

User Interface - Feature #1789

implement the toggle-box widget

10/30/2012 09:41 AM - Greg Shah

Status:	Closed	Start date:	02/10/2015
Priority:	Normal	Due date:	
Assignee:	Eugenie Lyzenko	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	vendor_id:	GCD
billable:	No		
Description			
Subtasks:			
Feature # 2513: implement GUI runtime support for TOGGLE-BOX			Closed
Related issues:			
Related to User Interface - Bug #2174: Toggle-box widget implementation abnor...		Rejected	
Related to User Interface - Feature #1791: implement dynamic UI support		Closed	02/21/2013 11/08/2013

History

#1 - 10/31/2012 02:54 PM - Greg Shah

- Target version set to Milestone 12

#2 - 05/28/2014 03:54 PM - Greg Shah

- Assignee set to Eugenie Lyzenko

Start by writing testcases to demonstrate the toggle-box behavior and all its options.

Then add conversion support (TRPL rules and the stubbed out Java classes/methods needed to compile).

Finally, implement a new ChUI toggle-box widget with full options support. The implementation will probably be very similar to the Button implementation. It may even make sense to inherit from Button, if the core functionality is similar enough.

For an example, see [#1788](#) for how the rectangle widget was recently added. Of course, toggle-box has interactive behavior so it is more complex, but the idea is the same.

#3 - 05/29/2014 05:52 PM - Eugenie Lyzenko

Start by writing testcases to demonstrate the toggle-box behavior and all its options.

Do we need to cover only ChUI area? Or including Graphical Interface for Windows too?

#4 - 05/30/2014 08:06 AM - Greg Shah

For now, we don't have GUI support yet in the rest of the client. So you can only implement ChUI right now. Later on, we will add GUI support.

#5 - 06/05/2014 07:03 AM - Eugenie Lyzenko

- File *evl_testcases_20140605a.zip* added

After analyzing of the ABL Reference book the testcases to demonstrate the behavior in all possible options according to 4GL documentation have been added. The respective screen shots are also attached for debug purposes.

Shifting to the conversion stage.

#6 - 06/05/2014 09:55 AM - Greg Shah

Please add tests for the following cases:

1. Check if ATTR-SPACE or other options can have an impact on the label positioning. All the labels in your examples are missing a space between the brackets and the label ([]label instead of [] label). This doesn't look right.
2. From the 4GL docs:

You can specify TOGGLE-BOX for any LOGICAL value, or any calculated value whose expression evaluates to a LOGICAL value.

How can we use "any calculated value whose expression evaluates to a LOGICAL value"? Try to create an example.

3. Make sure you ASSIGN the results of the edits back to the original lvalues and then use them in the following code. This proves that your testcases works (and will be a better test of P2J).

Also: please check all these into testcases/uast/toggle_box/.

#7 - 06/06/2014 12:37 PM - Eugenie Lyzenko

- File *toggle_laf_4gl.doc.zip* added

1. Check if ATTR-SPACE or other options can have an impact on the label positioning. All the labels in your examples are missing a space between the brackets and the label ([]label instead of [] label). This doesn't look right.

I've made more tests(even for dynamically created widget) and every time the CHUI and GUI shapes differs in space between [] and label. GUI case looks like having such spaces but CHUI not. The attached pictures are examples from 4GL document demonstrating the same behavior.

And ATTR-SPACE attribute is not applicable for TOGGLE-BOX widget. Will try another ones.

#8 - 06/06/2014 01:28 PM - Greg Shah

ATTR-SPACE can be put into the frame phrase:

```
DISPLAY my-toggle-var WITH FRAME my-frame ATTR-SPACE.
```

#9 - 06/06/2014 07:39 PM - Eugenie Lyzenko

- File *toggle_screens_20140606.zip* added

- File *evl_testcases_20140606a.zip* added

The attached screens are to illustrate point [#1](#) and [#2](#) above. The ATTR-SPACE does not affect the TOGGLE-BOX looking. Again, Windows GUI has some distance between [] and label text itself. CHUI does not make the extra space. I think in theory it is possible to simulate [] label layout in CHUI by defining TOGGLE-BOX without any label(or empty label) and drawing the label text by other statement(PUT SCREEN for example). Or we can do even more simple - by defining the label for TOGGLE-BOX with embedded extra space in label text front: " label text".

