

User Interface - Feature #2614

Feature # 2252 (Closed): implement GUI client support

implement GUI runtime support for RADIO-SET widget

07/23/2015 01:08 PM - Greg Shah

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Eugenie Lyzenko	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	GUI Support for a Complex ADM2 App	<b>version:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Related issues:</b>			
Related to User Interface - Feature #2676: implement the equivalent to config...			<b>Closed</b>

History

#1 - 07/23/2015 01:25 PM - Greg Shah

This task is for full implementation of the GUI widget implementation for RADIO-SET. Both static and dynamic usage is needed.

Make sure that the existing widget options, attributes and methods are fully supported in the GUI runtime code.

#2 - 09/20/2015 06:51 AM - Greg Shah

- Assignee changed from Igor Skornyakov to Eugenie Lyzenko

#3 - 09/23/2015 11:25 AM - Eugenie Lyzenko

Created task branch 2614a from trunk revision 10939.

#4 - 09/23/2015 02:31 PM - Eugenie Lyzenko

Make sure that the existing widget options, attributes and methods are fully supported in the GUI runtime code.

Looks like currently in P2J radio-set supports only CHARACTER internal data type. Do we need support for the rest: LOGICAL, INTEGER, DECIMAL, DATE and INT64?

Asking because I need to make work timetable to be in time and calculate the efforts.

#5 - 09/23/2015 03:12 PM - Greg Shah

I have not checked all places in the app code, but my quick looks shows that CHARACTER, INTEGER and LOGICAL are all in use. I think we need to expect complete data type support so that we are not surprised later.

I also see that they allow assigning unknown value as one of the choices like this:

```
define var bogus as logical
  VIEW-AS RADIO-SET HORIZONTAL
    RADIO-BUTTONS
      "Don't Know", ?,
      "Yes", yes,
      "No", no
    SIZE ...
```

#### #6 - 09/23/2015 10:03 PM - Eugenie Lyzenko

Task branch 2614a for review updated to revision 10940.

Just prepare to the separation of the GUI code. For now I'm inclining to make GUI specific class for RadioSet too, not only for RadioButtons. Because radio buttons init/key processing are different in GUI. Continue working.

#### #7 - 09/24/2015 09:07 PM - Eugenie Lyzenko

Task branch 2614a for review updated to revision 10941.

Several fixes and more preparation to the having prototype. Adding calls to load SVG images for button check mark state controls. Prepare to rebase.

#### #8 - 09/24/2015 09:07 PM - Eugenie Lyzenko

Rebased task branch 2614a from P2J trunk revision 10940, new branch revision is 10942.

#### #9 - 09/24/2015 09:33 PM - Eugenie Lyzenko

Strange shadow corner started to draw after rebase with 10940. Please see the picture, problematic area is in red. Does not disappear even suppress the RadioSet painting. Some clip issue from container used as base for RadioSet?

#### #10 - 09/24/2015 09:34 PM - Eugenie Lyzenko

- File *radio\_set\_test0\_p2j\_gui\_20150924.jpg* added

The picture.

#### #11 - 09/25/2015 05:00 AM - Greg Shah

Starting with rev 10937, there were substantial changes to container processing, scrolling, borders... this could have been introduced there. See [#1811-1276](#) for an example.

#### #12 - 09/25/2015 05:17 AM - Greg Shah

Then again, you were already based off of 10939 before the recent rebase, so this must be due to some change in 10940. There were some clipping fixes in OutputManager to deal with regressions from 10937. There was a change in ScrollPane viewport sizing. There were coordinate and clipping changes in TopLevelWindow. There were drawing and location changes in BorderedPanelGuiImpl. There were changes in border drawing for LineBorderGuiImpl. There were sizing and scrollbar processing changes in ScrollPaneGuiImpl. There were drawing, z-order and other changes in WindowGuiImpl. There were serious clipping changes in AbstractGuiDriver.draw(). There were sizing changes in NativeScrollContainer.

Those are the most obvious possibilities. I haven't seen this particular behavior in other widgets.

**#13 - 09/25/2015 05:51 AM - Constantin Asofiei**

Eugenie Lyzenko wrote:

Strange shadow corner started to draw after rebase with 10940. Please see the picture, problematic area is in red. Does not disappear even suppress the RadioSet painting. Some clip issue from container used as base for RadioSet?

This one is most likely related to PaintEvent's generated for incorrect coordinates. It was exposed by the AbstractGuiDriver.draw() clip changes. I'm currently trying to chase down and solve all these issues - there are others related to scrollbar buttons when pressed, their clipped/repainted region is not the same as the area that is being drawn.

