User Interface - Feature #1805

add missing UI method support

10/30/2012 11:59 AM - Greg Shah

Status:	Closed	**Start date:**	02/18/2013
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	GUI Support for a Complex ADM2 App		
billable:	No	**version:**	
vendor_id:	GCD		

Description

Subtasks:

Feature # 2005: add conversion support for methods	**Closed**
Feature # 2006: add runtime support for new methods in GUI	**Closed**
Feature # 2144: add minimum runtime support for methods (server)	**Closed**

History

#1 - 10/31/2012 02:50 PM - Greg Shah

- Target version set to Milestone 12

#2 - 02/18/2013 06:45 PM - Greg Shah

The following are needed with high priority:

```
DESELECT-ROWS()
MOVE-BEFORE-TAB-ITEM()
```

#3 - 02/21/2013 05:59 AM - Eugenie Lyzenko

- File evl_upd20130220a.zip added

- File ui_meth_test.p.20130220a.zip added

Added support for MOVE-BEFORE-TAB-ITEM. The method DESELECT-ROWS has already supported. However we do not have the creation support for browse widget the method DESELECT-ROWS applied. It is just possible to define a handle and convert/compile: handle:deselect-rows()

#4 - 02/21/2013 05:14 PM - Greg Shah

The code changes look fine. Please be prepared to merge soon when we are able to schedule the update.

> However we do not have the creation support for browse widget the method DESELECT-ROWS applied.

I don't understand what you mean. Can you please show the 4GL that is not supported?

#5 - 02/21/2013 05:28 PM - Eugenie Lyzenko

Here is the sample we do not convert:
...
define browse bStat query q1 display.
create browse hBr.
...
Before using browse we need to create it. For static version of Browse the code:
bStat:deselect-rows().
is not converted(empty line)

#6 - 02/21/2013 05:33 PM - Eugenie Lyzenko

What will be the testing sequence of update application? This method update will be the first one, then update for attribute(#1804)? Or we can combine them into single drop?

#7 - 02/21/2013 05:51 PM - Greg Shah

I noticed that you have the attributes update built on top of the methods, which is fine. I am also OK with combining the updates, but in that case I prefer you to only have a single history entry for each file.

In regard to the staticBr:deselect-rows() problem:

The CREATE BROWSE statement is not yet supported (but it will be soon).

However, I am surprised that the static DEFINE BROWSE approach doesn't work.

Please create a testcase that uses DEFINE BROWSE and has a real frame that displays something. Include the use of deselect-rows() and possibly other attribute/method usage. Make sure that the code works in the 4GL. It is possible that if the static browse is not used in a real frame that P2J might drop that code. If the browse works in the 4GL, then test a conversion in P2J. Let me know the results.

#8 - 02/21/2013 06:40 PM - Eugenie Lyzenko

- File browse_ui_test0.p added

You are right. If browse not used in real code the P2J removes the code. The example in browse_ui_test.p properly converts
...
brws:deselect-rows().
...
into
...
fFrameFrame.widgetBrws().deselectRows();
...
And conversion code is compiled. Sorry for false alarm.

#9 - 02/12/2014 01:57 PM - Vadim Gindin

As far as I can see DESELECT-ROWS method is really implemented.. What needs to be done in the task #2144 for this task?

#10 - 02/12/2014 04:14 PM - Greg Shah

I'm not sure. Having it on our list may be just a mistake. Please write a testcase (or find one in the testcases/uast/ project) to confirm that DESELECT-ROWS is working and then post your results. If it is working then no more effort will be needed. Make sure to test it with 4 cases:

1. invalid browse
2. no records selected
3. 1 record selected
4. more than 1 record selected

Make sure to check if the return value is always true in the 4GL for all these cases.

Also make sure to check the status of the record buffer. The 4GL docs state that it "clears" the associated record buffer. I'm not sure what that means, but it possible means that the buffer is NOT AVAILABLE(buffer).

#11 - 02/14/2014 05:56 AM - Vadim Gindin

1) query is not opened.

```
Browse widget must contain at least one record before DESELECT-ROWS method is used. (2108)
deselect-rows have returned FALSE
```

2) query is opened but rows are empty.

```
Browse widget must contain at least one record before DESELECT-ROWS method is used. (2108)
deselect-rows have returned FALSE
```

3) Invalid handle

```
Invalid handle.  Not initialized or points to a deleted object. (3135)
Cannot access the DESELECT-ROWS attribute because the widget does not exist. (3140)
```

4) create browse - handle is browse but not in frame

```
Unable to realise BROWSE widget because it is not in a frame. (4040)
Unknown error code 1 for attribute ? on the BROWSE widget. (4104)
**Unable to query attribute DESELECT-ROWS for BROWSE widget. (4077)
deselect-rows have returned FALSE
```

5) handle points to a button

```
**DESELECT-ROWS is not a queryable attribute for BUTTON widget. (4052)
```

```
deselect-rows have returned FALSE
```

6) cases when code is correct and no rows selected or one or several rows are selected.
Result: deselect-rows function works correctly and returns TRUE.

#12 - 02/14/2014 05:37 PM - Greg Shah

Good findings. Go ahead and implement the proper behavior.

```
Invalid handle.  Not initialized or points to a deleted object. (3135)
Cannot access the DESELECT-ROWS attribute because the widget does not exist. (3140)
```

These may already be implemented in a generic way. Test it to make sure. But I think all the other behaviors need code.

#13 - 02/15/2014 12:13 PM - Vadim Gindin

- File conversion_err.log added

I faced with the following conversion problem. The parser makes wrong annotations with links to schemaname, bufname and dbname during the parsing of define browse statement. It leads to the error:

```
..
Caused by: com.goldencode.p2j.pattern.CommonAstSupport$UserGeneratedException: Cannot find buffer named p2j_te
st.cs-test_p2j_test.cs-test for ref type 21! [FIELD_CHAR id <1352914698433> 23:13]
..
```

Full log in attached file *conversion_err.log*

Here is the test procedure:

```
   def var i as int.

   def temp-table testt no-undo
       field tot-hrs   as int
       field proj-mgr  as int
       field one       as char
       field yes_no    as logical.

   do i = 1 to 20:
      create testt.
      tot-hrs  = i + 40.
      proj-mgr = 1.
      one      = "I".

      if i < 11 then
          yes_no = yes.
      else
          yes_no = true.
  end.
```

```
  def query q for testt.
  def browse brws query q no-lock no-wait
      display test
      enable yes_no auto-return
      with multiple 5 down centered.

  open query q
      for each testt where tot-hrs <> 60 no-lock.

  form brws with frame f-frame.

  on "F7" anywhere do:
    if brws:deselect-rows() in frame f-frame <> true then message "false".
  end.

  enable all with frame f-frame.
  wait-for close of current-window.
```

Temp-table is called testt and here is correct ast part of the statement from the line 11:

```
              <ast col="5" id="1352914698330" line="11" text="tot-hrs" type="FIELD_INT">
                <annotation datatype="java.lang.Long" key="oldtype" value="2345"/>
                <annotation datatype="java.lang.String" key="schemaname" value="testt.tot-hrs"/>
                <annotation datatype="java.lang.String" key="bufname" value="testt"/>
                <annotation datatype="java.lang.String" key="dbname" value=""/>
                <annotation datatype="java.lang.Long" key="recordtype" value="14"/>
                <annotation datatype="java.lang.String" key="name" value="tot-hrs"/>
                <annotation datatype="java.lang.Long" key="type" value="373"/>
                <annotation datatype="java.lang.Boolean" key="reachable" value="true"/>
              </ast>
```