For point [#2](#) above. I have performed some investigation and the result is:

1. On the one hand it is not possible to use the form like:

```
display 3 > 2 view as toggle-box...  
or  
display var1 <= var2 view as toggle-box...
```

2. On the other hand in my understanding the words "...calculated value whose expression evaluates..." in 4GL doc here means to have boolean variable anyway that accept the result of the expression evaluating:

```
...  
boolRes = intVar2 >= intVar1.  
display boolRes view-as toggle-box ...  
...
```

This code works(testcase *toggle_box7.p*). The difference between this case and usual toggle-box widget is here we omit toggle-box definition the define variable statement. I guess we can consider a toggle-box here as another read-only representation of the logical variable.

The testcases are also attached here. And will be checked soon into *uast/toggle_box*.

#10 - 06/06/2014 07:53 PM - Eugenie Lyzenko

Testcases have been committed in bzt as 1146.

#11 - 06/09/2014 03:04 PM - Greg Shah

CHUI does not make the extra space. I think in theory it is possible to simulate [] label layout in CHUI by defining TOGGLE-BOX without any label(or empty label) and drawing the label text by other statement(PUT SCREEN for example). Or we can do even more simple - by defining the label for TOGGLE-BOX with embedded extra space in label text front: " label text".

Don't worry about the space between the brackets and the label in ChUI mode. If the 4GL doesn't have an option that does it, then we won't either. The testcase screen shots just looked awkward, so I wondered if we were missing something. I guess the Progress 4GL designers just weren't very careful in their approach.

2. On the other hand in my understanding the words "...calculated value whose expression evaluates..." in 4GL doc here means to have boolean variable anyway that accept the result of the expression evaluating:

```
...  
boolRes = intVar2 >= intVar1.  
display boolRes view-as toggle-box ...  
...
```

We don't have to do anything to support this.

When you convert your testcases, what is the result? We had already put in some support as far as I know, so what is left to do on the conversion part?

#12 - 06/09/2014 04:30 PM - Eugenie Lyzenko

- File *evl_upd20140609a.zip* added

- File *evl_testcases_20140609a.zip* added

Yes, partially the conversion has already been implemented. Except handling of the R/W attribute CHECKED. I have added the support for the attribute in the update attached. For now I've not found anymore TOGGLE-BOX specific attribute to be added. But looks like we need to enumerate all TOGGLE-BOX attributes to find out the conversion support level.

The other issue is for dynamic widget creation support(toggle_box4.p testcase). It is converting but has the compilation error:

```
...
[javac] /home/evl/timco_new/p2j/src/com/goldencode/testcases/toggle_box/ToggleBox4.java:31: error: no suitable method found for createToggleBox(handle,String,Class<ToggleBox4.TriggerBlock0>,ToggleBox4)
[javac]         DynamicWidgetFactory.createToggleBox(toggle1, "VALUE-CHANGED", TriggerBlock0.class, ToggleBox4.this);
[javac]                                     ^
[javac]         method DynamicWidgetFactory.createToggleBox(handle,character) is not applicable
[javac]             (actual and formal argument lists differ in length)
[javac]         method DynamicWidgetFactory.createToggleBox(handle,String) is not applicable
[javac]             (actual and formal argument lists differ in length)
[javac]         method DynamicWidgetFactory.createToggleBox(handle) is not applicable
[javac]             (actual and formal argument lists differ in length)
[javac] /home/evl/timco_new/p2j/src/com/goldencode/testcases/toggle_box/ToggleBox4.java:53: error: cannot find symbol
[javac]         toggle1.unwrapWidget().isChecked()
[javac]                                 ^
[javac]         symbol:   method isChecked()
[javac]         location: interface CommonWidget
[javac] 2 errors
...
```

There are 2 issues here:

1. To define isChecked() to the CommonWidget interface
2. To find out if the DynamicWidgetFactory.createToggleBox(toggle1, "VALUE-CHANGED", TriggerBlock0.class, ToggleBox4.this); is the correct creation procedure and why it is not compiling if it is correct.

Two new tests have been added to the test set: toggle_box0.p just to verify the simplest case and toggle_box8.p to test the CHECKED attribute set/get.

#13 - 06/09/2014 04:40 PM - Greg Shah

1. To define isChecked() to the CommonWidget interface

Yes, you certainly need to add stubbed support for the CHECKED attribute.

2. To find out if the DynamicWidgetFactory.createToggleBox(toggle1, "VALUE-CHANGED", TriggerBlock0.class, ToggleBox4.this); is the correct creation procedure and why it is not compiling if it is correct.

No this is not correct. The problem is due to the fact that we do not support the TRIGGERS phrase. As far as I know, we don't need to support that at this time. Please just use a standard ON statement to setup the trigger separately from the CREATE widget statement.