Also, about combo's drop-down: the inner scrollpane has a width wider than the viewport's width, which triggers drawing issues in the Web driver (the outer rectangle for a highlighted item in the drop-down "leaks" outside the drop-down). Although this is related to scrollpane's width too, the JS code should not have drawn outside of the clipping region.

**#14 - 09/25/2015 07:15 AM - Eugenie Lyzenko**

- *File radio\_set\_test0\_p2j\_gui\_20150925.jpg added*

Greg Shah wrote:

Those are the most obvious possibilities. I haven't seen this particular behavior in other widgets.

Thanks for help. I've found the issue. I had to fill the background of the radio-set. Now the problem is gone(see the picture).

**#15 - 09/25/2015 05:21 PM - Eugenie Lyzenko**

- *File radio\_set\_test1\_1\_p2j\_gui\_20150925.jpg added*

- *File radio\_set\_test1\_0\_4gl\_gui.jpg added*

- *File radio\_set\_test1\_1\_4gl\_gui.jpg added*

- *File radio\_set\_test1\_0\_p2j\_gui\_20150925.jpg added*

Task branch 2614a for review updated to revision 10943.

This is the first working prototype for review. Under active development so not cleaned from commented out code. Handles the keyboard and mouse selection, enabled/disabled state, added the SVG images to draw the check marks. The screenshots to compare with original 4GL pictures attached here.

The another issue I've found is the wrong frame title text color.

Continue working for different modes(horizontal) support, different data types and sizing.

#### #16 - 09/28/2015 09:38 PM - Eugenie Lyzenko

Task branch 2614a for review updated to revision 10944.

Adding support for horizontal radio-set and expand attributes. Also checked to have correct width/height in chars/pixels. The code is in active development so not cleaned from commented code.

Continue working on the size phrase processing.

#### #17 - 09/29/2015 10:58 AM - Eugenie Lyzenko

Rebased task branch 2614a from P2J trunk revision 10941, new branch revision is 10945.

#### #18 - 09/29/2015 09:43 PM - Eugenie Lyzenko

Task branch 2614a for review updated to revision 10946.

The update adds changes to support size phrase handling in radio-set for characters/pixels. The tests shows identical width/height attributes value for characters/pixels in 4GL and P2J.

Continue working. The remaining main parts are to implement dynamic widget support and different data type handling. And final testing of course.

#### #19 - 09/30/2015 03:10 PM - Eugenie Lyzenko

Task branch 2614a for review updated to revision 10947 and 10948(this is just syntax change for comment).

This update completes dynamically created radio-set support. Had to change the RadioSetWidget class to distribute enable flag to the included radio-buttons for dynamic widget. So the dynamically created radio-set is now working properly.

Continue working on radio-set to implement other data types support.

#### #20 - 09/30/2015 09:02 PM - Eugenie Lyzenko

The data types support investigation shows in general radio-set handles the not only character data. But there are some issues:

1. For logical data type the unknown value(?) is not supported on conversion level at view-as phrase. The code:

```
define var bogus as logical
  VIEW-AS RADIO-SET HORIZONTAL
  RADIO-BUTTONS
    "Don't Know", ?,
    "Yes", yes,
    "No", no
```

generates exception:

EXPRESSION EXECUTION ERROR:

```
tw.graft("one_item_init", null, node.id, "labtype", "character", "lvartype", "STRING", "lvarvalue", v1, "datty
pe", execLib("get_type", ref2), "ldatatype", litMap.get(ref2.type), "ldatvalue", v2)
  ^ { No replacement value found for key:  ldatatype }
```

...

java.lang.RuntimeException: ERROR! Active Rule:

RULE REPORT

```

Rule Type :    WALK
Source AST:  [ block ] BLOCK/ @0:0 {8508330213377}
Copy AST   :  [ block ] BLOCK/ @0:0 {8508330213377}
Condition  :  res == null and (evalLib("widgettype", ref.type) or ref.type == prog.symbol)
Loop       :  false
--- END RULE REPORT ---

```

2. Using int64 data type does not allow to set up big integers as initial radio-button value. The constant 111000000000 produces java compilation error due to it is considering as int, not long.

3. The generated new ControlSetItem(new character("One"), new integer(value) for radio-button item does not use int64 as expected, using integer instead.

#### #21 - 10/01/2015 12:19 PM - Eugenie Lyzenko

Found a way to support unknown(?) data initial for radio-button.

