

User Interface - Bug #3889

incorrect initial widget and screen values for combo boxes and selection lists

01/16/2019 01:42 PM - Sergey Ivanovskiy

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:	Adrian Lungu	% Done:	50%
Category:		Estimated time:	0.00 hour
Target version:			
billable:	No	case_num:	
vendor_id:	GCD	version:	
Description			
Related issues:			
Related to User Interface - Bug #3690: BUTTON widget reaction on SPACEBAR and...		Closed	
Related to User Interface - Bug #4415: combo-box item selection via screen-va...		New	

History

#1 - 01/16/2019 01:57 PM - Sergey Ivanovskiy

- File test-widgets-screen-values.p added

This program should print the following output:

```
test 1
cbWidget1=3
cbWidget2=
cbWidget3=
cbWidget4=0
?
?
?
?
test 2
?
?
?
cbWidget4=0
?
?
?
?
```

#3 - 06/11/2019 12:25 PM - Greg Shah

- Subject changed from *The initial widget and screen values for combo boxes and selection lists* to *incorrect initial widget and screen values for combo boxes and selection lists*
- Assignee set to Adrian Lungu
- Start date deleted (01/16/2019)

#4 - 07/01/2019 02:29 PM - Adrian Lungu

- % Done changed from 0 to 50
- Status changed from New to WIP

#5 - 07/23/2019 10:02 AM - Greg Shah

Code Review Task Branch 3690a Revision 11322

1. For ComboBoxWidget, GenericWidget and GenericFrame, we know that some changes are needed, but is not clear to me whether the changes are correct. We have been fighting this issue in various forms for a long time. I think the reason is that we have created layers of patches adding ever more complex conditional processing trying to fit various conditions as we find anomalies in 4GL application code. Instead, we need to write a more complete set of testcases than just the one posted in [#3889-1](#). I think the tests need to orthogonally handle these factors:

- widget lifecycle
 - uninitialized and unrealized
 - initialized and unrealized
 - uninitialized and realized
 - initialized and realized
- data type
 - test all BDT types that can be used in data holding/editing widgets
 - initialization based on a range of values that go with that type
 - default value
 - unknown value
 - representative range of values (e.g. for numbers it should test large negative, small negative, zero, small positive and large positive)
- widget type (all types that can represent data should be tested, including all 3 browse column types)
- format strings (this depends on the data type)
 - default
 - non-default (we only have to make something unambiguously different from the default so that we can detect in which cases the format is honored)
- ChUI vs GUI

I may be missing something here, especially in the lifecycle part. Do we need to consider visibility or something else? Still, I think the list above would give us a quite good model for implementing screen-value. Trying to fix this one bug at a time is a terrible approach. It is time to do this correctly.

2. For ComboBoxModel, the change seems reasonable but it directly contradicts the documented return of 0 for "no selection". To what degree do the callers of this code need to be adjusted. If they are all safe and this change is correct, then the javadoc should be updated to make that clear.

#6 - 07/23/2019 10:02 AM - Greg Shah

- Related to Bug #3690: *BUTTON* widget reaction on *SPACEBAR* and *ENTER* keys added

#7 - 07/25/2019 12:52 PM - Adrian Lungu

- File *combo_box_tests.tar.gz* added

- File *combo_box_require_char_data_type.png* added

I made some tests onto which I ran revision 11322 of branch 3690a. By now, I have included all combo-box sub-types: *DEFAULT*, *SIMPLE*, *DROP_DOWN* and *DROP_DOWN_LIST*. For each once, there are two data types tested: *INT* and *CHAR*. Each scenario includes 3 types of declaration for the combo-box: uninitialized, initialized with a value from *LIST-ITEMS* or initialized with a value not from *LIST-ITEMS*. Finally, there are several messages describing the behavior of the combo-box if assigns are made.

After all this, it was easy to notice some issues related to the value, screen-value or the painting of the value inside the combo-box:

1. After an *ASSIGN*, if the *SCREEN-VALUE* is "?", the *VALUE* should be as well "?". Now, the *VALUE* is set to the default: 0 for *INT* and "" for *CHAR*
2. In case of a *DROP-DOWN-LIST* initialized with a value not from *LIST-ITEMS*, the *SCREEN-VALUE* should be "?". Now, the *SCREEN-VALUE* is set to the invalid initial value. (however, the combo-box is painted with no value in it, which is correct)
3. Also, an uninitialized integer *DROP-DOWN-LIST* should have *SCREEN-VALUE* "?". Now, the *SCREEN-VALUE* is set to 0. (however, if we use *char* instead of *int*, the combo-box behaves as expected)
4. If we use a *DROP_DOWN* with an invalid initial value, the value should be painted in the combo-box. Now, the combo-box is empty. (however, the *SCREEN-VALUE* is set correctly)

I will continue my investigation with a static code analysis, starting with the commented snippets. Also, I will create tests for the other widgets and data types to get a comprehensive understanding of the bug.

As a remark, I have noticed that the *INT* values work for the *SIMPLE* and *DROP_DOWN* combo-boxes, despite the fact that in Progress, these two require only the *CHAR* data type. I have also attached some tests to reflect that. Should it be considered a bug at this stage?

#8 - 07/25/2019 03:30 PM - Greg Shah

I have noticed that the *INT* values work for the *SIMPLE* and *DROP_DOWN* combo-boxes, despite the fact that in Progress, these two require only the *CHAR* data type. I have also attached some tests to reflect that. Should it be considered a bug at this stage?

Probably not. Since it is invalid, we should not see existing 4GL code that uses this which means it should not be a compatibility issue. We can consider this an enhancement of FWD, unless you find some reason to exclude it.

- Related to Bug #4415: combo-box item selection via screen-value assignment has deviations added

Files

test-widgets-screen-values.p	2.25 KB	01/16/2019	Sergey Ivanovskiy
combo_box_tests.tar.gz	1.4 KB	07/25/2019	Adrian Lungu
combo_box_require_char_data_type.png	31.6 KB	07/25/2019	Adrian Lungu