#14 - 06/09/2014 04:41 PM - Greg Shah

Code Review 0609a

Please test how unknown value works for the CHECKED attribute and implement it in ToggleBoxWidget.setChecked(logical checked). Otherwise the changes look good. Please get this done quickly so we can move on.

#15 - 06/09/2014 05:54 PM - Eugenie Lyzenko

Please test how unknown value works for the CHECKED attribute and implement it in ToggleBoxWidget.setChecked(logical checked). Otherwise the changes look good. Please get this done quickly so we can move on.

The runtime error has been generating when trying to set CHECKED attribute to unknown:

```
...  
**Unable to assign UNKNOWN value to attribute CHECKED on TOGGLE-BOX @TOGGLE-BOX variable name@. (4083)  
...
```

The application continues execution.

#16 - 06/09/2014 07:02 PM - Eugenie Lyzenko

- File *evl_upd20140609b.zip* added

Added stubs for compiling the converted code. The message for UNKNOWN set value is not added yet.

The question:

In getter method we need to choose which one we leave, either returning boolean or logical value because we can not have both simultaneously.

#17 - 06/10/2014 10:30 AM - Greg Shah

Code Review 0609b

1. The `GenericWidget.isChecked()/setChecked()` methods should simply execute: `throw new RuntimeException("CHECKED attribute is undefined");`.
2. Please do implement the error processing for unknown value. The Progress name is kept during conversion. You should be able to read the widget name from the `GenericFrame.getOriginalName()` method.
3. In regard to your question:

In getter method we need to choose which one we leave, either returning boolean or logical value because we can not have both simultaneously.

All attribute getters must return wrappers (e.g. logical) instead of Java primitives. Otherwise, how would we handle things like when an attribute is set to unknown value?

#18 - 06/10/2014 11:17 AM - Eugenie Lyzenko

- File *evl_upd20140610a.zip* added

Implemented error message and clearing other questions.

Start planning to implement runtime processing.

#19 - 06/10/2014 11:20 AM - Greg Shah

Code Review 0610a

It looks good.

#20 - 06/11/2014 04:14 PM - Eugenie Lyzenko

- File *evl_upd20140611a.zip* added

This update adds base skeleton for ToggleBox runtime implementation. This is very early version just for review the way I'm going on.

The idea is to implement toggle-box with the same base as button but not inheriting from button. Because toggle-box has some specific data to differentiate from button and we will need to keep two configs in sync, one for ToggleBox and other for underlying Button in same class. Moreover conceptually the toggle-box is more like to the radio-button than button. So I decided to mix commons from radio-button and button into new implementation.

The issues to resolve:

1. Handle sync for screen value and variable value for toggle-box
2. Debug and verify(key/event processing for toggle-box)
3. Refine widget drawing code

Continue working.

#21 - 06/11/2014 04:53 PM - Greg Shah

Code Review 0611a

This is a very good start.

1. Please rename `ToggleBoxImpl.drawChekBox()` to `ToggleBoxImpl.drawCheckBox()`.
2. The `RadioButton` can only be used as a set of related buttons in a `ButtonGroup`. It has quite a bit of behavior that is unique to its "mutually exclusive" button behavior. I suspect it is not a good fit as a model for `ToggleBox`, since `ToggleBox` does not have the grouping features. On the other hand, I do agree that `Button` is probably very close. If your approach (to base `ToggleBox` on `Button`) does work out, please create a common base class for `Button/ToggleBox` and move all the common code into that class.

#22 - 06/11/2014 05:18 PM - Eugenie Lyzenko

1. Please rename `ToggleBoxImpl.drawChekBox()` to `ToggleBoxImpl.drawCheckBox()`.

OK. Sorry this was just syntax error.

2. The `RadioButton` can only be used as a set of related buttons in a `ButtonGroup`. It has quite a bit of behavior that is unique to its "mutually exclusive" button behavior. I suspect it is not a good fit as a model for `ToggleBox`, since `ToggleBox` does not have the grouping features. On the other hand, I do agree that `Button` is probably very close. If your approach (to base `ToggleBox` on `Button`) does work out, please create a common base class for `Button/ToggleBox` and move all the common code into that class.

OK. The main differences from `Button`(in addition to `CHECKED` attribute) on my guess are: the `ToggleBox` has internal value to be set/get and does not have `AutoGo` feature in key handling. And this means we need the `DataContainer` interface to be implemented for `ToggleBox`. And key processing is also different in `ToggleBox` comparing to `Button`.

#23 - 06/11/2014 10:24 PM - Eugenie Lyzenko

- File *evl_upd20140611b.zip* added

With this update correct handling for ASSIGN, screen value and variable value differences added. Also for now I have removed LabeledWidget interface and implementation due to no need to use it here.