The remaining issue is how to add "L" to the int64 literal for respective radio button item:

```

...
rs.setItems(new ControlSetItem[]
{
    new ControlSetItem(new character("One"), new int64(111000000000)),
    new ControlSetItem(new character("Two"), new int64(222000000000)),
    new ControlSetItem(new character("Three"), new int64(333000000000))
}, true);
...

```

the correct code should be:

```

...
rs.setItems(new ControlSetItem[]
{
    new ControlSetItem(new character("One"), new int64(111000000000L)),
    new ControlSetItem(new character("Two"), new int64(222000000000L)),
    new ControlSetItem(new character("Three"), new int64(333000000000L))
}, true);
...

```

The point [#3](#) from previous message is false alarm. Everything is OK there.

**#22 - 10/01/2015 12:39 PM - Greg Shah**

The remaining issue is how to add "L" to the int64 literal for respective radio button item

See rules/convert/literals.rules and search on use64BitInt. The support is already there, but it is not being used for some reason.

**#23 - 10/01/2015 04:50 PM - Eugenie Lyzenko**

Greg Shah wrote:

See rules/convert/literals.rules and search on use64BitInt. The support is already there, but it is not being used for some reason.

Task branch 2614a for review updated to revision 10949.

This update adds unknown value support for radio-buttons.

Also the fix for int64 literals initialization for radio-button item. The investigation shows this special case is just not handling by rules/convert/literals.rules. I think because it is neither variable init value(which is handling properly in literals.rules) nor assignment. So I suggest to put respective change in radio-button init code. Let me know if any objections.

Continue working on different data type processing for radio-set.

**#24 - 10/01/2015 06:10 PM - Greg Shah**

The investigation shows this special case is just not handling by rules/convert/literals.rules. I think because it is neither variable init value(which is handling properly in literals.rules) nor assignment.

That code also handles any literals in expressions. So it handles virtually all uses.

So I suggest to put respective change in radio-button init code. Let me know if any objections.

I prefer not to put such code in lots of places throughout our rules. I know we do some conversion of literals in the set\_attr\_ex function (see common-progress.rules). It is something I really don't like but we haven't spent the time to fix it.

Can you please provide some more details about what specific TRPL code is doing the conversion in this radio-set case?

## #25 - 10/01/2015 07:25 PM - Eugenie Lyzenko

The sample code to convert is:

```
...
define variable Rs as int64
    view-as radio-set radio-buttons
        "One", 111000000000, "Two", 222000000000, "Three", 333000000000
        initial 222000000000.
...
```

The rule to convert is frame\_generator.xml function gen\_vas\_phrase where all VIEW-AS phrases are processing. For LIST-ITEMS, LIST-ITEM-PAIRS and RADIO-BUTTONS another function gen\_set\_items is called to generate the ControlSetItem[] array with proper data item with initial value. The function gen\_set\_items takes the pair(string label, initial constant) and construct the new item like new ControlSetItem(new character("One"), new int64(111000000000)) according to data type and constant value. The constant can be STRING, NUM\_LITERAL, DEC\_LITERAL, DATE\_LITERAL, UNKNOWN\_VAL. But if the type is NUM\_LITERAL it is possible to have either integer or int64.

I did put the debugging println calls in rules/convert/literals.rules rule file and looks like radio-buttons clause is not going through place where other NUM\_LITERAL cases are processing(even for simple condition type == prog.num\_literal). That's why I decided to use gen\_set\_items function local change. This should work for all list, item pair with VIEW-AS phrase.

## #26 - 10/02/2015 05:58 AM - Greg Shah

OK, go ahead with the change.

## #27 - 10/02/2015 06:23 AM - Eugenie Lyzenko

Task branch 2614a for review updated to revision 10950.

Adding change clarification and some NPE protection logic for int64 constant related change. Preparing to rebase.

## #28 - 10/02/2015 07:09 AM - Eugenie Lyzenko

Rebased task branch 2614a from P2J trunk revision 10942, new branch revision is 10951.

## #29 - 10/02/2015 07:17 PM - Eugenie Lyzenko

Task branch 2614a for review updated to revision 10952.

The change fixes the runtime error for int64 values outside the current format. The radio-set should call format ignoring condition in this case. Also small fix to consider pixel based button width/height for vertical/horizontal widget as the same value(the longest button) to simplify button drawing.

So far the implementation is completed in base. Need to spend some time (1 day) to perform additional testing and fix any issues found(may be 1 day more). Then when everything is OK I think we can out this into testing. That's the near days plan for radio-set.

