFF-HOME - high level event for browse widget, "A keyboard or mouse action that tries to move before the first row of a browse."
  - PARENT-WINDOW-CLOSE - high level event for window widget, "An event that each descendant window receives when the common ancestor window in that family receives a WINDOW-CLOSE event."
  - DESELECT
    - not sure if this is the same as the DESELECTION event documented as a direct manipulation event, there is no other documented event that can match
    - How it is generated? "For all selected widgets in a frame, click the mouse SELECT button on an unselected widget or in empty space in the frame."
    - Affects "Frame and field-level widgets with SELECTABLE attribute set to TRUE; browses. For a single selected EXTEND button on a widget, click the mouse selected widget."
    - Processing
      - "Internal: Sets the widget's SELECTED attribute to FALSE. This setting takes effect after any trigger for the event executes."
      - "Screen: Removes the highlight from the affected widget or widgets."

### #3 - 02/13/2020 11:39 AM - Greg Shah

- Status changed from New to WIP

- Assignee set to Greg Shah

I've added testcases/uast/browse/browse\_num\_entries.p which shows how num-entries works:

```
def temp-table test no-undo
  field txt    as char
  field num    as int
  field flag   as logical
  index pi_num is primary num ascending.

def var last-num as int init 30 no-undo.
def var i        as int.

message "How many records?" update last-num format "999".

do i = 1 to last-num
  transaction:

  create test.
  assign txt = string(i) + " text"
         num = i
         flag = (i mod 2) eq 0.

end.

def query q for test.
def browse brws query q no-lock no-wait
  display test
  with 5 down centered.
form brws with frame f-frame.

/* 0 */
message "Before OPEN QUERY" browse brws:num-entries.

open query q
  for each test no-lock.

/* 0 */
message "After OPEN QUERY" browse brws:num-entries.

on f7 of browse brws
do:
  /* this is an undocumented attribute that works on browse */
  /* the value depends on the number of records in the query AND on the down value of the browse */
  /* if query_record_count < browse_down then query_record_count else browse_down */
  /* perhaps this is best described as "returns the number of records that are visible at this time" */
  message "F7" browse brws:num-entries.
end.

enable all with frame f-frame.

wait-for go of current-window.

disable all with frame f-frame.
hide frame f-frame.

/* shows same as F7 case */
message "After HIDE" browse brws:num-entries.
```

**#4 - 02/13/2020 05:21 PM - Greg Shah**

Branch 4335a revision 11420 implements the runtime for the BROWSE:NUM-ENTRIES undocumented attribute. The conversion was already supported. The testcase from [#4164-3](#) is checked in as `uast/testcases/browse/browse_num_entries.p`. That testcase works properly with the new implementation.

Stanislav: Please review.

**#5 - 02/14/2020 10:42 AM - Stanislav Lomany**

- File `browse_num_entries2.p` added

Stanislav: Please review.

The approach in `getNumEntries` doesn't work if a browse is not fully filled and the number of rows is more than DOWN, see the testcase. I suppose the only way to get the correct number of rows is to query the client side.

**#6 - 02/20/2020 12:15 PM - Greg Shah**

Stanislav Lomany wrote:

Stanislav: Please review.

The approach in `getNumEntries` doesn't work if a browse is not fully filled and the number of rows is more than DOWN, see the testcase. I suppose the only way to get the correct number of rows is to query the client side.

Do we already have an API for this? If not, do we store this value on the client?

**#7 - 02/20/2020 02:50 PM - Greg Shah**

4335a revision 11428 adds conversion and server-side support for EDITOR:AUTO-INDENT. The client-side runtime still needs to be implemented. I think the code needs to be changed in `com.goldencode.p2j.ui.client.Editor` (see both `placeLF()` and `splitLine()`).

Interestingly, TABs are used for focus change and they seem to be eaten, even if there is no other widget in the tab order (when the editor is the only enabled widget). So it looks like only space chars need to be considered. We should test if there are any other whitespace chars that are honored (either by typing the characters or from the clipboard or reading from file or variable).

The behavior is pretty simple, it works no matter if you are splitting a line or if you are adding a line by hitting enter at the end of a line. It works anywhere there is a preceding line.

**#8 - 02/20/2020 03:08 PM - Greg Shah**

We have usage of both NEXT-TAB-ITEM and PREV-TAB-ITEM in a customer application. Frames are used as well as widgets. Our gap analysis marks both attributes as `rt_lvl_partial` (conversion is fully supported already). The comment (at least for NEXT-TAB-ITEM) is that frames aren't supported. But as far as I can see the support seems to be there for both widgets and frames.

Can someone please provide more detail on the parts that are missing in regard to NEXT-TAB-ITEM and PREV-TAB-ITEM for frames?

**#9 - 02/20/2020 03:15 PM - Greg Shah**

- % Done changed from 0 to 20

There is no further conversion work in this task. Everything is just runtime at this point.

**#10 - 02/20/2020 03:18 PM - Stanislav Lomany**

- % Done changed from 20 to 0

Do we already have an API for this? If not, do we store this value on the client?

There's no server API for it, you can create a function to get `Browse.getCacheSize()` from the client side.

**#11 - 02/20/2020 03:20 PM - Stanislav Lomany**

- % Done changed from 0 to 20

**#12 - 02/21/2020 07:07 AM - Constantin Asofiei**

Greg Shah wrote:

Can someone please provide more detail on the parts that are missing in regard to NEXT-TAB-ITEM and PREV-TAB-ITEM for frames?

A part which is missing is the `GenericFrame` support - something like `frame f1:next-tab-item` will not work, unless all access to these attrs is converted via the frame handle.

**#13 - 02/21/2020 07:20 AM - Greg Shah**

Is this just a conversion issue? The methods are implemented in `CommonFrame` and `FrameWidget`.

**#14 - 02/21/2020 07:34 AM - Constantin Asofiei**

Greg Shah wrote:

Is this just a conversion issue? The methods are implemented in CommonFrame and FrameWidget.

GenericFrame throws an exception for this - the fix is simple, just delegate the call to FrameWidget.

**#15 - 03/13/2020 03:45 PM - Greg Shah**

Task branch 4335a was merged to trunk as revision 11345.

**#16 - 03/18/2020 06:51 PM - Roger Borrello**

Greg Shah wrote:

Stanislav Lomany wrote:

Stanislav: Please review.

The approach in getNumEntries doesn't work if a browse is not fully filled and the number of rows is more than DOWN, see the testcase. I suppose the only way to get the correct number of rows is to query the client side.

Do we already have an API for this? If not, do we store this value on the client?

What is NUM-ENTRIES attribute hold? Is it the number of entries left? When I ran the browse\_num\_entries2.p in 4GL, 3 was displayed. In FWD, is displayed 11.

**#17 - 03/18/2020 07:01 PM - Stanislav Lomany**

What is NUM-ENTRIES attribute hold? Is it the number of entries left?

It is the number of data rows in the view.

**#18 - 03/19/2020 08:39 AM - Greg Shah**

Read the comments in browse\_num\_entries.p and look carefully at the current FWD implementation/comments. Make sure you read [#4164-3](#) through [#4164-6](#) and also [#4164-10](#).

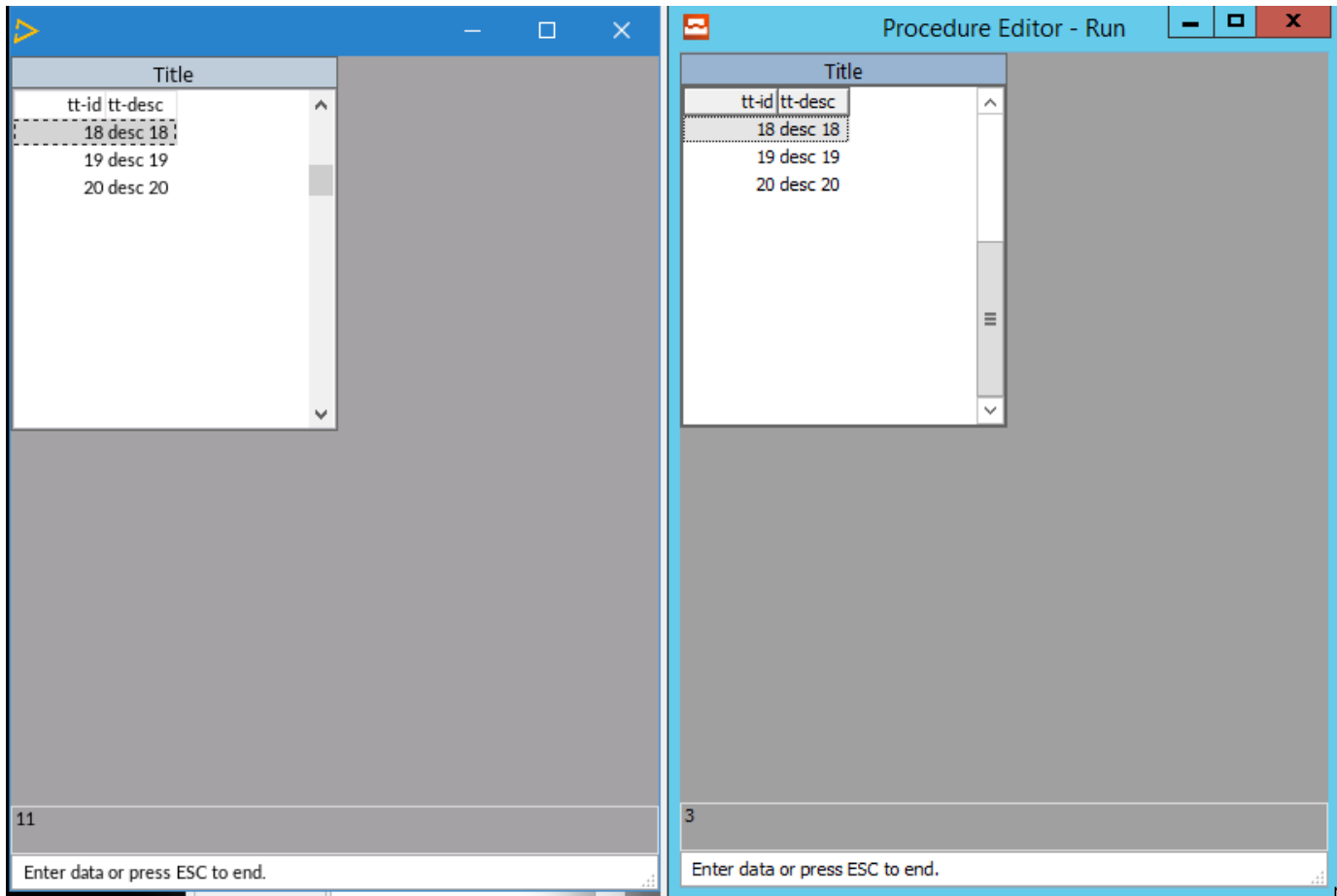
BTW, I've already implemented the NEXT-TAB-ITEM and PREV-TAB-ITEM change, but just haven't checked it in yet.