Continue working with key handling and widget drawing.

#24 - 06/12/2014 09:05 AM - Greg Shah

Code Review 0611b

The changes look fine.

#25 - 06/13/2014 07:17 AM - Eugenie Lyzenko

- File *evl_upd20140613a.zip* added

- File *evl_testcases_20140613a.zip* added

The support for VALUE-CHANGED event has been added to the ToggleBox. The testcase to verify - *toggle_box10.p*. The idea is to substitute KeyInput event for ENTER and Spacebar with artificial SE_VALUE_CHANGED event. The server side widget code is also modified because the value for CHECKED attribute stored in ToggleBoxConfig instance become outdated when we change the CHECKED attribute on the client side (and until implicit or explicit assign execution). So it is required to retrieve the fresh value from screen buffer. This is how the 4GL works in this case.

Continue working.

#26 - 06/13/2014 10:38 AM - Greg Shah

Code Review 0613a

The changes look good. The only thing to change is to save off the screen-buffer changes into the config instance. I'm not sure it will make a difference, but it will make the state of that more consistent, which may save an error later.

#27 - 06/13/2014 02:17 PM - Eugenie Lyzenko

- File *evl_testcases_20140613b.zip* added

- File *evl_upd20140613b.zip* added

This new update fixes the note for previous one and has cleaned widget drawing code. Looks like no additional changes required to make draw correct. The testcase to verify - *toggle_box11.p*. But new 4GL feature has been discovered with layout setting. When trying to set toggle-box width to value < 3 produces the error message (test *toggle_box12.p*):

```
...
**Unable to realise TOGGLE-BOX check1. (4025)
**Unable to realise TOGGLE-BOX check11. (4025)
Procedure complete. Press space bar to continue.
...
```

The application stops execution from this point.

The statements that generate this error: either `def frame...` or `display....`. The interesting fact is the incorrect size setting itself in:

```
...
def var check1 as logical no-undo view-as toggle-box size 1 by 1.
...
```

does not produce any error.

The other interesting fact is if we have more than one toggle-box with too short width - the message will display for all such widgets, not for the first encountered as I could expect. I think the key word here is "realise" in the message which may mean trying to place it to the frame(may be view() call in term of P2J). Anyway I guess we need to find single place for this exception, probably on the server side will be better.

The other question is about dynamically created widget support(toggle_box4.p test).

1. Do we have this for other widgets?
2. If yes, what level of support for toggle-box we need here?

I'm asking because for example the order of option in create toggle-box ... assign ... is very important. If we put FRAME = FRAME f1:HANDLE after option that requires frame value to be set - the NullPointerException will be raised(there is no valid frame defined at this time in this case).

#28 - 06/13/2014 07:41 PM - Eugenie Lyzenko

- File *toggle13.jpg* added

Another screen artifact has been found working with toggle-box widget. Take a look at toggle_box13.jpg. The problematic area is marked in red. I think this is something like phantom cursor. It is blinking and disappears when moving focus to another widget. There is no such effect in 4GL.

#29 - 06/16/2014 02:02 PM - Greg Shah

Code Review 0613b

The changes look fine.

the message will display for all such widgets, not for the first encountered as I could expect. I think the key word here is "realise" in the message which may mean trying to place it to the frame(may be view() call in term of P2J).

Yes, "realize" means the first time that a given frame is made visible. There are some things that only happen at that time. For example, one can only set the PARENT attribute of a FRAME before it is realized.

If you can implement the realize error processing at the server, it is best. Please note that it probably does not raise an ERROR condition, otherwise you would not see multiple messages.

The other question is about dynamically created widget support(toggle_box4.p test).

1. Do we have this for other widgets?

Yes, we at least support conversion for dynamic widgets. I'm not sure how complete our runtime support is, but I suspect it is pretty close because our conversion approach uses many of the same methods that we already use for static widgets/frames.

2. If yes, what level of support for toggle-box we need here?

Please provide the same level of support as we have for other widgets.

I'm asking because for example the order of option in create toggle-box ... assign ... is very important. If we put FRAME = FRAME f1:HANDLE after option that requires frame value to be set - the NullPointerException will be raised(there is no valid frame defined at this time in this case).

What does the 4GL do here? I believe that this can't be done in the 4GL either. It probably raises an error. We should do the same thing, but we may wait on that since it could require lots of code to be implemented across all places that are sensitive. Please investigate the behavior and post it here. Then we will probably create a new task and defer the work.