### #30 - 10/05/2015 09:13 PM - Eugenie Lyzenko

Task branch 2614a for review updated to revision 10953.

The change is a result of the final testing. Contains many small fixes for radio-button size calculation in different cases, dynamic widget logic and image type to draw in mouse pressed state.

Preparing to rebase with recent trunk. And if there are no notes/objections - I guess we can put this version into regression testing cycle.

### #31 - 10/05/2015 10:02 PM - Eugenie Lyzenko

Rebased task branch 2614a from P2J trunk revision 10943, new branch revision is 10954.

Prepared for possible testing if no notes or issues to fix.

### #32 - 10/06/2015 08:32 AM - Greg Shah

Code Review Task Branch 2614a Revision 10954

1. This code in frame\_generator.xml is not generic enough:

```
<rule>ref2.type == prog.num_literal and  
    ref.parent.parent.getPrevSibling() != null and  
    ref.parent.parent.getPrevSibling().getFirstChild() != null and  
    ref.parent.parent.getPrevSibling().getFirstChild().type == prog.kw_int64)
```

The problem is that the DEFINE VARIABLE is not guaranteed to always use AS INT64. There can also be a LIKE some-var or LIKE some-field.

The safer way to do this is to read the "type" annotation from the variable definition node (top-level node) and compare it with prog.var\_int64. Look at the tree for DEFINE VARIABLE to see what I mean.

2. Please move the RadioSet.selectButton() down below the public methods (and above the private methods).
3. If the commented code is not needed in RadiosSet.refreshItems(), please remove it.
4. In GuiWidgetFactory the com.goldencode.p2j.util imports should use the wildcard.
5. Some of the comments in the class javadoc for RadioSetGuilmpl seem to be inaccurate (copied from the parent class).
6. Is it possible in Windows to have non-default colors for drawing the buttons/highlights? Such a feature would make it trickier to use SVG images (although we could probably "edit" them after loading and before rendering as in-memory images).
7. Please post some example screenshots for some representative testcases (both horizontal and vertical) that show P2J and the same 4GL code running on windev01.



### #33 - 10/06/2015 10:10 AM - Eugenie Lyzenko

4. In GuiWidgetFactory the com.goldencode.p2j.util imports should use the wildcard.

The reason I did this is issue with Text class used in this file. Importing ....util.\* gives the compilation error:

```
...
[javac] /home/evl/timco_new/p2j/src/com/goldencode/p2j/ui/client/gui/GuiWidgetFactory.java:93: error: refe
rence to Text is ambiguous
[javac]         classMap.put(Text.class, TextGuiImpl.class);
[javac]             ^
[javac]     both class com.goldencode.p2j.util.Text in com.goldencode.p2j.util and class com.goldencode.p2j.
ui.client.Text in com.goldencode.p2j.ui.client match
[javac] 1 error
...
```

We have two different classes with same name in different packages.

### #34 - 10/06/2015 10:28 AM - Greg Shah

It should look like this:

```
import com.goldencode.p2j.util.*;

// explicit import to resolve conflict with com.goldencode.p2j.util.Text
import com.goldencode.p2j.ui.client.Text;
```

### #35 - 10/06/2015 12:47 PM - Eugenie Lyzenko

- File 4gl\_screens.zip added

- File p2j\_screens.zip added

7. Please post some example screenshots for some representative testcases (both horizontal and vertical) that show P2J and the same 4GL code running on windev01.

The radio-set testcases has been uploaded into bzt repo. Revision 1373. The p2j-4gl screenshots attached here.

**#36 - 10/06/2015 05:07 PM - Greg Shah**

The screenshots look good. When do you plan to check in the version with code review changes and any remaining needed fixes?

**#37 - 10/06/2015 06:43 PM - Eugenie Lyzenko**

Greg Shah wrote:

The screenshots look good. When do you plan to check in the version with code review changes and any remaining needed fixes?

In a 10 minutes I guess.

**#38 - 10/06/2015 06:55 PM - Eugenie Lyzenko**

Task branch 2614a for review updated to revision 10955.

Notes resolution([#1-#5](#)). For the first one I used not exact approach as you offered but similar one. Getting "classname" annotation and comparing to "int64". I've struggled with `#{int}getNoteLong("type") prog.var_int64` and `#{int} ref.getAncestor(3).getAnnotation("type") prog.var_int64` and it causes nothing and conversion exception respectively(at least the second is strange because the uploaded version works fine).