**#19 - 03/19/2020 09:16 AM - Roger Borrello**

- File num\_entries.png added

- File num\_entries2.png added

FWD returning 11 leads me to believe there are still 11 rows in the view. Evidence is the slider on the scrollbar has a large proportion of scrollbar below, while the 4GL the slider (in addition to filling the bottom of the scrollbar) is at the bottom.



When you click above the slider, there are 3 clicks to get to the top. Once it is at the top, 1 click below the slider takes you all the way to the bottom of the view, and F7=11 is then accurate, as there are 11 lines in the view. 4GL has no such F7 function.

Title	
tt-id	tt-desc
10	desc 10
11	desc 11
12	desc 12
13	desc 13
14	desc 14
15	desc 15
16	desc 16
17	desc 17
18	desc 18
19	desc 19
20	desc 20

11  
F7 11

Enter data or press ESC to end.

Title	
tt-id	tt-desc
11	desc 11
12	desc 12
13	desc 13
14	desc 14
15	desc 15
16	desc 16
17	desc 17
18	desc 18
19	desc 19
20	desc 20

3

Enter data or press ESC to end.

How should I approach this?

**#20 - 03/19/2020 09:33 AM - Greg Shah**

As Stanislav noted above, FWD returning 11 is a BUG. So you don't need to analyze how 11 relates to anything.

4GL has no such F7 function.

You must be testing the wrong code in the 4GL. The "F7 function" that you mention is just a trigger in the 4GL code. It should be there in both FWD and the 4GL.

Please read the testcases and the previous notes carefully. Stanislav has already provided the details on what needs to be done in [#4164-10](#).

**#21 - 03/19/2020 02:24 PM - Roger Borrello**

Stanislav Lomany wrote:

Do we already have an API for this? If not, do we store this value on the client?

There's no server API for it, you can create a function to get `Browse.getCacheSize()` from the client side.

Is there already some data sharing from client to server going on, so I can see an example? Is `LogicalTerminal's refreshCurrentRow` implementation of `ClientExport's refreshCurrentRow` doing the same type of servicing?

**#22 - 03/19/2020 02:33 PM - Greg Shah**

Yes, except you will only have the `browseld` parameter and you must return an `int`.

**#23 - 03/19/2020 03:28 PM - Roger Borrello**

Greg Shah wrote:

Yes, except you will only have the `browseld` parameter and you must return an `int`.

What's the relationship between the `PageRowCount` (number of rows in the browse viewport set either by `DOWN` or `SIZE` options) and the `CacheSize` (number of entries visible in the browse)? They sound very similar.



**#24 - 03/19/2020 03:32 PM - Stanislav Lomany**

Consider you have a browse:

```
data row 1
data row 2
<empty row>
<empty row>
<empty row>
```

DOWN is 5, cache size is 2.

**#25 - 03/19/2020 05:04 PM - Roger Borrello**

Stanislav Lomany wrote:

Consider you have a browse:

[...]

DOWN is 5, cache size is 2.

Thanks.

I've added on the server:

- ClientExports: public int getBrowseCacheSize(int browseld);
- BrowseWidget: private int getBrowseCacheSize() Currently I have this storing LogicalTerminal.getBrowseCacheSize(this) in private Integer cacheSize
- LogicalTerminal: public static int getBrowseCacheSize(BrowseWidget bw) which returns locate().client.getBrowseCacheSize(bw.getId());

On the client:

- ThinClient: public int getBrowseCacheSize(int browseld) where this is returning eventDrawingBracket(browse, () -> res[0] = browse.getCacheSize());

Does that look accurate?

I see that on the client, the value returned by Browse.getCacheSize is dcData.length. In order to make this available to the server,

**#26 - 03/19/2020 05:22 PM - Stanislav Lomany**

- ClientExports: public int getBrowseCacheSize(int browseId);
- BrowseWidget: private int getBrowseCacheSize()
- LogicalTerminal: public static int getBrowseCacheSize(BrowseWidget bw)

I would prefer to name functions in other way, e.g. getBrowseRowsNum, because the "cache" term is internal to client-side browse part.

```
eventDrawingBracket(browse, () -> res[0] = browse.getCacheSize());
```

You don't need to start an eventDrawingBracket because no drawing happens here.

**#27 - 03/19/2020 05:27 PM - Greg Shah**

I don't see any value in BrowseWidget.getBrowseCacheSize() when getNumEntries() can call LogicalTerminal.getBrowseCacheSize() directly.

**#28 - 03/19/2020 06:16 PM - Roger Borrello**

Updated per suggestions. Thanks!

```
Committing to: /home/rfb/secure/code/p2j_repo/p2j/active/4231b/
```

```
modified src/com/goldencode/p2j/ui/chui/ThinClient.java
modified src/com/goldencode/p2j/ui/ClientExports.java
modified src/com/goldencode/p2j/ui/BrowseWidget.java
modified src/com/goldencode/p2j/ui/LogicalTerminal.java
Committed revision 11377.
```

**#29 - 03/20/2020 04:32 AM - Stanislav Lomany**

Review rev 11377: Roger, you forgot to remove the array need for eventDrawingBracket. You can just write return browse != null ? browse.getCacheSize() : 0; Otherwise the update is good.

**#30 - 03/20/2020 09:01 AM - Roger Borrello**

Stanislav Lomany wrote:

Review rev 11377: Roger, you forgot to remove the array need for eventDrawingBracket. You can just write return browse != null ?

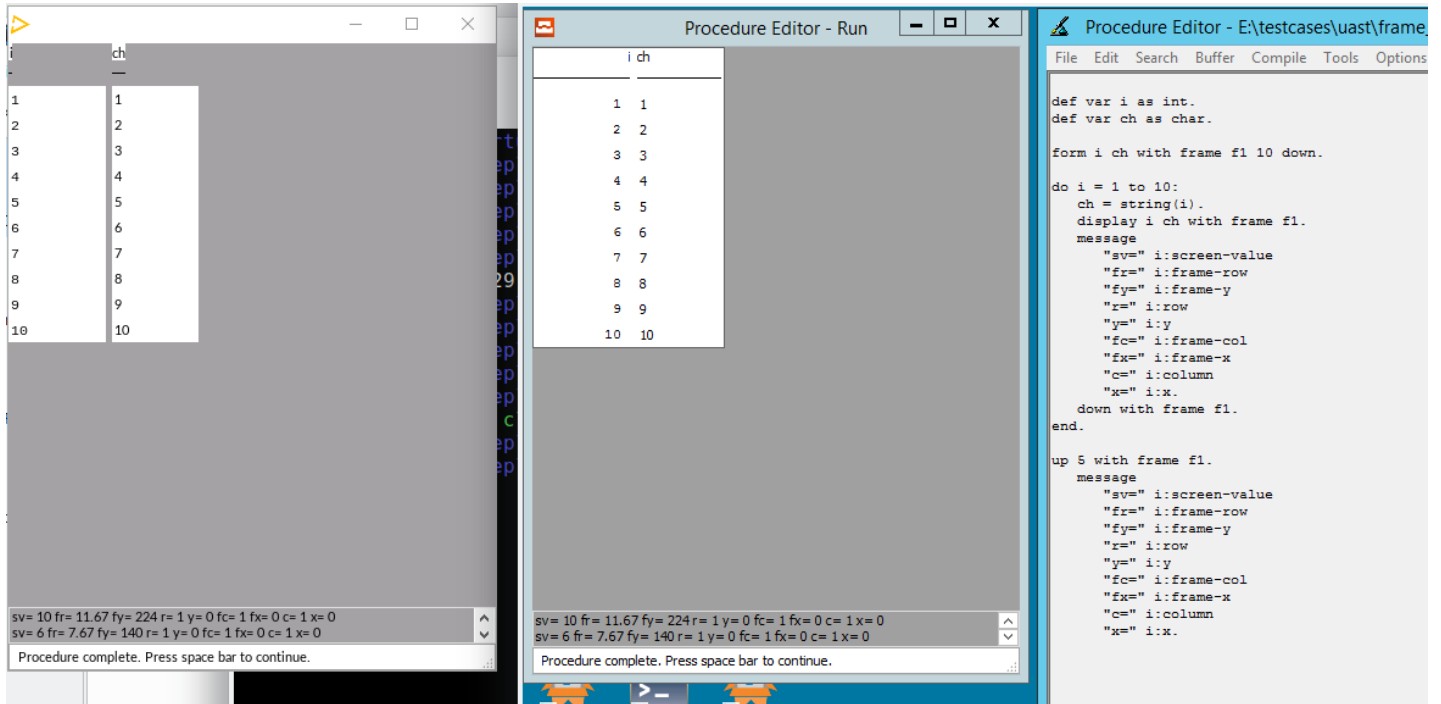
browse.getCacheSize() : 0; Otherwise the update is good.