The problematic area is marked in red. I think this is something like phantom cursor. It is blinking and disappears when moving focus to another widget. There is no such effect in 4GL.

Yes, please fix it.

#30 - 06/16/2014 03:54 PM - Eugenie Lyzenko

- File toggle4.jpg added

What does the 4GL do here? I believe that this can't be done in the 4GL either. It probably raises an error. We should do the same thing, but we may wait on that since it could require lots of code to be implemented across all places that are sensitive. Please investigate the behavior and post it here.

It depends on the option we are trying to set before frame is known. But in 4GL this can cause the error too. The one thing is stable - the option setting will not be done. For example if the option is SENSITIVE = TRUE - the error is displaying and the toggle-box is not become sensitive. If the option is VISIBLE = TRUE like in the sample:

```
...  
CREATE TOGGLE-BOX toggle1 ASSIGN  
  LABEL = "Check here means TRUE"  
  SENSITIVE = TRUE  
  VISIBLE = TRUE  
  FRAME = FRAME f1:HANDLE.  
...
```

the error for VISIBLE option is not displaying but toggle-box will not become visible. Like in screen toggle4.jpg.

#31 - 06/16/2014 04:45 PM - Greg Shah

I have added a note to [#1791](#) to describe the problem and to reference note 30 in this task. You don't need to do anything further on this ordering problem right now.

#32 - 06/17/2014 05:39 PM - Eugenie Lyzenko

The problematic area is marked in red. I think this is something like phantom cursor. It is blinking and disappears when moving focus to another widget. There is no such effect in 4GL.

Yes, please fix it.

Debugging shows the root cause. This is the `ThinClient.waitForWorker()`(as the part of the `update()` sequence) the line 9344:

```
...
    if (status)
    {
        startInputStatus(true);
        statusSet = true;
    }
...
```

Here we force turn the cursor on inside `startInputStatus()`. And it is useless to turn the cursor off inside `ToggleBox` code. The `ThinClient.waitForWorker()` will be called after the `ToggleBox` handles the focus gaining code.

I think the solution can be do not force the cursor ON if the widget to be focused is not expecting this. The `ToggleBox` is the case. Some widgets require the cursor(like `FillIn`) but some does not(like `ToggleBox` or `Button` I guess).

The fix for now can be the following modification of the `ThinClient.waitForWorker()` code:

```
...
    if (status)
    {
        Widget widOnFocus = UiUtils.getCurrentFocus();

        startInputStatus((widOnFocus == null) || !(widOnFocus instanceof ToggleBox));
        statusSet = true;
    }
...
```

This is the fix for `ToggleBox` that works. More general approach will require to introduce new method into `Widget/AbstractWidget` interface/implementation to request the widget for cursor requirement when focused, say `forceCursorOnFocus()`. Widgets that does not require the cursor on when focused will override this method returning false, otherwise the method will return true from `AbstractWidget`. So the code can look like:

```
...
    if (status)
    {
        Widget widOnFocus = UiUtils.getCurrentFocus();

        startInputStatus((widOnFocus == null) || widOnFocus.forceCursorOnFocus());
        statusSet = true;
    }
...
```

Note here we will force the cursor ON when there is no focused widgets. Not sure if this is the possible situation, I just make this for previous logic compatibility(when we always turn the cursor ON).

What do you think about these approaches?

Another point - the current `ToggleBox` handles correctly the widget messages: `VALUE-CHANGED`, `ENTRY` and `LEAVE`. This is all we need from

messages perspective, right?

#33 - 06/17/2014 09:44 PM - Eugenie Lyzenko

- File `evl_upd20140617a.zip` added

This update fixes UI artifact using new Widget interface method implementation. Also the `ToggleBox` class has been simplified to avoid double execution of the same code.

#34 - 06/18/2014 08:42 AM - Greg Shah

Code Review 0617a

1. In the javadoc for these changes, "cusror" should be "cursor".
2. The javadoc for `ToggleBoxImpl.forceCursorOnFocus()` should explain that the return value is always false.

What do you think about these approaches?

I really like the approach. The code is good.

the current `ToggleBox` handles correctly the widget messages: `VALUE-CHANGED`, `ENTRY` and `LEAVE`. This is all we need from messages perspective, right?

I think so.

Is there anything left to do in this task?

#35 - 06/18/2014 09:52 AM - Eugenie Lyzenko

2. The javadoc for `ToggleBoxImpl.forceCursorOnFocus()` should explain that the return value is always false.

OK.

Is there anything left to do in this task?

Two things:

1. Verify the dynamically creation widget code works on the same level as for other widgets.
2. Implement the error messages and application interrupt when trying to realize the toggle-box with width less than 3(mandatory minimum size for toggle-box).