The note [#6](#) is under investigation. I'll report some thoughts later today.

**#39 - 10/06/2015 08:57 PM - Eugenie Lyzenko**

6. Is it possible in Windows to have non-default colors for drawing the buttons/highlights? Such a feature would make it trickier to use SVG images (although we could probably "edit" them after loading and before rendering as in-memory images).

Some considerations:

1. There are several themes that we need to support. The question: do we need to support dynamic theme change?
2. Currently we use "classic" theme. Different themes have a bit different features(in color changing point of view)
3. Within the "classic" theme it is possible to change system controls colors but not all. The button foreground is not changing.
4. The SVG images we currently use consist of only shapes with 4 colors, foreground(black), background(white), shadow(dark grey) and disabled(pale grey). Other image space is transparent(inherits the background of the surface the image is drawing on). 3 of the 4 above colors can be changed(except image foreground) in Windows I think.

So we have two opportunities I guess:

- substitute colors in SVG images with another "current"
- play with Java XOR+Alpha channel image rendering features and leave the original colors unchanged, this can give some approximation with what we have in Windows.

The second approach needs some investigations in Java rendering feature to find out if we can implement this or not.

How urgent or critical this point is?

**#40 - 10/07/2015 04:48 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

Some considerations:

2. Currently we use "classic" theme. Different themes have a bit different features(in color changing point of view)

In fact the radio set in Windows classic and Windows XP/7 basic theme are so different that they will need different SVG sources.

3. Within the "classic" theme it is possible to change system controls colors but not all. The button foreground is not changing.

There are two ways how the colors can be changed. (1) Through different color schemes (for example high contrast) or (2) by assigning specific color in the theme settings. Specifically the window background color also affects the background color of the radio set (and also editor, combo box, checkbox).

**#41 - 10/07/2015 08:05 AM - Greg Shah**

1. There are several themes that we need to support. The question: do we need to support dynamic theme change?

Not right now. We will implement themes support in task [#2676](#).

Part of this support is making sure that the default system colors swap out to match the chosen theme. The other part is supporting drawing differences. As Hynek notes, the other themes may require different SVG sources. And we will still need to support customer-defined system color overrides for the theme-specific sources.

I asked the question because we need to know the extent to which we will need to modify our SVG drawing for themes support.

substitute colors in SVG images with another "current"

When we implement the solution in [#2676](#), we will read the default SVG from the jar, edit the color definitions inside it (if needed) and then render it.

play with Java XOR+Alpha channel image rendering features and leave the original colors unchanged, this can give some approximation with what we have in Windows

Approximation is not acceptable. Our result must be an exact match.

You don't need to do anything else on this right now. I will add notes to [#2676](#) to ensure that we handle these cases.

#### #42 - 10/07/2015 08:21 AM - Greg Shah

Code Review Task Branch 2614a Revision 10955

The changes look good. One thing:

```
/**
 * RadioSet is a implementation of radio-set widget, a mutually exclusive set
 * of {@link RadioButton} objects.
 * <p>
 * WARNING: this class needs updates to support BaseDataType instances. It
 * is currently hard coded to STRINGS.
 * <p>
 * TODO<SIY>: this class does not require dedicated CHUI/GUI versions because
 * it delegates drawing to buttons, but refreshItems() requires update in order
 * to maintain buttons physical position/size according to GUI rules.
 *
 */
@SuppressWarnings("unchecked")
public class RadioSetGuiImpl
```

Is this text accurate? It seems like it was copied from the parent class. For that matter, I think even the parent class text is no longer be correct. Both the BDT stuff and the comments about CHUI/GUI are wrong now, I think.

#### #43 - 10/07/2015 08:22 AM - Greg Shah

Besides the javadoc thing in note 42, is there anything else that needs to be done before this task is complete? Are there any known bugs or limitations?

#### #44 - 10/07/2015 08:24 AM - Eugenie Lyzenko

In fact the radio set in Windows classic and Windows XP/7 basic theme are so different that they will need different SVG sources.

Consider attached images. From XP classic and Win7 classic. They are the same. I did not mean some "basic" theme. This is particular classic theme came from Windows 95 first release. And this is what currently used in WINDEV01 system.

...Specifically the Window background color also affects the background color of the radio set (and also editor, combo box, checkbox).

Yes but in SVG approach it is not important. Image background for SVG is transparent. Only shape color is the subject of discussion. Open the SVG

radio-button image with Inkscape to see what I mean.