Thanks! I hadn't worked with eventDrawingBracket so I was being conservative.

### #31 - 03/20/2020 11:10 AM - Roger Borrello

- File frame\_loc\_attributes\_get.png added

This test is promising:



Code below. What else can I do to validate?

```
def var i as int.  
def var ch as char.  
  
form i ch with frame f1 10 down.  
  
do i = 0 to 9:  
  ch = "#" + string(i).  
  display i ch with frame f1.  
  message  
    i:screen-value ":"  
    "x=" i:x          /* pix loc of wid left edge rel to left parent edge */  
    "y=" i:y          /* pix loc of wid top edge rel to top parent edge */  
    "c=" i:column     /* col pos of wid left edge rel to left (same as x) */  
    "r=" i:row        /* row pos of wid top edge rel to top (same as y) */  
    "fx=" i:frame-x  /* pix loc of wid left edge rel to ul of frame */  
    "fy=" i:frame-y  /* pix loc of wid top edge rel to ul of frame */  
    "fc=" i:frame-col /* Dec col pos of wid left edge rel to ul of wid frame */  
    "fr=" i:frame-row /* Dec col pos of wid top left edge rel to ul of wid frame */  
  .  
  down with frame f1.  
end.  
  
up 5 with frame f1.  
message  
  i:screen-value ":"
```

```

"x=" i:x /* pix loc of wid left edge rel to left parent edge */
"y=" i:y /* pix loc of wid top edge rel to top parent edge */
"c=" i:column /* col pos of wid left edge rel to left (same as x) */
"r=" i:row /* row pos of wid top edge rel to top (same as y) */
"fx=" i:frame-x /* pix loc of wid left edge rel to ul of frame */
"fy=" i:frame-y /* pix loc of wid top edge rel to ul of frame */
"fc=" i:frame-col /* Dec col pos of wid left edge rel to ul of wid frame */
"fr=" i:frame-row /* Dec col pos of wid top left edge rel to ul of wid frame */
.

```

### #32 - 03/20/2020 05:17 PM - Greg Shah

Please read the 4GL docs to see what the FRAME-COL function is meant to do. It is the counterpart of FRAME-ROW and both are used to see the location in (characters) of the frame's upper left corner. So in order to test this, you probably need to use it on frames that are displayed at different locations on the screen, right? Also, what happens when the frame is not displayed at all?

### #33 - 03/26/2020 10:17 AM - Greg Shah

Branch 4231b revision 11395 includes GenericFrame changes that complete next-tab-item and prev-tab-item per [#4164-12](#). Those items are now done (and the gap marking was also updated accordingly).

I've updated [#4164-1](#) to highlight the remaining work.

### #34 - 03/31/2020 10:59 AM - Greg Shah

- Related to Feature #4394: add some frame options and attributes added

### #35 - 04/03/2020 08:53 AM - Greg Shah

All you care about is positioning the frame in different locations. This is trivial to do in the AT phrase of the frame phrase.

```
WITH frame f AT { COLUMN column ROW row | X x Y y }
```

### #36 - 04/03/2020 09:34 AM - Roger Borrello

I had a testcase:

```

def var i as int init 0 format 99.

def frame f0 i with width 10 title "F0 Title".
def frame f1 i with width 20 title "F1 Title".
def frame f2 i with width 30 title "F2 Title".
def frame f3 i with width 40 title "F3 Title".
def frame f4 i with width 50 title "F4 Title".
def frame f5 i with width 40 title "F5 Title".
def frame df0 i with view-as dialog-box width 50 title "Default Dialog Title".

message "display-type=" session:display-type.

{frame_layout\frame_loc_attributes_helper.i &fr="f0"}
{frame_layout\frame_loc_attributes_helper.i &fr="f1"}
{frame_layout\frame_loc_attributes_helper.i &fr="f2"}
{frame_layout\frame_loc_attributes_helper.i &fr="f3"}

```

```
{frame_layout\frame_loc_attributes_helper.i &fr="f4"}
{frame_layout\frame_loc_attributes_helper.i &fr="f5"}
{frame_layout\frame_loc_attributes_helper.i &fr="df0"}
```

The helper:

```
view frame {&fr}.
message
  "{&fr}: "
  "x=" i:x          /* pix loc of wid left edge rel to left parent edge */
  "y=" i:y          /* pix loc of wid top edge rel to top parent edge */
  "c=" i:column     /* col pos of wid left edge rel to left (same as x) */
  "r=" i:row        /* row pos of wid top edge rel to top (same as y) */
  "fx=" i:frame-x  /* pix loc of wid left edge rel to ul of frame */
  "fy=" i:frame-y  /* pix loc of wid top edge rel to ul of frame */
  "fc=" i:frame-col /* Dec col pos of wid left edge rel to ul of wid frame */
  "fr=" i:frame-row /* Dec col pos of wid top left edge rel to ul of wid frame */
  .
wait-for go of frame {&fr}.
```

I should just be able to update the frame phrase with AT column x row y like:

```
def frame f0 i with width 10 title "F0 Title" at column 0 row 0.
def frame f1 i with width 10 title "F0 Title" at column 1 row 10.
def frame f2 i with width 10 title "F0 Title" at column 2 row 20.
...
```

**#37 - 04/03/2020 09:54 AM - Roger Borrello**

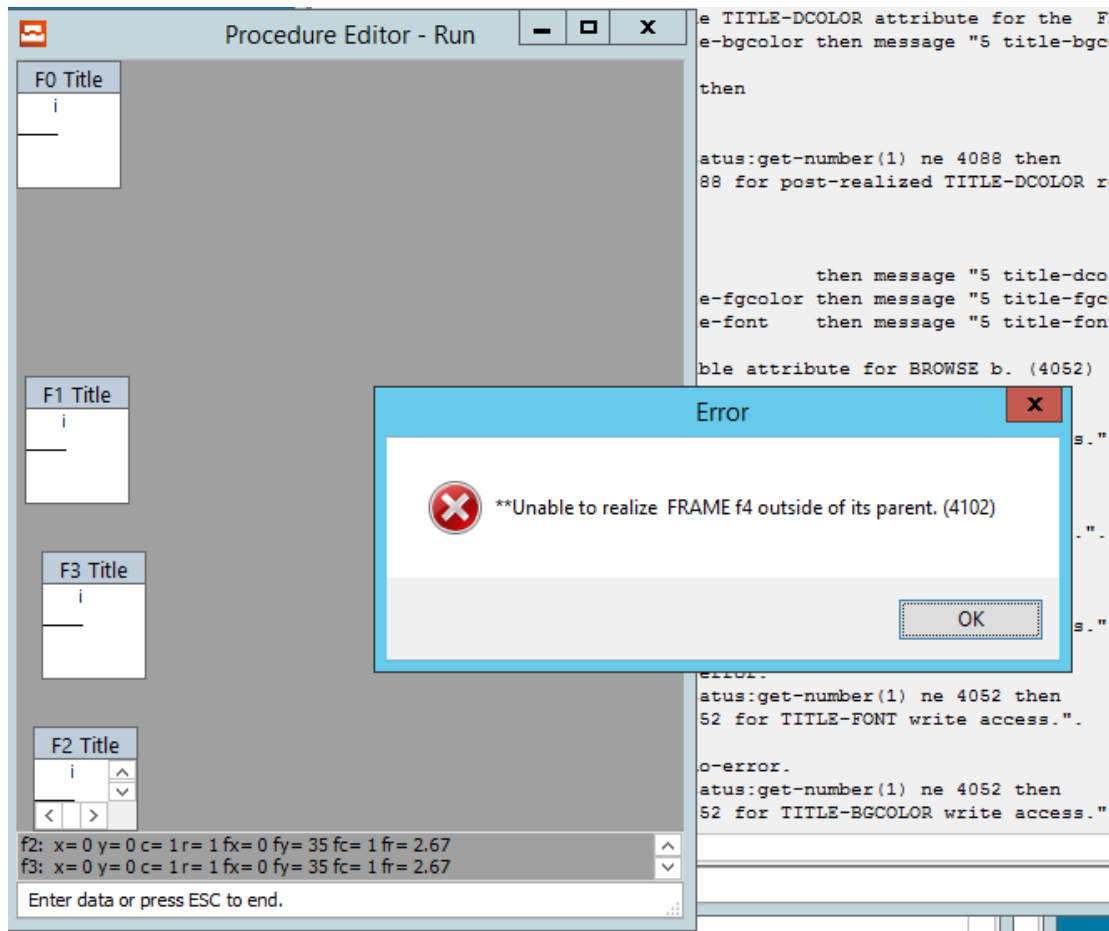
- File `frame_loc_attributes_get2.png` added

Greg Shah wrote:

All you care about is positioning the frame in different locations.