#36 - 06/18/2014 09:56 AM - Eugenie Lyzenko

Also may be as one more thing to do is to refactor the Button and ToggleBox implementation to merge possible common code in separate class. But may be we do not need this because the differences we have can be more than commons.

#37 - 06/18/2014 02:04 PM - Greg Shah

Also may be as one mo thing to do is to refactor the Button and ToggleBox implementation to merge possible common code in separate class

Please do this refactoring now.

#38 - 06/18/2014 02:39 PM - Eugenie Lyzenko

Also may be as one more thing to do is to refactor the Button and ToggleBox implementation to merge possible common code in separate class

Please do this refactoring now.

OK.

And BTW for dynamic widget creation. I guess the level support is the same. I have compared Button and ToggleBox created by CREATE WIDGET The result is the same - converted and compiled but at runtime we have exception(same exception for both Button and ToggleBox):

```
...
java.lang.IllegalStateException: Missing type attribute for method mapping entry [method-mapping: null]
    at com.goldencode.p2j.util.SourceNameMapper.buildMethod(SourceNameMapper.java:1164)
    at com.goldencode.p2j.util.SourceNameMapper.buildExternalProgram(SourceNameMapper.java:1070)
    at com.goldencode.p2j.util.SourceNameMapper.initMappingData(SourceNameMapper.java:996)
    at com.goldencode.p2j.util.SourceNameMapper.convertJavaProg(SourceNameMapper.java:192)
    at com.goldencode.p2j.util.SourceNameMapper.convertJavaProgSafe(SourceNameMapper.java:208)
    at com.goldencode.p2j.util.BlockManager.checkJavaCall(BlockManager.java:6978)
    at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:6829)
    at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:218)
    at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:200)
    at com.goldencode.testcases.toggle_box.ToggleBox4.execute(ToggleBox4.java:18)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.Utils.invoke(Utils.java:1237)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1679)
    at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1179)
    at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:363)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
```

```

at java.lang.reflect.Method.invoke (Method.java:606)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:693)
at com.goldencode.p2j.net.Conversation.block (Conversation.java:319)
at com.goldencode.p2j.net.Conversation.run (Conversation.java:163)
at java.lang.Thread.run (Thread.java:745)
[06/18/2014 20:00:02 MSK] (StandardServer.invoke:SEVERE) {00000001:00000007:bogus} Abnormal end!
java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke (Method.java:606)
    at com.goldencode.p2j.util.Utills.invoke (Utills.java:1237)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute (StandardServer.java:1679)
    at com.goldencode.p2j.main.StandardServer.invoke (StandardServer.java:1179)
    at com.goldencode.p2j.main.StandardServer.standardEntry (StandardServer.java:363)
    at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke (Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block (Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run (Conversation.java:163)
    at java.lang.Thread.run (Thread.java:745)
Caused by: java.lang.RuntimeException: Unable to load name mapping data.
    at com.goldencode.p2j.util.SourceNameMapper.initMappingData (SourceNameMapper.java:1005)
    at com.goldencode.p2j.util.SourceNameMapper.convertJavaProg (SourceNameMapper.java:192)
    at com.goldencode.p2j.util.SourceNameMapper.convertJavaProgSafe (SourceNameMapper.java:208)
    at com.goldencode.p2j.util.BlockManager.checkJavaCall (BlockManager.java:6978)
    at com.goldencode.p2j.util.BlockManager.topLevelBlock (BlockManager.java:6829)
    at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:218)
    at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:200)
    at com.goldencode.testcases.toggle_box.ToggleBox4.execute (ToggleBox4.java:18)
    at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke (Method.java:606)
    at com.goldencode.p2j.util.Utills.invoke (Utills.java:1237)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute (StandardServer.java:1679)
    at com.goldencode.p2j.main.StandardServer.invoke (StandardServer.java:1179)
    at com.goldencode.p2j.main.StandardServer.standardEntry (StandardServer.java:363)
    at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke (Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block (Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run (Conversation.java:163)
    at java.lang.Thread.run (Thread.java:745)
...

```

So I think this is not ToggleBox specific and we can postpone this subtask if you do not mind. Let me know if any notes or objections.

#39 - 06/18/2014 02:45 PM - Greg Shah

So I think this is not ToggleBox specific and we can postpone this subtask if you do not mind. Let me know if any notes or objections.

OK, no objections.