Take a look at fields schemaname, bufname and dbname. They have correct values.
And here as the ast node for the define browse statement and the display part of it:

```
       <ast col="5" id="1352914698430" line="23" text="display" type="KW_DISP">
           <annotation datatype="java.lang.Boolean" key="reachable" value="true"/>
           <ast col="0" id="1352914698432" line="0" text="expression" type="EXPRESSION">
             <annotation datatype="java.lang.Boolean" key="reachable" value="true"/>
             <ast col="13" id="1352914698433" line="23" text="test" type="FIELD_CHAR">
               <annotation datatype="java.lang.Long" key="oldtype" value="2345"/>
               <annotation datatype="java.lang.String" key="schemaname" value="p2j_test.cs-test.test"/>
               <annotation datatype="java.lang.String" key="bufname" value="p2j_test.cs-test"/>
               <annotation datatype="java.lang.String" key="dbname" value="p2j_test"/>
               <annotation datatype="java.lang.Long" key="recordtype" value="12"/>
               <annotation datatype="java.lang.String" key="name" value="test"/>
               <annotation datatype="java.lang.Long" key="type" value="365"/>
               <annotation datatype="java.lang.String" key="format" value="&amp;quot;x(8)&amp;quot;"/>
               <annotation datatype="java.util.ArrayList" key="label">
                 <listitem datatype="java.lang.String" value="&amp;quot;Character&amp;quot;"/>
               </annotation>
               <annotation datatype="java.lang.Boolean" key="reachable" value="true"/>
             </ast>
           </ast>
...
```

Our fields have other values. Where can I find the rules for the generation of these annotations? Is this type of define browse statement is really supported?

#14 - 02/15/2014 12:48 PM - Constantin Asofiei

Vadim: you have a typo on line 23: you need to have display testt instead of display test. This was easily spotted when I ran the program on lindev01.

Hynek/Marius: when the program is ran in swing mode, I get a NPE:

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
    at com.goldencode.p2j.ui.Color.toForeground(Color.java:293)
    at com.goldencode.p2j.ui.client.swing.ColorMapper.toFont(ColorMapper.java:137)
    at com.goldencode.p2j.ui.client.chui.driver.swing.ChuiSimulator.paintComponent(ChuiSimulator.java:716)
    at javax.swing.JComponent.paint(JComponent.java:1054)
    at javax.swing.JComponent.paintToOffscreen(JComponent.java:5221)
    at javax.swing.RepaintManager$PaintManager.paintDoubleBuffered(RepaintManager.java:1482)
    at javax.swing.RepaintManager$PaintManager.paint(RepaintManager.java:1413)
    at javax.swing.RepaintManager.paint(RepaintManager.java:1206)
    at javax.swing.JComponent._paintImmediately(JComponent.java:5169)
    at javax.swing.JComponent.paintImmediately(JComponent.java:4980)
    at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:770)
    at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:728)
    at javax.swing.RepaintManager.prePaintDirtyRegions(RepaintManager.java:677)
    at javax.swing.RepaintManager.access$700(RepaintManager.java:59)
    at javax.swing.RepaintManager$ProcessingRunnable.run(RepaintManager.java:1621)
    at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:251)
    at java.awt.EventQueue.dispatchEventImpl(EventQueue.java:721)
    at java.awt.EventQueue.access$200(EventQueue.java:103)
    at java.awt.EventQueue$3.run(EventQueue.java:682)
    at java.awt.EventQueue$3.run(EventQueue.java:680)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.security.ProtectionDomain$1.doIntersectionPrivilege(ProtectionDomain.java:76)
    at java.awt.EventQueue.dispatchEvent(EventQueue.java:691)
    at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:244)
    at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:163)
    at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:151)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:147)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:139)
    at java.awt.EventDispatchThread.run(EventDispatchThread.java:97)
```

I know Marius worked on the swing color implementation, but I don't know in which part the bug is. Can one of you take a look and fix this?

Vadim: please check if the terminal client shows this problem too.

#15 - 02/15/2014 03:47 PM - Hynek Cihlar

Color.toForeground receives a null Color instance hence the NPE. It looks like ColorMapper.toFont should handle the case when a screen display data cell contains no color definition.

I wouldn't do a default-color logic in Color.toForeground since it is too low level.

#16 - 02/17/2014 05:47 AM - Marius Gligor

My changes for ChUI colors implementation was done only in ColorMapper class.

Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
at com.goldencode.p2j.ui.Color.toForeground(Color.java:293)
at com.goldencode.p2j.ui.client.swing.ColorMapper.toFont(ColorMapper.java:137)
at com.goldencode.p2j.ui.client.chui.driver.swing.ChuiSimulator.paintComponent(ChuiSimulator.java:716)
at javax.swing.JComponent.paint(JComponent.java:1054)

Strange but Color.java 293 on my p2j project point to a javadoc line!!!
Inside ColorMapper.toFont I added only some lines for colors but the original code contains also other lines of code like color.hasReverse(), color.hasBold() and color.hasBlink()
which should also throw NPE in case of color is null. Why this not happen before my changes?

#17 - 02/17/2014 02:15 PM - Constantin Asofiei

Marius Gligor wrote:

> Strange but Color.java 293 on my p2j project point to a javadoc line!!!

This is because the stacktrace is from H003 version of Color.java - H004 was not released yet at the time the stacktrace was captured.

> Why this not happen before my changes?

The test was just written/checked, so the problem looks like a latent bug. Please add this bug to your list, it should be covered by the #2240 - implement proper color support in the Swing ChUI and AJAX ChUI clients task.

#18 - 02/17/2014 02:23 PM - Marius Gligor

Sorry but I don't understood. What version of Color is on my project H003 or H004?

#19 - 02/17/2014 02:30 PM - Constantin Asofiei

Marius Gligor wrote:

> Sorry but I don't understood. What version of Color is on my project H003 or H004?

Check the header of your Color.java - the number of the last history entry gives you the version. H004 was released with revision 10467 yesterday, so if you have the latest rev, then you have H004. Line 293 in H003 corresponds with line 289 in H004.

#20 - 04/03/2014 05:34 AM - Vadim Gindin

I've found the following bug in Java implementation during the testing. I ran *browse_ui_test1.p* from the /uast folder. It started and displayed browse widget with filled rows. When I press down-arrow - the error is rised:

```
None of the widgets used in the WAIT-FOR statement are in a state (such as SENSITIVE) such that the specified event can occur. WAIT-FOR terminated. (4123)
```

I debugged it and found that it appears in ThinClient.stopNoAvailableWidgets() because of the following reason. Procedure contains following exit condition: wait-for close of current-window.. ThinClient stores handles of exit components in the field exitResources and ThinClient.viewWorker at some moment checks if all such handles are valid. This array at the moment of error contains only one handle - obviously the handle of the current-window and server-side validation of this handle fails.

#21 - 04/03/2014 06:09 AM - Constantin Asofiei

Vadim Gindin wrote:

> I ran *browse_ui_test1.p* from the /uast folder.

Please commit the test to the testcases project.

#22 - 04/03/2014 07:15 AM - Vadim Gindin

Sorry, I thought that it is already committed. Now I've committed it (the file *browse_ui_test1.p*)

#23 - 04/04/2014 06:39 AM - Constantin Asofiei

Vadim Gindin wrote:

> Sorry, I thought that it is already committed. Now I've committed it (the file *browse_ui_test1.p*)

The test works fine on my system: client started, pressed down/up arrows, pressed F7, then F4 to exit. Please post the stacktrace and some other details about how you run the testcases.

#24 - 04/04/2014 06:44 AM - Constantin Asofiei

PS: my test was with the latest P2J from bzr, without your #2162 changes.

#25 - 04/07/2014 05:07 PM - Vadim Gindin

There is no stacktrace.. I just pressed down-arrow right after start and got the error. I also have the last revision from bzr but I have my changes from 2162.

#26 - 04/08/2014 03:16 AM - Constantin Asofiei

Vadim Gindin wrote:

> There is no stacktrace.. I just pressed down-arrow right after start and got the error. I also have the last revision from bzr but I have my changes from 2162.

When you work on multiple tasks, is best to not mix the code from one task with the other - you can create multiple P2J projects in your IDE, one for each task and each one always synced with latest bzr. Also, running the test with your 0403a.zip from 2162 doesn't show any problems to me.

#27 - 04/24/2014 04:04 AM - Vadim Gindin

There are something confusing me.

1. It's a comments to methods Frame.moveToTop() and Frame.moveToBottom(), pointing that they related to tab order. But it don't seems that way. Anyway the client implementation of them uses AbstractContainer.widgets array. If these methods are really conforms to tab order operations. It means that I should use this array for my implementation too. Isn't it?

2. If not should I implement the client part too? If yes what to use?

3. There are also moveAboveWidget and moveBelowWidget that use FrameFocusTransferManager for it's implementation. They implement Z-order as far as I can understand. Z-order and tab order are different things here. Isn't it?

#28 - 04/24/2014 08:57 AM - Greg Shah

> It's a comments to methods Frame.moveToTop() and Frame.moveToBottom(), pointing that they related to tab order. But it don't seems that way.

Progress allows you to tab between frames. You have to have multiple frames enabled for editing at the same time. But the focus processing is quite complex as a result of this design. The z-order does have an effect on the focus processing and tab order.

We have some documented research and conclusions in the "Progress Focus Management" section of the com/goldencode/p2j/ui/chui/package.html document. Please review that section carefully.

#29 - 05/01/2014 02:05 PM - Vadim Gindin

Here are my testcases for MOVE-BEFORE-TAB-ITEM attribute.
1. Simplest working case. The value is a handle one of widgets of the same field-group.
Test: *01_working_handle.p*
Description: One frame, three buttons. We are assigning the valid handle.
Result: Tab order is changed successfully.

2. The value is FIRST-TAB-ITEM value of the current frame.
Test: *02_working_from_1tabitem.p*
Description: One frame, three buttons. We are assigning the valid handle.
Result: Tab order is changed successfully.

3. The value is a handle of the frame nested to the current frame.
Test: *03_working_nested_frame.p*
Description: One frame, three buttons and nested frame. We are assigning the valid handle.
Result: Tab order is changed successfully.

4. The value is '?'
Test: *04_unknown_val.p*
Description: One frame, three buttons. We are assigning an unknown handle.
ERROR:
**Attribute MOVE-BEFORE-TAB-ITEM for the BUTTON btn2 was passed an invalid widget. (4062)

5. The value is the handle of the neighbor frame (to current frame)
Test: *05_foreign_frame.p*
Description: two frames, three buttons in the first frame. We are assigning the handle of the second frame.
ERROR:
**Unable to query attribute MOVE-BEFORE-TAB-ITEM for BUTTON btn2. (4077)

6. The value is the handle of the button nested to the frame that is nested to the current frame.
Test: *06_button_of_the_nested_frame.p*
Description: One frame, three buttons in the first frame, nested frame with another one button. We are assigning the handle of the button in the nested frame.
ERROR:
**Unable to query attribute MOVE-BEFORE-TAB-ITEM for BUTTON btn2. (4077)

7. The value is the handle of current frame.
Test: *07_own_frame.p*
Description: One frame, three buttons in the first frame, We are assigning the handle of the current frame.
ERROR:
**Unable to query attribute MOVE-BEFORE-TAB-ITEM for BUTTON btn2. (4077)

#30 - 05/01/2014 02:55 PM - Vadim Gindin

I read Focus Management chapter as you advised and took it into account. At this moment I just want to understand the current implementation. That is why I asked the 3 questions in the note #27. Could you answer them?

About Focus management. Described algorithm is really complex, as you wrote and I have some hope that I won't need to make changes in it. As far as I can understand MOVE-BEFORE-TAB-ITEM is applied only to the widgets in the same field-group. My tests shows that it can be interpreted as this attribute is applied to widgets on the same level of widgets tree. Is it correct? Please take a look at my tests and advise me whether I need to add some specific test cases.

#31 - 05/02/2014 09:00 AM - Greg Shah

This simple testcase is all you would need to answer your own questions:

```
def var one as char.
def var two as char.
def var three as char.

display one two three with frame f0.
enable all with frame f0.
wait-for "go" of frame f0.
```

Convert it and step through the code in a debugger to see how it responds to the TAB.

> 1. It's a comments to methods Frame.moveToTop() and Frame.moveToBottom(), pointing that they related to tab order. But it don't seems that way. Anyway the client implementation of them uses AbstractContainer.widgets array. If these methods are really conforms to tab order operations.

Yes, they certainly do relate to this. Look at focusWorker() to see a good example. A quick check of the other usage in this file shows that this list is the entire basis for focus processing (which is ordered by the "tab order"). The order of the list is critical.

> It means that I should use this array for my implementation too. Isn't it?

Anything that changes the tab order must change the order of this list to make it work.

> 3. There are also moveAboveWidget and moveBelowWidget that use FrameFocusTransferManager for it's implementation. They implement Z-order as far as I can understand. Z-order and tab order are different things here. Isn't it?

No, it is not about z-order. The FrameFocusTransferManager manages the FOCUS (the current position of the cursor for input). Within a single frame, the TAB order determines the forward/back sequencing of focus changes. But between frames, focus changes must be handled in a more complex manner which is affected by z-order. But the FrameFocusTransferManager does NOT manage z-order.

#32 - 05/02/2014 09:20 AM - Greg Shah

While you are doing this work, please implement the MOVE-AFTER-TAB-ITEM method and the FIRST-TAB-ITEM/NEXT-TAB-ITEM/LAST-TAB-ITEM attributes. It will also make it easier to write tests that can work fully.

I am worried that we are missing something with child frames. The 4GL docs state this:

```
If handle specifies a frame, the tab order of the method widget is positioned so that it
precedes the first widget parented by the frame in that frames own tab order. For more
information on how frames owned by a field group participate in the tab order of that
field group, see the FRAME widget reference entry.
```

This means that you need to expand the cases similar to 03_working_nested_frame.p. You need to see what happens to the tab order when you move the nested child frame around to different positions, from different starting positions. See also what happens when you move widgets around the nested child frame.

I also think you need to change all of the tests to allow you to list out the tab order before and after. You can use FIRST-TAB-ITEM/NEXT-TAB-ITEM/LAST-TAB-ITEM to do this non-interactively.

#33 - 05/07/2014 04:02 PM - Vadim Gindin

1) I added programmed order definition.

2) This peace of Progress doc looks strange. How the method widget can become a part of it sibling frame own tab order while the method widget asides not inside it? How we really can talk about separate tab orders? When we are in the last widget of a nested frame and we are pressing TAB we are going to the next widget in a common tab order. Isn't it? Anyway I'm testing it.

#34 - 05/13/2014 05:38 AM - Vadim Gindin

Added additional tests, 08-16 for the case with nested frames. All works as expected

#35 - 05/15/2014 03:15 AM - Vadim Gindin

I think It would be wrong to use AbstractContainer.widgets order for the focus management. We really have two orders: the first is the widgets displaying order, and the second - the tab order. As far as I can understand AbstractContainer.widgets is just displaying order and the tab order is not implemented yet. It needed to add additional list of widgets ids for tab order and use id in the AbstactContainer.focusWorker.

There is another important moment: where to implement the moving logic. If we are trying to keep the client "thin" as possible, then I should implement this logic on the server: add widgets ids list for the tab order to ScreenDefinition and manage it there.

#36 - 05/15/2014 08:46 AM - Greg Shah

> I think It would be wrong to use AbstractContainer.widgets order for the focus management.

Perhaps. But the code today does use AbstractContainer.widgets for both focus management and for drawing. As far as we know, this is not causing any problems. It is possible that this will be a problem in the future if we hit some behavior of the 4GL where the drawing and tab order are different. Consider that the 4GL doesn't really allow widgets to draw on top of one another. They are always laid out in their own space, with no overlaps. The only exception that comes to mind is when we must draw the drop-down portion of the combo-box, this can temporarily draw over other widgets. IIRC, we have a special drawing loop for that processing (it doesn't redraw everything, just the drop-down portion). When we implement menus (including popup menus), we will have to do something similar.

I don't (yet) know of any requirements that will force us to split the widgets list into 2. For now, I really don't want to rewrite all of the focus processing to separate it. The focus processing is quite complex and it is sensitive to change. I recall it took quite a while to get right. There are complexities added because of cross-frame processing, triggers and the possibility of nested triggers. So I don't want to introduce risk here unless we must. Do you know of some important functional reason that we need to make this change.

I realize that we are "conflating" z-order and tab order. You are correct that traditionally in a UI these are 2 separate things. But since the initial implementation did not require this (smarter) approach, I prefer not to change it now unless we need to.

> the tab order is not implemented yet

It is definitely implemented. It just may be a bit confusing because we use the same data structure for both drawing and tab order. Please review the code in ThinClient.enable(). The ENABLE processing is the primary way that 4GL apps setup the tab order. The ENABLE can be called multiple times with different widget lists, in order to define a non-default tab order. Likewise, ENABLE honors any value set with NEXT-PROMPT (which sets the next widget to get focus). All of this ultimately uses AbstractContainer.moveToTop() to maintain the tab order (the Frame.contentPane is a ScrollContainer which is an AbstractContainer).

> There is another important moment: where to implement the moving logic. If we are trying to keep the client "thin" as possible, then I should implement this logic on the server: add widgets ids list for the tab order to ScreenDefinition and manage it there.

The logic should be implemented in the place where it will be simplest. I think that is on the client. There is no need for the server to maintain duplicate state and logic.

#37 - 05/16/2014 03:44 AM - Vadim Gindin

- File vig_upd20140516a.zip added

1. Methods moveToTop and moveToBottom change the real place of the widget in the displaying order. They really moves a widget. At the same time the methods like moveBeforeTabItem do not moves a widget. They only change the tab order and do not affect displaying order and stay widget at the same place relatively the other widgets.

2. From the point №1 I see a necessity of two different lists. Lets assume we have a frame with 3 widgets in the following order: w1, w2, w3. For this frame there are 6 possible tab orders (all possible permutations). And vice versa. If we have some tab order of widgets there possible several variants of their positions relative to each other.

3. AbstractContainer.widgets will consist of that widgets in that concrete displaying order. In because of the point №2 it cannot provide possibility to contain different tab order. I.e. the state is described by exactly 2 lists: displaying and tab-ordering. When I wrote that tab order is implemented I expressed my thought wrong. *I wanted to say, that as far as I could understand there are only default tab order with tab events processing are implemented, but the methods changing a tab order are not implemented yet. That is why it is sufficient to have only one widgets list for the current implementation.*. At this moment I didn't see ThinClient.enable() and NEXT-PROMPT implementation. Probably I was wrong. I'm going to look there.

4. I prepared some implementation prototype yesterday. It uses two lists. It saves tab order in ScreenDefinition and transfers it to the client. The tab order list is added to AbstractContainer. This implementation does not work correctly at this moment but it really affects the tab order in a simple case without cross-frames. I would want you to look there and correct me.

#38 - 05/18/2014 09:09 AM - Vadim Gindin

I don't fully understand the real link between z-order and tab order. Definitely it really exists and it complex. I looked at ThinClient.enable() method and the NEXT-PROMPT implementation as you advised and here what I found.

1. Peculiarity of enable implementation is the case when there are widgets list is specified in ENABLE statement. In that case this widgets list is moved to top of z-order independently of whether it is full or not. It really modifies the AbstractContainer.widgets list. At this moment it's hard to qualify it and I'm going to analyze it deeper.

2. NEXT-PROMPT is implemented as a separate field Frame.focusId and do not modify AbstractContainer.widgets list. This field is used to set focus after AbstractContainer.widgets processing. So it cannot help us in this situation.

#39 - 05/20/2014 11:35 AM - Greg Shah

Before you spend additional time on implementation, please post a testcase here which would break in P2J because the TAB order is different from the Z-ORDER.

I don't know of any case. But if you find one, then we will take the added risk of the changes needed to split these up. If you cannot find such a case, then please implement the simple way. As far as I understand it, this should be quite straightforward because we already have the "tools" in place to move the widgets around.

#40 - 05/20/2014 04:40 PM - Vadim Gindin

At this moment I didn't find such test case. I don't feel free with terms. Could you describe how do you define z-order and it's conflation with tab order. I mean your definition corresponding to current implementation. It would be great if we can find out how the implementation author of the AbstractContainer.widgets planned to use this collection. I thought that it is intended to contain the widgets in displaying order but I'm not sure of that.

There are some different ways of using z-order (specification of parent-child relationships). For example, I can create 3 frames:

```
frame1->frame2
      ->frame3
```

and frame3 can overlay frame2 or vice versa depending on concrete positions of these frames. The display can looks like z-order is frame1->frame2->frame3 or frame1->frame3->frame2. But the real tab order does not depends on that visual effect and is defined by other rules.

OVERLAY attribute or function that can also affect z-order.

FRAME attribute of the nested frames

If the z-order has no definition at this moment it can took some time to test it, but I'm not sure if it really necessary now because it can take some valuable time. May be instead of digging into it I should implement the simple case you noted..

P.S. Could you advice me how to create nested frame between field-level widgets?

#41 - 05/21/2014 09:13 AM - Greg Shah

> At this moment I didn't find such test case.

That is good news.

> I don't feel free with terms. Could you describe how do you define z-order and it's conflation with tab order. I mean your definition corresponding to current implementation.

Z-ORDER is the visual stacking of widgets, such that any widgets that are higher in the Z-ORDER will obscure overlapping widgets that are lower in the Z-ORDER.

TAB order is the order in which FOCUS will change in response to events like a TAB (nextFocus()) or BACK-TAB (previousFocus()).

When I say these two concepts are conflated, I mean that at the individual FRAME level, we store/maintain a single data structure for both Z-ORDER and TAB order.

The 4GL is very limited in how WIDGETS can overlap. As far as I know they don't overlap (except as I have previously noted).

The 4GL does provide for FRAMES to overlap (OVERLAY...). That means that the FRAME Z-ORDER does matter BUT a FRAME is not part of the TAB order except to the extent that WIDGETS in that FRAME are enabled (activated for editing and placed into the TAB order). I believe that this is

not a factor that will affect how we manage TAB order.

> It would be great if we can find out how the implementation author of the AbstractContainer.widgets planned to use this collection. I thought that it is intended to contain the widgets in displaying order but I'm not sure of that.

The displaying order of WIDGETS within the same FRAME does not matter because those WIDGETS cannot overlap. For this reason, the purpose of AbstractContainer.widgets is not to manage the display order.

The original implementer of that code is not with the company now, but I am confident that my assessment is true.

> If the z-order has no definition at this moment it can took some time to test it, but I'm not sure if it really necessary now because it can take some valuable time. May be instead of digging into it I should implement the simple case you noted..

No more research is needed at this time. Please go ahead with the simple implementation.

> P.S. Could you advice me how to create nested frame between field-level widgets?

As far as I know, it is not possible.

Constantin? Do you know of any way to do this?

#42 - 05/21/2014 02:09 PM - Constantin Asofiei

Vadim, some thoughts:

1. I don't recall and can't find a way of jumping from frame F1 to frame F2 (both with enabled widgets) just by pressing the TAB key. As I recall from looking into the MAJIC source code, they jump between frames via custom triggers.
2. but TAB is not the only way for a widget to receive focus automatically (i.e. without a trigger explicly pointing to it): consider you have 2 or more frames, like this:

```
def var i as int.
def var j as int.
def var k as int.

form i j with frame f1 overlay row 1 col 1 title "F1".
form j with frame f2 overlay row 1 col 1 title "F2".
form k with frame f3 overlay row 1 col 1 title "F3".
```

```
enable all with frame f1.
enable all with frame f2.
enable all with frame f3.

on 1 anywhere do:
self:enabled = false.
end.

wait-for close of current-window.
```

> When pressing 1 and the widget is the last enabled widget in the frame, then the focus is moved automatically to the next enabled widgets: in this case, Z-ORDER I think will have something to do with how the next focus will be determined.
>
> P.S. Could you advice me how to create nested frame between field-level widgets?

I don't understand your question. How do you want the frame to look/behave? As far as I know, frames can't be nested (they can have as children only non-frame widgets). More, looking into the doc for MOVE-BEFORE-TAB-ITEM, it states that Both the method widget and the specified widget must be in the same field group. - thus the restriction is even more plausable, as you can't even jump outside of a field-group, with this method.

#43 - 05/21/2014 02:30 PM - Greg Shah

> I don't recall and can't find a way of jumping from frame F1 to frame F2 (both with enabled widgets) just by pressing the TAB key.

This is possible. We have documented cases of this in the chui/package.html "Focus Management" section.

> frames can't be nested (they can have as children only non-frame widgets)

The 4GL provides "frame family" support where you set the FRAME my-frame:FRAME = FRAME my-parent-frame.

However, as far as I know, a child frame does not take part as a widget in a FIELD-GROUP.

> More, looking into the doc for MOVE-BEFORE-TAB-ITEM, it states that Both the method widget and the specified widget must be in the same field group. - thus the restriction is even more plausable, as you can't even jump outside of a field-group, with this method.

This is very good news.

#44 - 05/22/2014 02:15 AM - Constantin Asofiei

Greg Shah wrote:

> I don't recall and can't find a way of jumping from frame F1 to frame F2 (both with enabled widgets) just by pressing the TAB key.

This is possible. We have documented cases of this in the chui/package.html "Focus Management" section.

Do you have some specific case in mind, when triggers are not involved at all? The "Focus Management" section describes cases when focus is adjusted automatically (as widgets/frames get disabled/hidden) or manually, via APPLY or triggers...

> frames can't be nested (they can have as children only non-frame widgets)

The 4GL provides "frame family" support where you set the FRAME my-frame:FRAME = FRAME my-parent-frame.

Yes, you are correct, in this case the frames are "merged" and TAB can walk between both frames.

#45 - 05/26/2014 04:25 PM - Vadim Gindin

- File vig_upd20140526a.zip added

Here is the first working variant. It works for simple cases (without nested frames and with only one field-group). Following difficulties:
1) The linkage on field-groups requires the logic on the server, because it is exist only there. Commonly speaking there are not the one tab order for the frame, but probably each tab order for the each field-group.
Is it needed to test down-frame?!
1a) It is needed to assign some IDs to the field-group to have a possibility to distinguish field-groups. How to do it better?
2) Nested frames. The problem is that there are no field frame in GenericFrame. It need for the support of the linkage between frames: frame frame2:frame = frame frame1:handle. (frame:frame property).

#46 - 05/27/2014 01:35 PM - Constantin Asofiei

Vadim Gindin wrote:

> Here is the first working variant. It works for simple cases (without nested frames and with only one field-group). Following difficulties:
> 1) The linkage on field-groups requires the logic on the server, because it is exist only there. Commonly speaking there are not the one tab order for the frame, but probably each tab order for the each field-group.

Have you found that a frame can have more than one field-group (except for the down frame)? I mean, a way to group the widgets in a frame, in separate field-groups...

> Is it needed to test down-frame?!

Yes, do some tests. For now, assume that only the widgets part of current-iteration are editable - thus, you don't need to care about the full frame body, just the current-iteration's widgets.

> 1a) It is needed to assign some IDs to the field-group to have a possibility to distinguish field-groups. How to do it better?

What is the goal to assign IDs to field-groups? This looks like something related to the current-iteration changes...

> 2) Nested frames. The problem is that there are no field frame in GenericFrame. It need for the support of the linkage between frames: frame frame2:frame = frame frame1:handle. (frame:frame property).

How does this code convert now?

About your changes:
CommonFrame

- setFrame - check how the frame f:handle = h. and frame f:frame convert now in P2J.
- getFirstTabItem, setFirstTabItem - yes, they are needed, as they correspond to the FIRST-TAB-ITEM attribute. Have you found other way?
- moveBeforeTabItem - do you have any conversion tests for this method? How did you find that the method requires two parameters, not one? Or is it just for internal usagE?

CommonWidget

- moveBeforeTabItem - does the signature really not match the one in CommonFrame?

ScrollPane has no real changes, just a history entry

Before commenting any more on the logic, please commit the testcases you are using. You need to keep in mind that the tab order can be altered by the UPDATE statement (and ENABLE I think), is not set just by the frame definition.

More: what happens when the move-before-tab-item and the like are used in triggers?

#47 - 05/27/2014 04:15 PM - Vadim Gindin

First of all I actualized testcases. See folders MOVE_BEFORE_TAB_ITEM and first_tab_item.

Constantin Asofiei wrote:

> Vadim Gindin wrote:
>
> > Here is the first working variant. It works for simple cases (without nested frames and with only one field-group). Following difficulties:
> > 1) The linkage on field-groups requires the logic on the server, because it is exist only there. Commonly speaking there are not the one tab order for the frame, but probably each tab order for the each field-group.
>
> Have you found that a frame can have more than one field-group (except for the down frame)? I mean, a way to group the widgets in a frame, in separate field-groups...

No I don't have such test. I think it's not possible.

> Is it needed to test down-frame?!
>
> Yes, do some tests. For now, assume that only the widgets part of current-iteration are editable - thus, you don't need to care about the full frame body, just the current-iteration's widgets.
>
> 1a) It is needed to assign some IDs to the field-group to have a possibility to distinguish field-groups. How to do it better?
>
> What is the goal to assign IDs to field-groups? This looks like something related to the current-iteration changes...

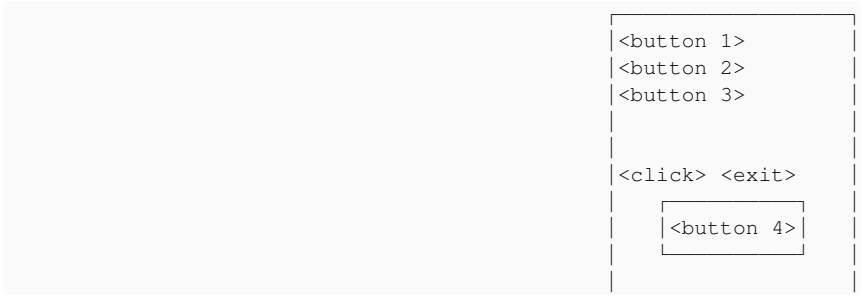
The goal is to have a possibility to keep track of field-groups tab order separately. I want to add some sort of Map<FieldGroup, TabOrder> to the ScreenDefinition for that.

> 2) Nested frames. The problem is that there are no field frame in GenericFrame. It need for the support of the linkage between frames: frame frame2:frame = frame frame1:handle. (frame:frame property).
>
> How does this code convert now?

frame2.setFrameHandle(frame1) where frame1 is the FrameWidget or handle.

Actually I found that FrameWidget has the frame property, that can be used for that case. I only have to add setFrameHandle and getFrameHandle to GenericFrame. Those implementation would delegate calls to corresponding FrameWidget found as a result of the call asWidget().

I also found that this property is not implemented properly at this moment. Here are the screens showing the difference in behavior of Progress and P2J. Progress:



```
                                                    └──────────────────┘
```

and P2J:

```
    ┌──────────────────┐
    │                  │
    │<button 1>        │
    │<button 2>        │
    │<button 3>        │
    │                  │
    │                  │
    │<click> <exit>    │
    │                  │                          ┌──────────┐
    │                  │                          │<button 4>│
    │                  │                          └──────────┘
    │                  │
    │                  │
    │                  │
    │                  │
    └──────────────────┘
```

The file is *working_nested_frame.p*.
In a P2J there are 2 separate screens for the main and nested frames. Nested frame contains only button4. Please take a look.

Should I implement corresponding behavior of frame:frame property?

> About your changes:
> CommonFrame
>
> - setFrame - check how the frame f:handle = h. and frame f:frame convert now in P2J.
> - getFirstTabItem, setFirstTabItem - yes, they are needed, as they correspond to the FIRST-TAB-ITEM attribute. Have you found other way?

The point is that these methods are applicable only to field-group (as written in docs). At the same time MOVE-BEFORE-TAB-ITEM is applicable to widget and takes another widget in the same field-group as a parameter.
But getFirstTabItem, setFirstTabItem are added to CommonFrame, GenericWidget and so on.

> - moveBeforeTabItem - do you have any conversion tests for this method? How did you find that the method requires two parameters, not one? Or is it just for internal usagE?

Please take a look at MOVE_BEFORE_TAB_ITEM subfolder for corresponding testcases. Methods with different signatures was added only for internal usage.

> CommonWidget
>
> - moveBeforeTabItem - does the signature really not match the one in CommonFrame?
>
> ScrollPane has no real changes, just a history entry
>
> Before commenting any more on the logic, please commit the testcases you are using. You need to keep in mind that the tab order can be altered by the UPDATE statement (and ENABLE I think), is not set just by the frame definition.

Current implementation uses existing collection AbstractContainer.widgets for the tab order. I think I will no need to process UPDATE and ENABLE separately..

> More: what happens when the move-before-tab-item and the like are used in triggers?

There will be no difference. The testcases use triggers.

#48 - 05/28/2014 12:58 PM - Constantin Asofiei

Vadim Gindin wrote:

> Actually I found that FrameWidget has the frame property, that can be used for that case. I only have to add setFrameHandle and getFrameHandle to GenericFrame. Those implementation would delegate calls to corresponding FrameWidget found as a result of the call asWidget().
>
> I also found that this property is not implemented properly at this moment.

Do you need this property to be implemented for current task? If is not a show-stopper, leave it for later.

Sorry, I didn't look at your tests, but please take a look at one of mine:

```
def var i1 as int.
def var i2 as int.
def var i3 as int.

form i1 i2 i3 with frame f1.

i1:move-before-tab-item(i3:handle in frame f1) in frame f1.

enable all with frame f1.

on 1 anywhere do:
  i1:move-before-tab-item(i3:handle in frame f1) in frame f1.
end.

on 2 anywhere do:
   update i1 i2 i3 with frame f1.
   enable  all with frame f1.
   message "Done 2".
end.