That is not really correct, because the attribute is off the widget, in this case i. See my new output... even though the frames are all over the place, the widget is still in the same location relative to its frame.

So my original thought, to find and position widgets all over the frame, would be a more valid test.



**#38 - 04/03/2020 10:16 AM - Greg Shah**

That is not really correct, because the attribute is off the widget, in this case i.

Please read [#4164-1](#) more carefully. FRAME-COL(frame) is a **built-in function NOT a widget attribute**. Your test is written wrong.

In regard to the error 4102, pick frame positions that are inside the same Window.

**#39 - 04/03/2020 10:38 AM - Roger Borrello**

Greg Shah wrote:

That is not really correct, because the attribute is off the widget, in this case i.

Please read [#4164-1](#) more carefully. FRAME-COL(frame) is a **built-in function NOT a widget attribute**. Your test is written wrong.

In regard to the error 4102, pick frame positions that are inside the same Window.

My bad... I will have results after conversion of the testcase.

```
view frame {&fr}.
/* Returns a DECIMAL value that is the column position of the left corner of a frame within its window. */
message "{&fr}=" frame-col({&fr}).
wait-for go of frame {&fr}.
```

**#40 - 04/03/2020 11:19 AM - Greg Shah**

I think you should check the FRAME-ROW(frame) builtin for all of the same case.

Make sure to test:

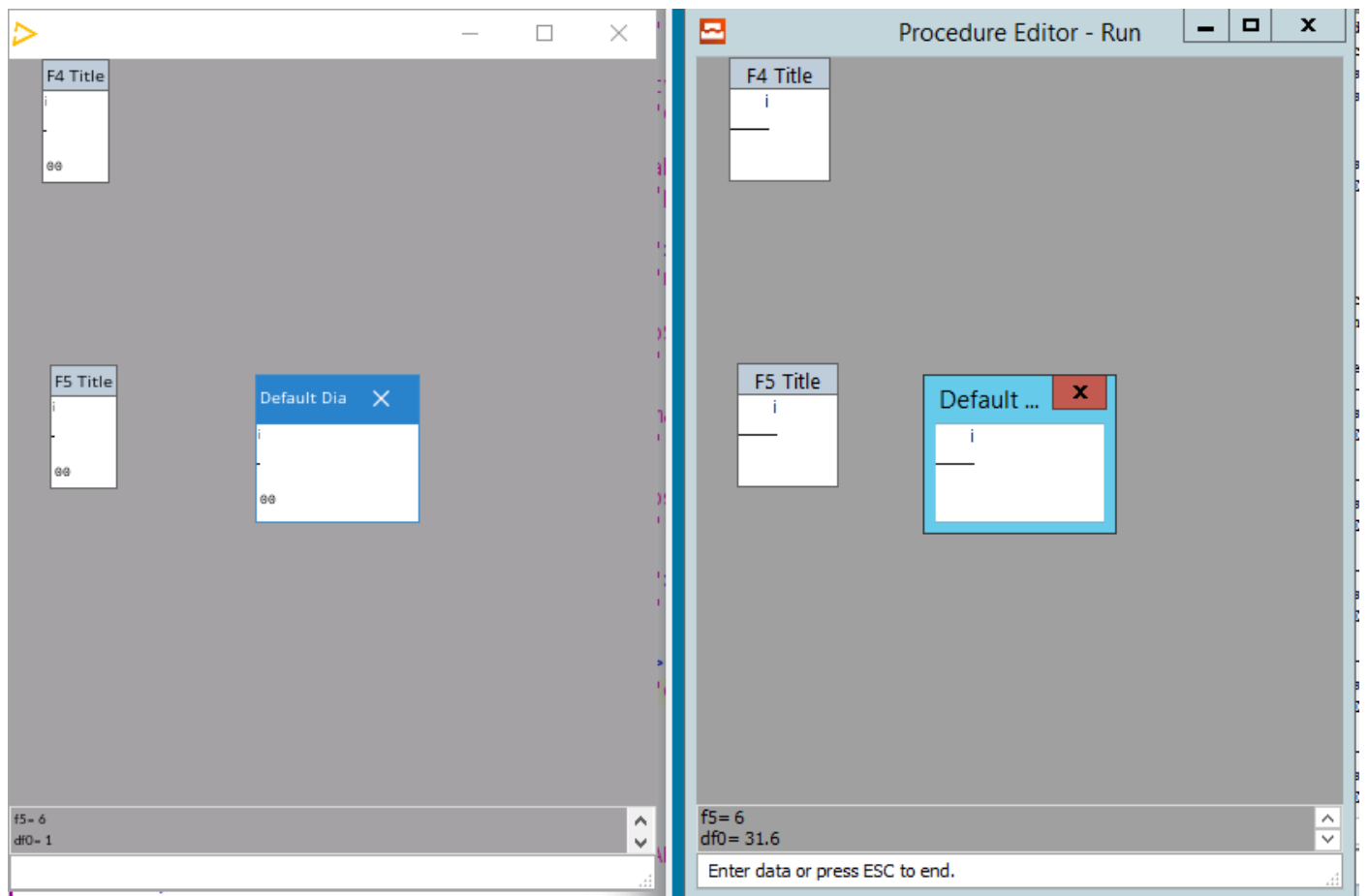
- What do these return before the frame is realized (made visible)? Is it different from post-realization?
- Can you set negative values or values outside the window for the frame, when the frame is not realized? If so what is returned for those cases?
- What happens if you use the same values for two frames that share the same window? I suspect the pre-realized/post-realized values might be different. For example, the 4GL may move the frame automatically.
- what happens when you display a frame that has no explicit positioning, but there are already other frames displayed? I think the location will be calculated automatically and will be different pre/post realization.

**#41 - 04/03/2020 12:33 PM - Roger Borrello**

- File `GUI_frame_loc_attributes_get2.png` added

I'll update, now that I (finally) understand. I do have some initial items, some not related to this testcase, but behaviors that are different from 4GL's. How do you want those reported?

With respect to what happens when you display a frame that has no explicit positioning, but there are already other frames displayed? This was covered in the current testcase's dialog frame `df5`. In 4GL it is reported as 31.6. In FWD, it was 1.





#### #42 - 04/03/2020 01:00 PM - Greg Shah

Make a list of the problems here and fix them.

#### #43 - 04/06/2020 04:33 PM - Roger Borrello

Greg Shah wrote:

Make a list of the problems here and fix them.

1. What do these return before the frame is realized (made visible)? Is it different from post-realization?

4GL pre-realization values are always 0. FWD are always 1.

#### #44 - 04/09/2020 01:01 PM - Roger Borrello

New testcase has expanded my ability to validate values, but I'm not sure it's on the mark yet. I was expecting the frame to move around, but it stays at (1,1).

```
def var i as int init 0 format 99.
def var b4col as int.
def var b4row as int.

def frame f0 i with title "F0 Title" at column 1 row 1.

/*message "display-type=" session:display-type.*/
os-delete "x.txt".
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=1 &rowval=1}
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=10 &rowval=2}
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=2 &rowval=3}
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=10 &rowval=4}
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=1 &rowval=5}
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=10 &rowval=4}
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=2 &rowval=3}
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=10 &rowval=2}
{frame_layout\frame_loc_attributes_helper.i &out="x.txt" &fr="f0" &colval=1 &rowval=1}
```

The helper include:

```
/* Returns a DECIMAL value that is the column position of the left corner of a frame within its window. */
assign
  b4col = frame-col({&fr})
  b4row = frame-row({&fr})
.
display i with frame {&fr} at col {&colval} row {&rowval}.
output to {&out} append.
message
  "{&fr}: b4col/b4row="b4col "/" b4row
  "aftcol/aftrow="frame-col({&fr}) "/" frame-row({&fr})
  "col/row="{&colval} "/" {&rowval}
  " i:"
  "x/y=" i:x "/" i:y /* pix loc of wid left/top edge rel to left parent edge */
  "c/r=" i:column "/" i:row /* col/row pos of wid left/top edge rel to left/top (same as x/y) */
  "fc/fr=" i:frame-col "/" i:frame-row /* Dec col pos of wid left/top edge rel to ul of wid frame */
  "fx/fy=" i:frame-x "/" i:frame-y /* pix loc of wid left/top edge rel to ul of frame */
.
output close.
wait-for go of frame {&fr}.
```

**#45 - 04/09/2020 01:05 PM - Constantin Asofiei**

Roger, the AT COL ROW in the frame phrase (for DISPLAY) sets the frame's definition location, not at runtime.