#40 - 06/18/2014 09:09 PM - Eugenie Lyzenko

- File *evl_upd20140618a.zip* added

This update for review includes refactoring for Button and ToggleBox common code. The idea is to use new base class AbstractButton for both of them and for all other button-like widgets further. This should avoid data duplication and intersection. Testing the changes locally with toggle-box tests.

#41 - 06/18/2014 09:21 PM - Greg Shah

Code Review 0618a

The changes look good. The only thing needed is to properly sort the methods in AbstractButton and ToggleBox (public then protected then private...).

#42 - 06/19/2014 03:03 PM - Eugenie Lyzenko

- File *evl_upd20140619a.zip* added

The only thing needed is to properly sort the methods in AbstractButton and ToggleBox (public then protected then private...).

Fixed with this update.

Also in this update the error processing for incorrect widget realization has been included for you to review the approach. The idea is the closest place that reflects to realize is GenericFrame.copyToScreenBuffer() where widgets are prepared to put into frame before display/update/set of another view() related statements. I have used already existed method valid() for every GenericWidget. This way we can write widget specific UI validation for every widget separately by overriding the default method.

The worked code check every widget and put the message. If at least one widget is not valid ad the end of the GenericFrame.copyToScreenBuffer() the client session stops.

The remaining issue is in P2J we have additional message "Press space bar to continue." while in 4GL the only message before stop is "Procedure complete. Press space bar to continue.". Also I think in P2J the error messages are displaying twice. Working on solution.

#43 - 06/20/2014 05:20 PM - Greg Shah

Code Review 0619a

1. We can't reuse valid() because this is there for other reasons (to provide support for the HANDLE:IS-VALID built-in function). Please use a new method named preRealizeCheck().
2. Does the 4GL really exit out from this error in an unstoppable way (UnstoppableExitException)? Or does it raise a real STOP or QUIT when this occurs?

#44 - 06/20/2014 05:28 PM - Greg Shah

The remaining issue is in P2J we have additional message "Press space bar to continue." while in 4GL the only message before stop is "Procedure complete. Press space bar to continue."

This sounds like a QUIT (QuitConditionException). Check this before you make any other changes.

#45 - 06/20/2014 07:08 PM - Eugenie Lyzenko

This sounds like a QUIT (QuitConditionException). Check this before you make any other changes.

Yes, looks like you are right. This is the usual exit for CHUI client session in 4GL for application finish. I opened the testcase in 4GL CHUI client session and after error messages and "Procedure complete. Press space bar to continue." the testcase finishes as if it goes from start to finish without errors.

So I guess this is the QuitConditionException and generating this exception the P2J works the same as 4GL.

#46 - 06/20/2014 07:54 PM - Eugenie Lyzenko

- File *evl_upd20140620a.zip* added

The update for review uses preRealizeCheck() to verify widget realization ability and QuitConditionException to quit the client session for realize errors from the GenericFrame code. So I think there is no more left to do with toggle-box if the update is OK.

#47 - 06/21/2014 04:41 AM - Greg Shah

Yes, looks like you are right. This is the usual exit for CHUI client session in 4GL for application finish.

...

QuitConditionException to quit the client session for realize errors

Please show the testcase you are using for checking the error processing. Does the 4GL really cause a QUIT for this case? That is very unusual.

If you enclose the code in a block like this:

```
DEF VAR err-flag AS LOGICAL INIT true.  
DO ON ERROR UNDO, LEAVE:  
  /* code that may generate an error */  
  err-flag = false.  
END.  
  
if err-flag then  
  MESSAGE "There was an ERROR."
```

Then you can tell if the code generates an error or not. You can also try ON END-KEY, ON STOP and ON QUIT instead of ON ERROR. If there is an ERROR, you can also use NO-ERROR and then check the ERROR-STATUS handle to see if it is recorded properly.

Please check all of this so we can be sure to match the proper behavior.

Also: check in your final testcases.

#48 - 06/21/2014 04:44 AM - Greg Shah

Code Review 0620a

The code looks good, especially the refactoring of common code into buildWidgetSpecificMessage(). Once you finish your testing as described in note 47, we will know if this is the final code or not.

#49 - 06/22/2014 05:24 PM - Eugenie Lyzenko

- File evl_testcases_20140622a.zip added

Then you can tell if the code generates an error or not. You can also try ON END-KEY, ON STOP and ON QUIT instead of ON ERROR. If there is an ERROR, you can also use NO-ERROR and then check the ERROR-STATUS handle to see if it is recorded properly.

Please check all of this so we can be sure to match the proper behavior.

The tests shows it is the STOP as you can see from linux and windows screens attached. The testcase - toggle_box16.p. So I'm changing to the StopConditionException.