**#45 - 10/07/2015 08:29 AM - Eugenie Lyzenko**

- File *XP\_classic.jpg* added

- File *Win7\_classic.jpg* added

And the pictures...

**#46 - 10/07/2015 08:36 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

In fact the radio set in Windows classic and Windows XP/7 basic theme are so different that they will need different SVG sources.

Consider attached images. From XP classic and Win7 classic. They are the same. I did not mean some "basic" theme. This is particular classic theme came from Windows 95 first release. And this is what currently used in WINDEV01 system.

If I remember correctly we are planning to support the Windows classic theme AND Windows XP/7 basic theme. On windows XP/7 the default isn't classic but basic. The basic is completely different. Not only the appearance but in the customization capabilities.

...Specifically the window background color also affects the background color of the radio set (and also editor, combo box, checkbox).

Yes but in SVG approach it is not important. Image background for SVG is transparent. Only shape color is the subject of discussion. Open the SVG radio-button image with Inkscape to see what I mean.

I was trying to point out that the background color of the classic Windows controls is customizable through the window color in the theme settings. This also affects the radio-set widget.

**#47 - 10/07/2015 09:40 AM - Eugenie Lyzenko**

Task branch 2614a for review updated to revision 10956.

Fixed javadoc comment mentioned in note 42.

Besides the javadoc thing in note 42, is there anything else that needs to be done before this task is complete? Are there any known bugs or limitations?

I think the implementation is complete. There are some issues I still worry about. As you can see I have shifted the radio-set clip are one point left. There is a strange pixel on the background that is cleaned with this fix. Also I have noted the tooltip sometimes produced some screen garbage(or not dismissed on time elapsed). Also the frame title color is still wrong as you can see from the P2J pictures. But all this is not radio-set specific. So I suggest to test/commit this implementation and then we can work on something from this list according to current work priority requirements.

**#48 - 10/07/2015 10:54 AM - Greg Shah**

So I suggest to test/commit this implementation and then we can work on something from this list according to current work priority requirements.

Agreed.

Please start regression testing.

Also, please create a new sub-task under [#2677](#) for the tooltip problem. The frame title is already an issue in there.

**#49 - 10/07/2015 11:43 AM - Eugenie Lyzenko**

Conversion testing is in progress.

**#50 - 10/07/2015 02:55 PM - Eugenie Lyzenko**

Conversion part completed, no regressions, generated code sets are identical. Continue with runtime testing.

**#51 - 10/07/2015 07:10 PM - Eugenie Lyzenko**

The main part completed without regressions, one test had to run in single mode. Continue with CTRL-C part.

The failures: tc\_job\_002, tc\_po\_item\_003, tc\_po\_item\_004.

The last two are also expected, correct?

**#52 - 10/07/2015 08:49 PM - Eugenie Lyzenko**

The CTRL-C tests passed without regressions. The results file: 2614a\_10956\_06069cc\_20151007\_evl.zip.

So I can commit 2614a into trunk if no objections.

#53 - 10/08/2015 07:41 AM - Greg Shah

The last two are also expected, correct?

Yes.

So I can commit 2614a into trunk if no objections.

Yes, please merge to trunk now.

#54 - 10/08/2015 08:34 AM - Eugenie Lyzenko

Branch 2614a was merged to trunk as revno 10944 then it was archived.

#55 - 10/08/2015 08:42 AM - Greg Shah

- Status changed from New to Closed

#56 - 11/16/2016 12:13 PM - Greg Shah

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

Files

radio_set_test0_p2j_gui_20150924.jpg	37.4 KB	09/25/2015	Eugenie Lyzenko
radio_set_test0_p2j_gui_20150925.jpg	38.7 KB	09/25/2015	Eugenie Lyzenko
radio_set_test1_0_4gl_gui.jpg	49 KB	09/25/2015	Eugenie Lyzenko
radio_set_test1_1_4gl_gui.jpg	50.8 KB	09/25/2015	Eugenie Lyzenko
radio_set_test1_0_p2j_gui_20150925.jpg	42.2 KB	09/25/2015	Eugenie Lyzenko
radio_set_test1_1_p2j_gui_20150925.jpg	42.9 KB	09/25/2015	Eugenie Lyzenko
p2j_screens.zip	558 KB	10/06/2015	Eugenie Lyzenko
4gl_screens.zip	637 KB	10/06/2015	Eugenie Lyzenko
Win7_classic.jpg	6.99 KB	10/07/2015	Eugenie Lyzenko
XP_classic.jpg	11.3 KB	10/07/2015	Eugenie Lyzenko