To move it at runtime, use the frame's col/row attributes.

**#46 - 04/09/2020 02:24 PM - Greg Shah**

As Constantin notes, you are telling the code to only have the frame positioned in col 1 row 1 because only the FIRST use of that is honored. Since you only ever have one frame, it is always set to the same value.

Instead of changing the attributes, you can just remove the frame definition and then use DIFFERENT frame names in each usage of the include file.

**#47 - 04/21/2020 09:26 AM - Roger Borrello**

Please confirm editor:auto-indent already converts and just needs runtime support only (so I can bypass)?

**#48 - 04/21/2020 09:57 AM - Greg Shah**

Roger Borrello wrote:

Please confirm editor:auto-indent already converts and just needs runtime support only (so I can bypass)?

From [#4164-7](#):

4335a revision 11428 adds conversion and server-side support for EDITOR:AUTO-INDENT. The client-side runtime still needs to be implemented. I think the code needs to be changed in com.goldencode.p2j.ui.client.Editor (see both placeLF() and splitLine()).

**#49 - 04/21/2020 06:34 PM - Roger Borrello**

Greg Shah wrote:

4335a revision 11428 adds conversion and server-side support for EDITOR:AUTO-INDENT. The client-side runtime still needs to be implemented. I think the code needs to be changed in com.goldencode.p2j.ui.client.Editor (see both placeLF() and splitLine()).

Interestingly, TABs are used for focus change and they seem to be eaten, even if there is no other widget in the tab order (when the editor is the only enabled widget). So it looks like only space chars need to be considered. We should test if there are any other whitespace chars that are honored (either by typing the characters or from the clipboard or reading from file or variable).

The behavior is pretty simple, it works no matter if you are splitting a line or if you are adding a line by hitting enter at the end of a line. It works anywhere there is a preceding line.

Assuming uast/editor\_auto\_indent.p the correct testcase for EDITOR:AUTO-INDENT...

## #50 - 04/22/2020 11:44 AM - Roger Borrello

With respect to FRAME-COL (and FRAME-ROW), I am debugging why these are incorrect when checked using this code:

```
def frame f0    i with width 10 title "Fr f0" at col 1 row 1.
def frame f1    i with width 20 title "Fr f1" at col 1 row 10.

display i with frame f0.
message "aftcol/aftrow="frame-col(f0) "/" frame-row(f0).

display i with frame f1.
message "aftcol/aftrow="frame-col(f1) "/" frame-row(f1).
```

This displays:

```
aftcol/aftrow= 1 / 1
aftcol/aftrow= 0 / 0
```

When in 4GL it is:

```
aftcol/aftrow= 1 / 1
aftcol/aftrow= 1 / 10
```

These values are retrieved from ThinClient methods getFrameCol

```
public double getFrameCol(int frameId)
{
    Frame frame = getFrame(frameId);
    double result = -1;

    if (frame != null && frame.isVisible() && !frame.isRedirected())
    {
        result = ConfigHelper.getAssignedColumn(frame.config());
    }

    if (result == BaseConfig.INV_COORD)
    {
        result = 0;
    }

    return ++result;
}
```

and getFrameRow:

```
public double getFrameRow(int frameId)
{
    Frame frame = getFrame(frameId);
    double result = 0;

    if (frame != null && frame.isVisible())
    {
        result = frame.location().y + 1;
    }

    return result;
}
```

These both determine whether or not the frame is visible using frame.isVisible(). I need to understand a little more before I can fix this.

1. For frame f0, isVisible is valid after the DISPLAY is processed (as you'd expect). But for frame f1, it doesn't become true after the DISPLAY.
2. Why is FRAME-COL retrieved from the configuration, but FRAME-ROW retrieved from the actual coordinates?

3. Why does `getFrameCol` include `!frame.isRedirected()` and `getFrameRow` doesn't?

Since they return DECIMAL, I believe they should be the *actual* coordinates, rather than what is configured. Documentation says, "Returns a DECIMAL value that is the column (row position) of the left (upper-left) corner of a frame within its window."

**#51 - 04/22/2020 01:41 PM - Roger Borrello**

During debug, I found that when the visible flag is set prior to the frame being displayed, the `getFrame(frameId)` returns `FrameGuiImpl`. After the frame is displayed, it returns `FrameChuiImpl`.

During the `f0` frame handling, the visible flag is found to be true in the `FrameChuiImpl`, so its values for `FRAME-COL` and `FRAME-ROW` are returned correctly. But for all the other frames, it is left as false.

I am not familiar enough with why `FrameGuiImpl` is the frame class before the display, and `FrameChuiImpl` after, to know if that's working correctly. I do see when the frame is displayed, the visible in `FrameGuiImpl` is what is set.

**#52 - 04/22/2020 02:22 PM - Constantin Asofiei**

Roger Borrello wrote:

During debug, I found that when the visible flag is set prior to the frame being displayed, the `getFrame(frameId)` returns `FrameGuiImpl`. After the frame is displayed, it returns `FrameChuiImpl`.

During the `f0` frame handling, the visible flag is found to be true in the `FrameChuiImpl`, so its values for `FRAME-COL` and `FRAME-ROW` are returned correctly. But for all the other frames, it is left as false.

I am not familiar enough with why `FrameGuiImpl` is the frame class before the display, and `FrameChuiImpl` after, to know if that's working correctly. I do see when the frame is displayed, the visible in `FrameGuiImpl` is what is set.

Area you using streams in your tests, or the redirected terminal ( unnamed stream)? In GUI we have ChUI mode for 'redirected terminal' or stream output, and GUI mode otherwise.

Can you post a breakpoint in the c'tor and see who creates `FrameChuiImpl` (on client-side)?

**#53 - 04/22/2020 02:39 PM - Roger Borrello**

Constantin Asofiei wrote:

Area you using streams in your tests, or the redirected terminal ( unnamed stream)? In GUI we have ChUI mode for 'redirected terminal' or stream output, and GUI mode otherwise.

I am using "output to..." to collect the output. I can try without. However, I would think this is a bug.

Can you post a breakpoint in the c'tor and see who creates FrameChuiImpl (on client-side)?

I did, and it shows StreamableWidget.convertToStreamable -> FrameGuiImpl.convertToStreamable -> WidgetFactoryAdapter.ChuiWidgetFactory.create ...

**#54 - 04/22/2020 02:42 PM - Constantin Asofiei**

Roger Borrello wrote:

I am using "output to..." to collect the output. I can try without. However, I would think this is a bug.

OK, so we are in a 'normal' path for redirected terminal in GUI. Yes, this looks like a bug, the configuration should be shared (or at least sync'ed back) to the GUI frame.

But, in my experience, a frame used for reports (to redirected terminal/streams) will not be used in GUI, too. So, create a duplicate test for this issue and add a bug in the UI project.

**#55 - 04/22/2020 03:01 PM - Roger Borrello**

Constantin Asofiei wrote:

Roger Borrello wrote:

I am using "output to..." to collect the output. I can try without. However, I would think this is a bug.

OK, so we are in a 'normal' path for redirected terminal in GUI. Yes, this looks like a bug, the configuration should be shared (or at least sync'ed back) to the GUI frame.

But, in my experience, a frame used for reports (to redirected terminal/streams) will not be used in GUI, too. So, create a duplicate test for this issue and add a bug in the UI project.

This is "normal" in that it's easier for testing, but I would agree, not in "regular" normal.

Testcase:

```
uast/frame_layout/frame_loc_attributes_get2-redirected.p
uast/frame_layout/frame_loc_attributes_get2.p
uast/frame_layout/frame_loc_attributes_helper.i
```

The helper takes a pre-processor directive for a log file. If it's there, it redirects. If it's not, it doesn't redirect.

**#56 - 04/22/2020 03:03 PM - Roger Borrello**

- Related to Bug #4625: Frame configuration not shared or synched back to the GUI frame on redirection added

**#57 - 04/22/2020 03:39 PM - Roger Borrello**

I also notice on the last frame (a dialog box) the F2 never is recognized in the wait-for go of frame df0. I have to use ESC.

Wasn't someone working in the keystroke area of the client that may have changed something related to this?