#50 - 06/22/2014 06:19 PM - Eugenie Lyzenko

- File evl_upd20140622a.zip added

New update for review uploaded. The StopConditionException has been raised to interrupt execution. The pause inserted before to reflect the behavior shown in pro -p toggle_box16.p in 4GL Linux. If we want to replace the direct exception with ErrorManager.postprocessOnAbend("") - we need to make this method in ErrorManager as public.

#51 - 06/23/2014 09:50 AM - Greg Shah

Code Review 0622a

I'm fine with the changes. Please get it regression tested (both conversion and runtime).

#52 - 06/23/2014 04:17 PM - Eugenie Lyzenko

The conversion part has been finished. The converted code is binary equal. Continue with runtime testing.

#53 - 06/23/2014 11:51 PM - Greg Shah

Please check in your testcases to testcases/uast/toggle_box/.

#54 - 06/24/2014 09:44 AM - Eugenie Lyzenko

- File evl_upd20140624a.zip added

The runtime regression has been found for Button related code for GSO 195 - help string is not updated correctly for Button in some conditions. The suggested fix is in this update for review in addition to the recent code base merge. The local toggle-box related tests are OK. The runtime regression testing will be restarted soon.

#55 - 06/24/2014 10:15 AM - Greg Shah

Code Review 0624a

The changes look good.

#56 - 06/24/2014 11:04 AM - Eugenie Lyzenko

Please check in your testcases to testcases/uast/toggle_box/.

Done. The revision is 1157.

#57 - 06/25/2014 08:23 AM - Eugenie Lyzenko

The main regression part completed without regression. Continue with CTRL-C part.

#58 - 06/25/2014 09:23 AM - Eugenie Lyzenko

The CTRL-C part has been completed without regressions. Hopefully 2-way and 3-way in a single session. The results have been uploaded to GCD share directory. I'm planning to commit the update in bzt withing 15 min. if there are no objections and distribute.

#59 - 06/25/2014 09:28 AM - Greg Shah

Go ahead with the check-in and distribution.

#60 - 06/25/2014 10:00 AM - Eugenie Lyzenko

The update 0624a has been committed in bzt as 10552.

#61 - 06/25/2014 10:20 AM - Greg Shah

- Status changed from New to Closed

- % Done changed from 0 to 100

#62 - 11/16/2016 12:12 PM - Greg Shah

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

Files

evl_testcases_20140605a.zip	185 KB	06/05/2014	Eugenie Lyzenko
toggle_laf_4gldoc.zip	25.8 KB	06/06/2014	Eugenie Lyzenko
toggle_screens_20140606.zip	340 KB	06/06/2014	Eugenie Lyzenko
evl_testcases_20140606a.zip	3.96 KB	06/06/2014	Eugenie Lyzenko
evl_upd20140609a.zip	64.1 KB	06/09/2014	Eugenie Lyzenko
evl_testcases_20140609a.zip	4.9 KB	06/09/2014	Eugenie Lyzenko
evl_upd20140609b.zip	80.8 KB	06/09/2014	Eugenie Lyzenko
evl_upd20140610a.zip	76.8 KB	06/10/2014	Eugenie Lyzenko
evl_upd20140611a.zip	91.6 KB	06/11/2014	Eugenie Lyzenko
evl_upd20140611b.zip	91.2 KB	06/12/2014	Eugenie Lyzenko
evl_upd20140613a.zip	91.4 KB	06/13/2014	Eugenie Lyzenko
evl_testcases_20140613a.zip	5.28 KB	06/13/2014	Eugenie Lyzenko
evl_upd20140613b.zip	91.2 KB	06/13/2014	Eugenie Lyzenko
evl_testcases_20140613b.zip	6.3 KB	06/13/2014	Eugenie Lyzenko
toggle13.jpg	23 KB	06/13/2014	Eugenie Lyzenko
toggle4.jpg	22.1 KB	06/16/2014	Eugenie Lyzenko
evl_upd20140617a.zip	211 KB	06/18/2014	Eugenie Lyzenko

evl_upd20140618a.zip	218 KB	06/19/2014	Eugenie Lyzenko
evl_upd20140619a.zip	281 KB	06/19/2014	Eugenie Lyzenko
evl_upd20140620a.zip	297 KB	06/20/2014	Eugenie Lyzenko
evl_testcases_20140622a.zip	86.8 KB	06/22/2014	Eugenie Lyzenko
evl_upd20140622a.zip	297 KB	06/22/2014	Eugenie Lyzenko
evl_upd20140624a.zip	297 KB	06/24/2014	Eugenie Lyzenko