wait-for go of frame f1.
```

The following can be observed:

1. if an ENABLE is executed after a move-before-tab-item all the tab ordering will be set to the enable's order.
2. if an ENABLE is executed before a move-before-tab-item, the tab ordering will consider the move-before-tab-item.
3. choose "no" at prompt, press 2, 1, F1 (once, to exit the trigger) then check the tab order: the move-before-tab-item called in a child event listening loop survives to the parent event listening loop. This is good news, as we don't need to scope the tab ordering.

Now about implementation: as I can tell, tab ordering has nothing to do with the widget chaining of the FIELD-GROUP's children. Instead, this is something very UI specific and I don't think it should be maintained on the server-side (unless you have some objections to this). So, try this:

1. when move-before-tab-item, a special command will be sent down to the client side (via pushScreenDefinition), to re-adjust the ordering of these two widgets (in AbstractContainer.widgets).
2. when first-tab-item or next-tab-item is called, this will always interrogate the client-side.
3. about the field-group - when move-before-tab-item is called, you can check on server-side for the widgets to be in the same field-group.

Let me know if anything looks wrong to you.

#49 - 05/29/2014 07:26 AM - Vadim Gindin

Constantin, thank you for the good test. It really will be useful.

Here are the problems I can formulate at this moment.

1) The current implementation holds tab order in ScreenDefinition that is transfered from server to client in pushScreenDefinition method. This ScreenDefinition is processed on the client side and affects AbstractContainer.widgets collection. So If I correctly understand, you would want me to get rid of ScreenDefinition.tabOrder collection. OK, I'm trying that. Correspondingly I want to ask. Is it possible to run several calls of tab order changing methods before pushing screen definition to the client? I.e. one each call of moveBeforeTabItem must be ended with pushScreenDefinition or not? I think it is not possible, but I'm not sure.

If it really not possible, than I can hold only one tab order change in ScreenDefinition. If it is possible, than I will have to hold a collection of changes in ScreenDefinition. In that case there will be no sense in comparing with the current iteration.

2) Field groups.
moveBeforeTabItem, getNextTabItem. In that case It is really sufficient and can be implemented. I'm checking that both widgets are in the same field-group on the server side and then pushing screen definition with this order change to the client.
getFirstTabItem, setFirstTabItem. You are writing that I should interrogate the client when these methods are called. I recall that these methods can be applied only to Field-group (see the documentation), not to frame and not to other widgets. There is no field-group component representation on the client side, so I don't understand how can I delegate these methods to the client. The real tab order is the order of widgets in the AbstractContainer.widgets collection on the client side. And there are no information about field-groups.

3) About interrogating of the client-side. The ScreenDefinition.widgets contains the widgets component configs. When the widget getters are accessed, for example getLabel it delegates call to corresponding component config without calling of the client-side. Why can't we do the same in getFirstTabItem and getNextTabItem?

4) Meanwhile, the meaning of methods *getFirstTabItem, setFirstTabItem* is to operate some widget, that will be defined depending on its field-group. Commonly speaking, getFirstTabItem, applied to some field-group, have to get its tab order (not frame tab order) and extract the first element from there.

ScreenDefinition.widgets on server side and AbstractContainer.widgets do not know anything about field-groups, so it is unclear how to be in this situation. Two ways are possible:
a) to take into account, that on frame always contains only one field-group and we can assume the field-group as a frame. In that case, we will have to study the down-frames case.
b) Implement field-groups support in ScreenDefinition on the server side (some additional data structure) and on the client-side (unclear how).

4) Once again. The current implementation works for the cases without nested frames (frame f1:frame=f2) and with assuming that field-group can be assumed as a frame. I don't know if it is "show-stopper" or it is not.

What do you think?

#50 - 05/29/2014 01:11 PM - Constantin Asofiei

Vadim Gindin wrote:

> Constantin, thank you for the good test. It really will be useful.
>
> Here are the problems I can formulate at this moment.
>
> 1) The current implementation holds tab order in ScreenDefinition that is transfered from server to client in pushScreenDefinition method. This ScreenDefinition is processed on the client side and affects AbstractContainer.widgets collection. So If I correctly understand, you would want me to get rid of ScreenDefinition.tabOrder collection. OK, I'm trying that. Correspondingly I want to ask. Is it possible to run several calls of tab order changing methods before pushing screen definition to the client? I.e. one each call of moveBeforeTabItem must be ended with pushScreenDefinition or not? I think it is not possible, but I'm not sure.

ScreenDefinition.tabOrder duplicates what AbstractContainer.widgets does on client-side, so is best to remove it and leave all logic on client side. I don't think pushScreenDefinition should be used - instead, call a new API (defined via ClientExports, implemented by LogicalTerminal on server-side and ThinClient on client-side): this API will immediately adjust the order in AbstractContainer.widgets, and will not be delayed (as you do now, via pushScreenDefinition). Same for the other cases (FIRST/LAST/NEXT/PREV-TAB-ITEM) - there will be ClientExports APIs, to send the call to the client-side, where the actual implementation will be.

> 2) Field groups.
> *moveBeforeTabItem*, getNextTabItem. In that case It is really sufficient and can be implemented. I'm checking that both widgets are in the same field-group on the server side and then pushing screen definition with this order change to the client.

NEXT-TAB-ITEM and PREV-TAB-ITEM apply to normal widgets, not field-groups. FIRST-TAB-ITEM, LAST-TAB-ITEM and MOVE-BEFORE-TAB-ITEM are related only to FIELD-GROUP's.

> *getFirstTabItem, setFirstTabItem*. You are writing that I should interrogate the client when these methods are called.

I think something like this can be done: on server-side, collect a set of widget IDs which are part of this FIELD-GROUP or the same FIELD-GROUP as the widget on which NEXT/PREV-TAB-ITEM is called. A client-side will be invoked and two parameters will be sent: the set of widget IDs (part of this field-group) and the current widget ID (in case of PREV/NEXT-TAB-ITEM); the client-side will search the tab-order, but only using this subset of widgets (to be limited to the FIELD-GROUP) - it will ignore other widgets not part of this field-group.

> 3) About interrogating of the client-side. The ScreenDefinition.widgets contains the widgets component configs. When the widget getters are accessed, for example getLabel it delegates call to corresponding component config without calling of the client-side. Why can't we do the same in getFirstTabItem and getNextTabItem?

I'm not sure what you mean here. The component configs will not be involved when invoking NEXT-TAB-ITEM and PREV-TAB-ITEM... After some preprocessing, to i.e. identify all widgets part of the same field-group, an API call will be sent to the client-side, where the actual implementation resides.

> 4) Meanwhile, the meaning of methods *getFirstTabItem, setFirstTabItem* is to operate some widget, that will be defined depending on its field-group. Commonly speaking, getFirstTabItem, applied to some field-group, have to get its tab order (not frame tab order) and extract the first element from there.

See above: the client side will receive a set of widgets composing the FIELD-GROUP, and will act on that.

> ScreenDefinition.widgets on server side and AbstractContainer.widgets do not know anything about field-groups, so it is unclear how to be in this situation. Two ways are possible:

See above about the 3rd way: all APIs work on a subset of widgets, part of the referenced FIELD-GROUP.

> 4) Once again. The current implementation works for the cases without nested frames (frame f1:frame=f2) and with assuming that field-group can be assumed as a frame. I don't know if it is "show-stopper" or it is not.

OK, but what I don't like is duplicating logic... which from experience will mean lots of headaches in the future. The nested frame case can be left for later (make sure to leave behind some TODO's/comments).

#51 - 06/02/2014 05:02 PM - Vadim Gindin

- File vig_upd20140602a.zip added

I corrected implementation corresponding to the last notes. Please take a look.

#52 - 06/03/2014 02:31 AM - Constantin Asofiei

Review for 0602a:

1. when we have APIs exposed via network servers (i.e. ClientExports), is best to avoid (de)serializing of java objects: instead of using i.e. ArrayList to send data to the other side, use Utils.integerCollectionToPrimitive to get an int array from a collection and send that to the other side; is a lot faster to transport it. On the other side, you can convert it to either set or list, as needed (depending on how you use it).
2. please double check all files for how you merged them: some of them have the headers merged incorrectly.
3. ClientExports.getFirstTabItem - doesn't this require a list of widgets (part of the target field-group), from which the first is needed? Assuming the frame can have more than one field-group...
4. GenericFrame.setFrameHandle, using handle.unwrapFrame
   - these APIs are now instance methods, not static
   - they always must be followed by a API call - you can't leave them "hanging" - see the javadoc
   - for this particular case, you don't need to call handle.unwrapFrame. Instead, use handle.getResource to get the WrappedResource instance, and before calling asWidget().setFrame, ensure that the resource is actually a frame widget. Thus, you don't really need handle.unwrapFrame, so please remove it (as your update added it).
5. Frame.addWidget I don't think is ever used - please make sure to not leave behind unused APIs (added by your update), imports, etc.
6. WidgetRegistry has no real changes

Otherwise, the logic looks good. If your tests work, I think all is left is address the review issues, cleanup the code (javadoc/formatting/etc) and put it in regression testing.

#53 - 06/03/2014 02:33 AM - Constantin Asofiei

PS: make sure to merge your update with the latest bzr version.

#54 - 06/06/2014 09:47 AM - Vadim Gindin

- File vig_upd20140606a.zip added

The update passed regression testing and have committed to bzr. Rev #10545

#55 - 06/06/2014 10:20 AM - Constantin Asofiei

Vadim Gindin wrote:

> The update passed regression testing and have committed to bzr. Rev #10545

Some issues about this update:

1. the header merge for Frame.java and GenericFrame.java is messed up
2. you left a TODO in FieldGroup.getFirstTabItem - is this necessary or was just a remark?
3. ClientExports.getFirstTabItem, getNextTabItem - aren't these supposed to work in the limits of a field-group? Currently, your implementation works with the entire frame. And this was a question asked in the review at note 52.
4. ThinClient.moveBeforeTabItem - this is not using fgWidgetIds at all, is not needed for the server side to send it to the client.

That's why there is a final review before releasing an update...

#56 - 06/07/2014 07:44 AM - Vadim Gindin

- File vig_upd20140607a.zip added

Haste makes waste - It's about me in this case. I'm sorry.

I'm working on your notes. I'm really forgot about field-groups in first-tab-item methods.

Going further. Working on this methods a faced with a problem with widgets IDs similar to the problem in the task about current-iteration (#2162). I have one test (*first_tab_item/down_frame/current.p*) with 4-down frame. Each row of it contains 2 FillIns. During the displaying of them I wrote the first and the second field-groups to special variables, and after all I'm trying to get first-tab-item from the both of field-groups. BTW none of these field-groups are current. getFirstTabItem methods returns the same field for both field-groups. That is the problem.

The reason is following: on the client side there are different widgets for the same fields in different rows, but these widgets have same IDs. I.e. there are 4 couples of FillIns with IDs 1001 and 1002.

I prepared the next update (under my previous), that is also prints in console client-side widgets in this manner:

```
1003: Skip com.goldencode.p2j.ui.client.Point@532071 com.goldencode.p2j.ui.client.Dimension@177134d
-1: com.goldencode.p2j.ui.chui.LabelImpl@1001fd2
-1: com.goldencode.p2j.ui.chui.LabelImpl@1292f2d
1002: FillIn com.goldencode.p2j.ui.client.Point@942e36 com.goldencode.p2j.ui.client.Dimension@6613f1 label = f
ld2 value =
1001: FillIn com.goldencode.p2j.ui.client.Point@1a75545 com.goldencode.p2j.ui.client.Dimension@cf439b label =
fld1 value =
1002: FillIn com.goldencode.p2j.ui.client.Point@15fa9d8 com.goldencode.p2j.ui.client.Dimension@187bd3a label =
 fld2 value =
1003: Skip com.goldencode.p2j.ui.client.Point@986c3f com.goldencode.p2j.ui.client.Dimension@eab0eb
1001: FillIn com.goldencode.p2j.ui.client.Point@52c5d com.goldencode.p2j.ui.client.Dimension@1a3a752 label = f
ld1 value =
1002: FillIn com.goldencode.p2j.ui.client.Point@13e32c7 com.goldencode.p2j.ui.client.Dimension@77bf0c label =
fld2 value =
1003: Skip com.goldencode.p2j.ui.client.Point@19fbb6d com.goldencode.p2j.ui.client.Dimension@1ca2f7c
1001: FillIn com.goldencode.p2j.ui.client.Point@f3aa19 com.goldencode.p2j.ui.client.Dimension@1246e92 label =
fld1 value =
1002: FillIn com.goldencode.p2j.ui.client.Point@d7443d com.goldencode.p2j.ui.client.Dimension@c13170 label = f
ld2 value =
1003: Skip com.goldencode.p2j.ui.client.Point@194b386 com.goldencode.p2j.ui.client.Dimension@1965eb8
1001: FillIn com.goldencode.p2j.ui.client.Point@124be57 com.goldencode.p2j.ui.client.Dimension@14c749b label =
 fld1 value =
```

Questions.
1) Am I using the right IDs?
2) How to extract right widgets and synchronize them with the server?
3) May be we should delay it as it was in #2162?

P.S. Should I revert my last update or I can just make sequential updates overt it?

#57 - 06/12/2014 06:24 AM - Constantin Asofiei

Vadim Gindin wrote:

> Questions.
> 1) Am I using the right IDs?

Yes, the IDs are right.

> 3) May be we should delay it as it was in #2162?

Yes, this should be done after we determine the extent to which #2162 needs to be implemented. Currently, on the client-side for each widget part of a down frame there is only one widget ID (for al widget occurrences in the down body).

> P.S. Should I revert my last update or I can just make sequential updates overt it?

You can make it sequential, thus treat it like an entire new update (new headers for each modified file, etc).

#58 - 06/13/2014 02:06 AM - Vadim Gindin

- File vig_upd20140613a.zip added

I prepared the next update that contains last corrections corresponding to your last notes. I did not add new header to GenericFrame and AbstractContainer, because changes of these files are only in headers. Should I add headers there too? I also added TODO in getFirstTabItem method about widget IDs problem. Take a look please.

#59 - 06/16/2014 01:24 PM - Greg Shah

Code Review 0613a

The changes look fine. Please get this runtime regression tested.

#60 - 06/17/2014 02:34 AM - Constantin Asofiei

- File vig_upd20140617a.zip added

Vadim, please use this version, the headers in Frame.java and GenericFrame.java are fixed now.

#61 - 06/24/2014 12:35 AM - Vadim Gindin

The latest update passed regression testing. Can I commit it?

#62 - 06/24/2014 12:55 AM - Greg Shah

Yes, please commit it and distribute it to the team.

#63 - 06/24/2014 02:56 AM - Vadim Gindin

Committed vig_upd20140617a.zip. Rev #10550

#64 - 03/23/2016 12:01 PM - Greg Shah

- Status changed from New to Closed

#65 - 11/16/2016 12:12 PM - Greg Shah

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

Files

evl_upd20130220a.zip	98.4 KB	02/21/2013	Eugenie Lyzenko
ui_meth_test.p.20130220a.zip	388 Bytes	02/21/2013	Eugenie Lyzenko
browse_ui_test0.p	682 Bytes	02/21/2013	Eugenie Lyzenko
conversion_err.log	19.9 KB	02/15/2014	Vadim Gindin
vig_upd20140516a.zip	150 KB	05/16/2014	Vadim Gindin
vig_upd20140526a.zip	152 KB	05/26/2014	Vadim Gindin
vig_upd20140602a.zip	344 KB	06/02/2014	Vadim Gindin
vig_upd20140606a.zip	330 KB	06/06/2014	Vadim Gindin
vig_upd20140607a.zip	301 KB	06/07/2014	Vadim Gindin
vig_upd20140613a.zip	301 KB	06/13/2014	Vadim Gindin
vig_upd20140617a.zip	303 KB	06/17/2014	Constantin Asofiei