**#58 - 04/22/2020 03:43 PM - Roger Borrello**

- % Done changed from 20 to 40

**#59 - 04/22/2020 03:45 PM - Roger Borrello**

When not using redirection, the testcase passes.

```
f0: b4col/b4row= 0 / 0 aftcol/aftrow= 1 / 1 col/row= 1 / 1 i: x/y= 0 / 0 c/r= 1 / 1 fc/fr= 1 / 2.67 fx/fy= 0 / 35
f1: b4col/b4row= 0 / 0 aftcol/aftrow= 1 / 10 col/row= 1 / 10 i: x/y= 0 / 0 c/r= 1 / 1 fc/fr= 1 / 2.67 fx/fy= 0 / 35
f2: b4col/b4row= 0 / 0 aftcol/aftrow= 10 / 1 col/row= 10 / 1 i: x/y= 0 / 0 c/r= 1 / 1 fc/fr= 1 / 2.67 fx/fy= 0 / 35
f3: b4col/b4row= 0 / 0 aftcol/aftrow= 10 / 10 col/row= 10 / 10 i: x/y= 0 / 0 c/r= 1 / 1 fc/fr= 1 / 2.67 fx/fy= 0 / 35
df0: b4col/b4row= 0 / 0 aftcol/aftrow= 1 / 1 col/row= 1 / 1 i: x/y= 0 / 0 c/r= 1 / 1 fc/fr= 1 / 2.67 fx/fy= 0 / 35
```

However, I'd still like to understand of setFrameCol and setFrameRow need to be merged to using the coordinates or the configuration.

**#60 - 04/29/2020 11:48 AM - Roger Borrello**

Hynek: I am trying to understand why the implementations for FRAME-COL and FRAME-ROW are so different, and Greg suggested asking you.

ThinClient.getFrameCol uses ConfigHelper.getAssignedColumn(frame.config()) while ThinClient.getFrameRow uses frame.location().y. The results are accurate either way, but was there any specific reason for the difference?

**#61 - 04/29/2020 12:12 PM - Hynek Cihlar**

Roger Borrello wrote:

Hynek: I am trying to understand why the implementations for FRAME-COL and FRAME-ROW are so different, and Greg suggested asking you.

ThinClient.getFrameCol uses ConfigHelper.getAssignedColumn(frame.config()) while ThinClient.getFrameRow uses frame.location().y. The results are accurate either way, but was there any specific reason for the difference?

Both of these methods were at some point changed by myself and by Constantin as part of bigger refactoring chunk. I don't think there is any valid reason for the different implementations. I don't even think we need to do a remote call to get the value of FRAME-ROW and FRAME-COL functions. Both could be determined from the frame's config using ConfigHelper.getEffectiveRow and ConfigHelper.getEffectiveColumn.

**#62 - 04/29/2020 12:36 PM - Roger Borrello**

Hynek Cihlar wrote:

Both of these methods were at some point changed by myself and by Constantin as part of bigger refactoring chunk. I don't think there is any valid reason for the different implementations. I don't even think we need to do a remote call to get the value of FRAME-ROW and FRAME-COL functions. Both could be determined from the frame's config using ConfigHelper.getEffectiveRow and ConfigHelper.getEffectiveColumn.

I will update ThinClient and run my testcases that verify the attributes, then submit for review.

**#63 - 04/29/2020 01:14 PM - Roger Borrello**

One other significant difference between the two is that getFrameCol includes a check for !frame.isRedirected() where getFrameRow does not. This condition of whether or not the col/row is retrieved. Which is more correct?

**#64 - 04/29/2020 01:27 PM - Hynek Cihlar**

Roger Borrello wrote:

One other significant difference between the two is that getFrameCol includes a check for !frame.isRedirected() where getFrameRow does not. This condition of whether or not the col/row is retrieved. Which is more correct?

Sorry, I can't answer this. But it should be easy to find out by testing in native 4GL.

**#65 - 04/29/2020 03:00 PM - Roger Borrello**

getEffectivexxx methods are returning 0.0 for clientColumn and clientRow when reading the attributes. The getAssignedxxx methods are returning the actual row/col accurately when I call them directly, because in the getEffectivexxx methods, they are protected by validating the row/col are BaseConfig.INV\_COORD before checking.

The testcases are getting frame-col(f0)/frame-row(f0) before/after realization. The frames are defined with at col x row y at the beginning.

It looks like the server-assigned values returned by getAssignedxxx are more accurate than getEffectivexxx (albeit 0-based). Does this warrant more investigation, or should I be OK using the getAssignedxxx methods?

**#66 - 04/29/2020 04:37 PM - Roger Borrello**

Updated src/com/goldencode/p2j/ui/chui/ThinClient.java in 4231b-11504. Please review.

**#67 - 04/29/2020 04:37 PM - Roger Borrello**

- % Done changed from 40 to 60

**#68 - 04/29/2020 05:28 PM - Roger Borrello**

From Hynek:

Both getAssigned and getEffective return 0-based values. These must be incremented by one (and perhaps also decremented if negative) before the values are returned to legacy code. See BaseEntity.getRow for an example. But please run some test cases in native 4GL to confirm the same behavior for FRAME-ROW and FRAME-COL.

I think you have to use getEffective. Consider the case when the frame position is not assigned explicitly, in this case getAssigned should/may return invalid coordinates.

Thanks,  
Hynek

**#69 - 04/29/2020 05:30 PM - Roger Borrello**

Roger Borrello wrote:

From Hynek:

Both getAssigned and getEffective return 0-based values. These must be incremented by one (and perhaps also decremented if negative) before the values are returned to legacy code. See BaseEntity.getRow for an example. But please run some test cases in native 4GL to confirm the same behavior for FRAME-ROW and FRAME-COL.

I think you have to use getEffective. Consider the case when the frame position is not assigned explicitly, in this case getAssigned should/may return invalid coordinates.

Thanks,  
Hynek

Thanks for that info. We do have checks for invalid coordinates, as part of the methods. Please review, as my testcases were successful after I incremented the return value.



#### #70 - 04/30/2020 02:29 PM - Hynek Cihlar

Code review 4231b revision 11504. The code changes seem OK. I think you should also test the changes in ChUI if you haven't.

#### #71 - 05/01/2020 01:05 PM - Roger Borrello

- % Done changed from 60 to 70

Testcase passes in ChUI, as well.

#### #72 - 05/14/2020 01:19 PM - Roger Borrello

### NO-ATTR-SPACE

All the documentation indicates ATTR-SPACE and NO-ATTR-SPACE have no effect. That they are supported only for backward compatibility.

#### #73 - 05/14/2020 01:58 PM - Greg Shah

Prove it (with testcases). You can only do this with frame that have many widgets. Make sure you test:

- more than 1 widget side by side on the same "row"
- widgets in multiple rows
  - explicitly using SKIP
  - implicitly by having too many widgets for the width of the frame
- ? (other stuff I'm not thinking about now)

#### #74 - 05/14/2020 02:42 PM - Roger Borrello

One issue is there isn't any documentation indicating the proper usage, or even what ATTR-SPACE and NO-ATTR-SPACE are supposed to do. The 9.1.E ABL reference says for backward compatibility. That's the oldest I've found. Internet searches, the same thing.

There are about 32 procedures in customer code that use either ATTR-SPACE or NO-ATTR-SPACE. I could start with them.

#### #75 - 05/14/2020 04:33 PM - Roger Borrello

Even though documentation for IS-ATTR-SPACE function indicates it is also supported only for backward compatibility, I'm looking through our code and getting the impression that ATTR-SPACE is like a setter for setting the "space-taking" attribute to yes and NO-ATTR-SPACE is the setter for setting the "space-taking" attribute to no, and IS-ATTR-SPACE is the getter for the "space-taking" attribute.

I've captured the sample code provided into ./uast/frames/is\_attr\_space.p, only to find we haven't implemented this built-in.

```
define variable termtyp as logical format "spacetaking/non-spacetaking".  
  
termtyp = is-attr-space.  
  
display "You are currently using a" termtyp no-label "terminal"  
with frame dl centered row 5.
```

The comment in gaps is is-attr-space is not supported but runtime backend is ready(for attribute). I don't think the conversion rule is in place, since conversion of the testcase yields:

```
[javac] /home/rfb/projects/VirtualBox-VMs/shared/projects/testcases/src/com/goldencode/testcases/frames/Is  
AttrSpace.java:31: error: no suitable method found for assign(no arguments)  
[javac]         termtyp.assign()
```

The implementations spread out:

- ui.client.Frame has isAttrSpace and setAttrSpace methods
- ui.ControlConfig has public boolean attrSpace = false; but no isAttrSpace or setAttrSpace methods
- ui.ControlEntity has setAttrSpace and setNoAttrSpace methods, but no isAttrSpace method.
- ui.FrameWidget has setAttrSpace and isAttrSpace methods.
- ui.GenericFrame has setAttrSpace method, but no isAttrSpace method.

#76 - 05/14/2020 06:23 PM - Roger Borrello

- File 20200514\_use-text.png added

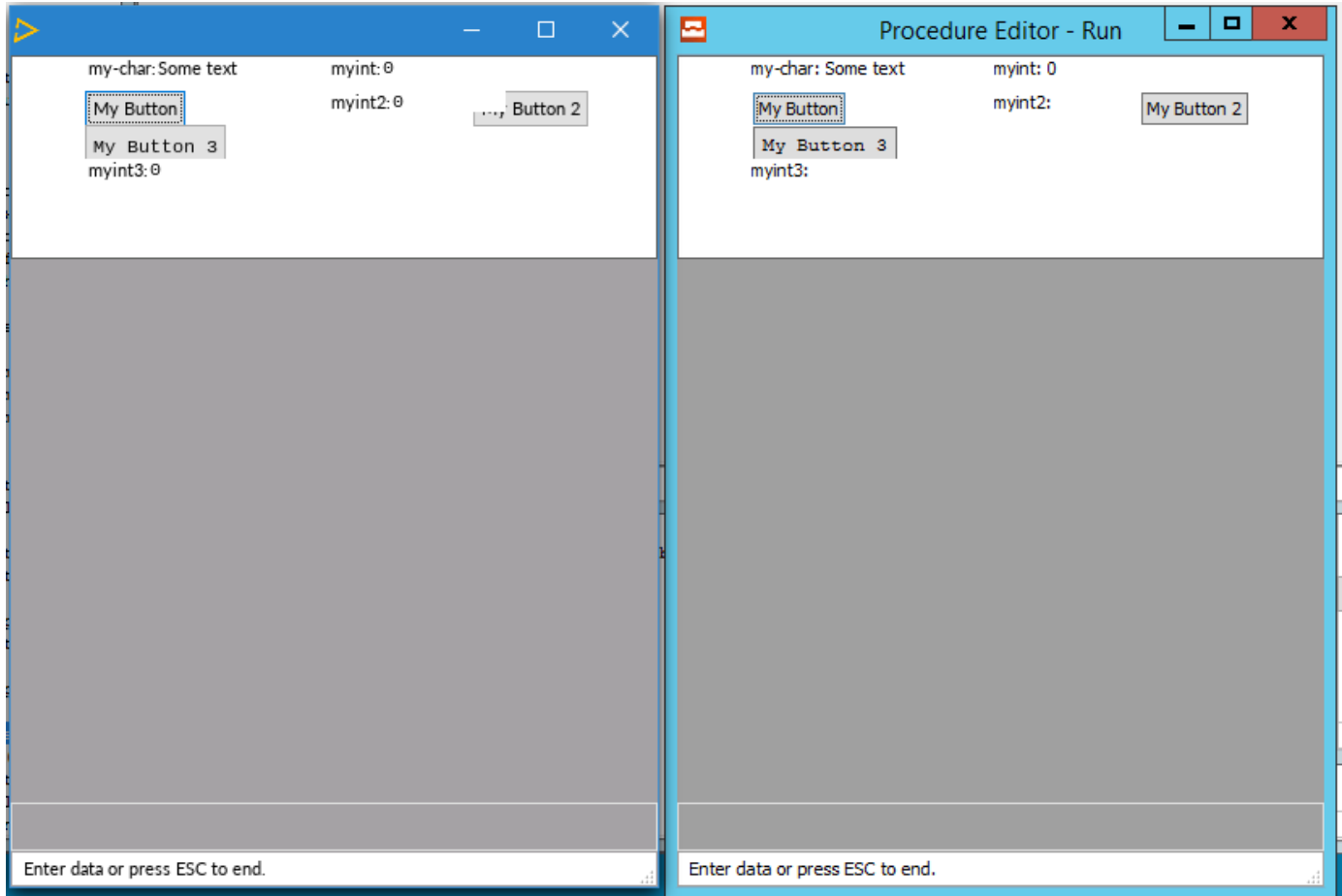
- File 20200514\_use-text-2.png added

## USE-TEXT

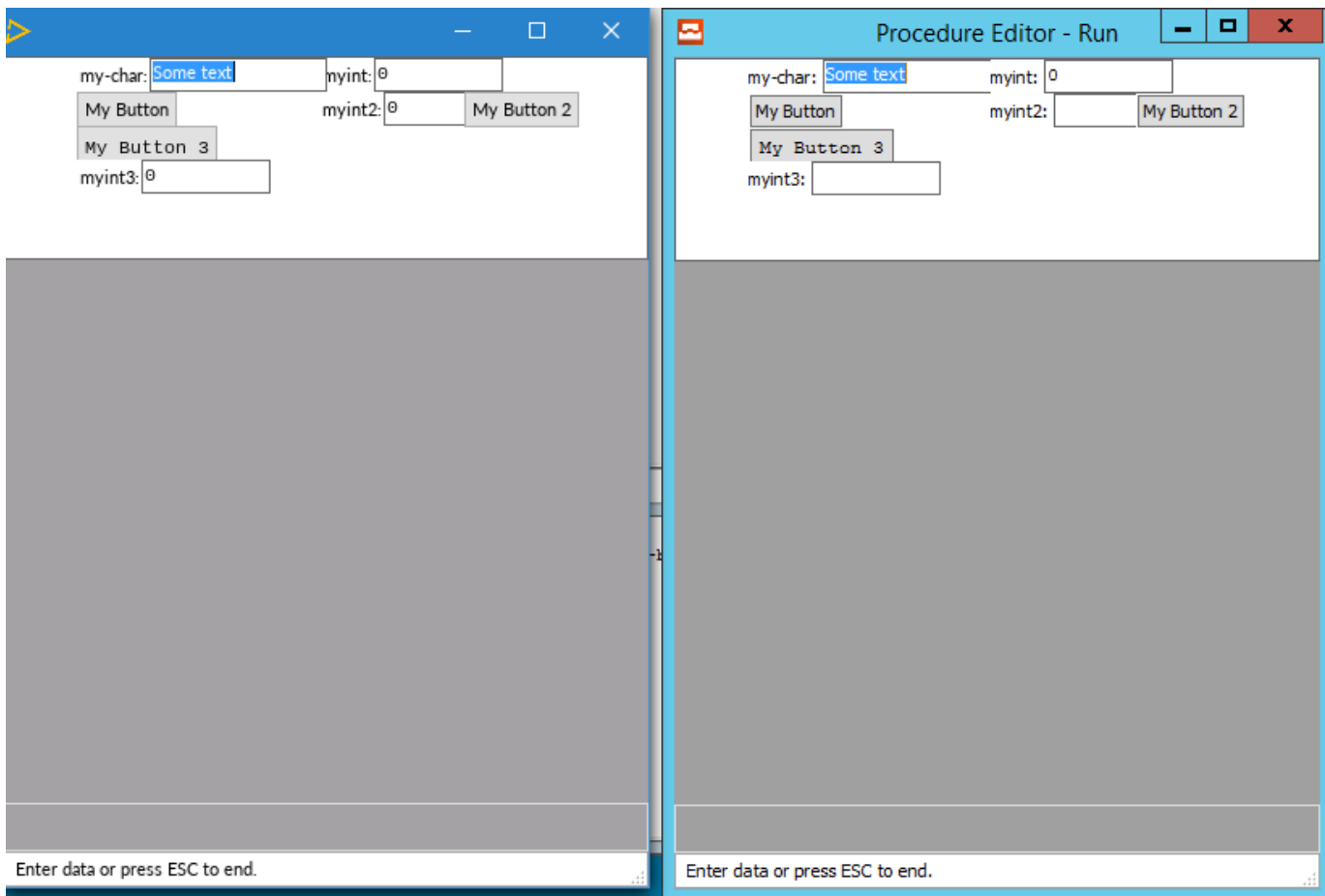
This looks like it's working properly. However, there are 2 UI issues, one might be related, the other unrelated.

1. The default values for the fill-ins that weren't included in the display statement seem to default to 0, whereas in 4GL, they are unknown. Give me an idea where to find this, and I can fix it.
2. There is an area around the fill-in widget that overlays an area of the screen, specifically the button next to it. Is there a precedent to the widgets, such that the button should lay on top of the fill-in? The second screen shot is the display without USE-TEXT which shows the fill-in is slightly covered by the button. When numbers are entered, they disappear behind it (in 4GL, as well) but you can see the commas getting added as you add more digits.

With USE-TEXT:



Without USE-TEXT:



Testcase: uast/frames/button\_mix\_use-text.p

```

def var my-char as character init "Some text" format "x(20)".
def var myint as int initial 0.
def var myint2 as int initial 2.
def var myint3 as int initial 3.

def button MyButton label "My Button".
def button MyButton2 label "My Button 2".
def button MyButton3 label "My Button 3" font 2.

form
  my-char    at column 10 row 1
  myint      at column 40 row 1
  MyButton   at column 10 row 2
  myint2     at column 40 row 2
  MyButton2  at column 58 row 2
  MyButton3  at column 10 row 3
  myint3     at column 10 row 4
  with frame ButtFrame side-labels centered size 80 by 6.

def frame ButtFrame
  my-char
  myint
  MyButton
  myint2
  MyButton2
  MyButton3
  myint3
  with centered use-text.

display my-char myint with frame ButtFrame.
enable all with frame ButtFrame.

on choose of MyButton do:
  myint = myint + 1.
  message "HELLO!!! (" myint ")".
end.

```

on F4 endkey.

wait-for endkey of MyButton.

### #77 - 06/18/2020 09:01 AM - Greg Shah

As part of this task, there were changes to the ThinClient.getFrameCol() method.

The original code:

```
public double getFrameCol(int frameId)
{
    Frame frame = getFrame(frameId);
    double result = 0;

    if (frame != null && frame.isVisible() && !frame.isRedirected())
    {
        result = ConfigHelper.getAssignedColumn(frame.config());
    }

    if (result == BaseConfig.INV_COORD)
    {
        result = 0;
    }

    return ++result;
}
```

The changed code which caused a regression in ChUI:

```
public double getFrameCol(int frameId)
{
    Frame frame = getFrame(frameId);
    double result = 0;

    if (frame != null && frame.isVisible() && !frame.isRedirected())
    {
        result = ConfigHelper.getAssignedColumn(frame.config()) + 1;
    }

    if (result == BaseConfig.INV_COORD)
    {
        result = 0;
    }

    return result;
}
```

The problem here is that this line `result = ConfigHelper.getAssignedColumn(frame.config()) + 1;` can now assign `result = -2147483647` where previously it could result in `result = -2147483648`. Consider that `INV_COORD Integer.MIN_VALUE -2147483648`. This means that the next code if `(result == BaseConfig.INV_COORD)` will be true in the original version and false in the changed version. In the original code that means the result would be reset to 0 and then incremented before returning it. But in the changed code the `-2147483647` was returned directly which is **bad**. This breaks our ChUI regression suite as you can imagine that code dependent upon this value being 1 will not work well when given `-2147483647` instead.

I've backed out this change in 4231b rev 11610. The proper fix will need to be in the next branch.

**#78 - 06/20/2020 12:00 PM - Greg Shah**

Task branch 4231b has been merged to trunk as revision 11347.

**#79 - 06/21/2020 09:28 PM - Roger Borrello**

Greg Shah wrote:

As part of this task, there were changes to the `ThinClient.getFrameCol()` method.

I've backed out this change in 4231b rev 11610. The proper fix will need to be in the next branch.

Fix checked into 3821c-11354.

**#80 - 06/22/2020 06:50 AM - Greg Shah**

Code Review Task Branch 3821c Revision 11354

This is incorrect. `ConfigHelper.getAssignedColumn(frame.config())` can return `BaseConfig.INV_COORD`. If you then add 1 to that number, then you will bypass the code below which resets it to 0.

If you test the broken ChUI application (test xxx 146), you will see the change does not work.

**#81 - 06/22/2020 08:16 AM - Roger Borrello**

Greg Shah wrote:

Code Review Task Branch 3821c Revision 11354

This is incorrect. `ConfigHelper.getAssignedColumn(frame.config())` can return `BaseConfig.INV_COORD`. If you then add 1 to that number, then you will bypass the code below which resets it to 0.

If you test the broken ChUI application (test xxx 146), you will see the change does not work.

If `ConfigHelper.getAssignedRow(frame.config())` can return `BaseConfig.INV_COORD` as well, I will have to modify `getFrameRow()` method, as well, as that is where I took the `getFrameCol()` code from.

**#82 - 06/22/2020 08:20 AM - Roger Borrello**

Roger Borrello wrote:

If ConfigHelper.getAssignedRow(frame.config()) can return BaseConfig.INV\_COORD as well, I will have to modify getFrameRow() method, as well, as that is where I took the getFrameCol() code from.

It does not. So the implementations will need to be a little different. I'll update.

**#83 - 06/22/2020 10:16 AM - Roger Borrello**

This was implemented in 3821c-11358. Please review, and let me know if there's a way to run an individual test for that ChUI application. I am scrambling to restore some disk space to the laptop so I can complete a customer build.

**#84 - 06/22/2020 04:30 PM - Greg Shah**

Code Review Task Branch 3821c Revision 11358

This seems better. Confirm it with testing.

**#85 - 07/07/2020 12:27 AM - Roger Borrello**

Greg Shah wrote:

Code Review Task Branch 3821c Revision 11358

This seems better. Confirm it with testing.

Regression testing was successful.

**#86 - 01/14/2021 11:05 AM - Greg Shah**

- Assignee changed from Greg Shah to Stanislav Lomany

Please see [#4164-1](#) for the remaining work (it is all in the runtime).

**#87 - 04/20/2021 06:21 PM - Stanislav Lomany**

My estimates:

SCROLL - unfortunately, that looks very time-consuming to me.

USE-TEXT - it works, but frame layout is not quite correct.

NO-ATTR-SPACE - documentation states that it has no effect.

auto-indent - that may take some time to investigate behavior in edge cases

SCROLL-NOTIFY, OFF-END, OFF-HOME, PARENT-WINDOW-CLOSE - shouldn't be a problem.

DESELECT(DESELECTION) - assuming that selection/deselection of widgets work fine, that shouldn't be a problem.

So, only SCROLL looks complicated as this point.

**#88 - 04/21/2021 05:32 AM - Greg Shah**

Focus on clearing everything except SCROLL. That one will be worked last.

**#89 - 04/21/2021 02:18 PM - Stanislav Lomany**

Guys, regarding USE-TEXT: currently in FWD this option does two things during conversion:

1. Sets `frame.setUseText(true)` which has no actual effect.
2. Converts widgets to `TextWidget` instead of `FillInWidget`.

It makes a widget affected by USE-TEXT to be correctly displayed as a text widget. However it may have incorrect width/height/font type because these parameters are controlled by `widget.explicitViewAs(true)` flag.

It would probably make sense to check `useText` flag of the parent frame in the same location as `explicitViewAs` flag. But the problem is that there may be no parent frame when `explicitViewAs` is checked (e.g. in `widget.initialize`).

So the obvious solution I see is to add `widget.explicitViewAs(true)` for widgets affected by USE-TEXT. What do you think?

And another issue to fix:

```
widget VIEW-AS FILL-IN ... WITH FRAME frame USE-TEXT
```

results in fill-in in 4GL and text in FWD.

**#90 - 04/21/2021 03:02 PM - Constantin Asofiei**

I don't understand how `explicitViewAs` flag is used during frame construction. AFAIK this flag is used only on client-side, after the frame has been setup. And a widget can't be pushed to the client-side without its frame.

**#91 - 04/21/2021 03:15 PM - Stanislav Lomany**

I don't understand how `explicitViewAs` flag is used during frame construction. AFAIK this flag is used only on client-side, after the frame has been setup.

Yes, it is used on the client side (defines if the font is fixed in `initialize`, and in `nativeWidth/nativeHeight`).

And a widget can't be pushed to the client-side without its frame.

That's good! That means I can use `TC.getFrame(config.frameId)` there.

**#92 - 04/21/2021 03:16 PM - Constantin Asofiei**

Constantin Asofiei wrote:

I don't understand how `explicitViewAs` flag is used during frame construction. AFAIK this flag is used only on client-side, after the frame has been setup. And a widget can't be pushed to the client-side without its frame.

OK, I see what you are referring to - `initialize` is called on client-side before the widget is attached to the frame as a child widget (and its parent is set). So you can't find it via that.

But the widget's `config.frameId` should be already set - and you can resolve it from the registry via its ID.

**#93 - 04/21/2021 03:16 PM - Constantin Asofiei**

Stanislav Lomany wrote:

I don't understand how `explicitViewAs` flag is used during frame construction. AFAIK this flag is used only on client-side, after the frame has been setup.

Yes, it is used on the client side (defines if the font is fixed in `initialize`, and in `nativeWidth/nativeHeight`).

And a widget can't be pushed to the client-side without its frame.

That's good! That means I can use `TC.getFrame(config.frameId)` there.

Exactly.

**#94 - 04/23/2021 10:10 AM - Stanislav Lomany**

USE-TEXT changes are committed 3821c rev 12317. Please review.



**#95 - 04/23/2021 11:03 AM - Hynek Cihlar**

Code review 3821c revision 12317.

Text.initialize assumes there is always a valid parent frame. This assumption may not hold true for dynamic widgets.

**#96 - 04/23/2021 05:59 PM - Stanislav Lomany**

Fixed in 3821c revision 12322.

**#97 - 05/04/2021 12:17 PM - Stanislav Lomany**

AUTO-INDENT committed 3821c rev 12364.

**#98 - 05/04/2021 02:28 PM - Stanislav Lomany**

I assume than implementation of SCROLL-NOTIFY should include SCROLL-HORIZONTAL and SCROLL-VERTICAL events (they are not quite the same).

**#99 - 05/04/2021 02:45 PM - Greg Shah**

Yes.

**Files**

---

browse_num_entries2.p	571 Bytes	02/14/2020	Stanislav Lomany
num_entries.png	14.9 KB	03/19/2020	Roger Borrello
num_entries2.png	25 KB	03/19/2020	Roger Borrello
frame_loc_attributes_get.png	50.1 KB	03/20/2020	Roger Borrello
frame_loc_attributes_get2.png	26.2 KB	04/03/2020	Roger Borrello
GUI_frame_loc_attributes_get2.png	14.8 KB	04/03/2020	Roger Borrello
20200514_use-text.png	19.5 KB	05/14/2020	Roger Borrello
20200514_use-text-2.png	18.2 KB	05/14/2020	Roger Borrello