

User Interface - Bug #8012

Fix combo-box corner cases

11/07/2023 06:49 AM - Radu Apetrii

Status: WIP	Start date:
Priority: Normal	Due date:
Assignee: Radu Apetrii	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	case_num:
billable: No	
vendor_id: GCD	
Description	

History

#1 - 11/07/2023 08:05 AM - Radu Apetrii

- Assignee set to Radu Apetrii
- Status changed from New to WIP

There are quite a lot of corner cases regarding combo-boxes. While working on another task, I managed to identify a few by creating some tests. The point of this task is to have a solution for all of them, even if there will be more to come.

The tests that I've created until this moment of time are on testcases/uast/combo_box/screen_value, added in rev. 2388. They mainly target the screen-value of the combo-boxes with/without list-item-pairs. Also, there is a file called ComboBoxAnalyzer.java which highlights the mismatches in results between Progress and FWD. One can use that file after they've made sure that the generated results (i.e. combo-box1-Progress.txt and combo-box1-FWD.txt) are in the same folder.

I've used branch 7806b to make changes and test these cases. I'm not done with the implementation, but I wanted to point out the branch that I've been using.

Feel free to add more people to the watching list and to post other cases that provide wrong results.

#4 - 11/21/2023 06:19 AM - Radu Apetrii

I added to testcases/uast/combo_box/screen_value some more tests regarding ADD-LAST, ADD-FIRST, LIST-ITEM-PAIRS, SCREEN-VALUE, INSERT, DELETE and combinations of them. The commit is on testcases rev. 2389.

Also, on 7806b rev. 14839 I added some more changes related to combo-boxes:

- Added a delete(long) function in ComboBoxWidget.
- Done some better handling on the append-type methods (such as INSERT, ADD-FIRST, ADD-LAST).
- Also, I rebased 7806b with trunk rev. 14835 since there were some changes of interest in one of the latest revisions.

As there are quite a lot of logic-based changes, I will create a separate post to explain each one of them.

#5 - 11/22/2023 05:25 AM - Radu Apetrii

After putting some thought into this, I came up with a format that, IMO, would make the changes relatively easy to read and examine. That is, present the problems in the following way: Hypothesis, Reason/Test that backs it up, Solution.

1. Calling SCREEN-VALUE before setting LIST-ITEM-PAIRS affects the next SCREEN-VALUE values. [ShowHide](#)

Test (extracted from testcases/uast/combo_box/screen_value/combo-box2.p):

```
DEFINE VARIABLE cbType1 AS CHARACTER
    VIEW-AS COMBO-BOX.
DEFINE VARIABLE cbType2 AS CHARACTER
    VIEW-AS COMBO-BOX.

DEFINE FRAME f-main cbType1 cbType2.
DO WITH FRAME f-main:
    cbType1:LIST-ITEM-PAIRS = "G, ".
    MESSAGE cbType1:SCREEN-VALUE. // ""

    MESSAGE cbType2:SCREEN-VALUE. // "?"
    cbType2:LIST-ITEM-PAIRS = "G, ".
    MESSAGE cbType2:SCREEN-VALUE. // "?"
END.
```

Solution:

- When setting the LIST-ITEM-PAIRS, check if we have used the SCREEN-VALUE before setting it again:

```
(listItemPairsConfig.usedScreenValue == null || listItemPairsConfig.usedScreenValue == false)
```

- listItemPairsConfig.usedScreenValue equal to null means that no getScreenValue was called up until this point.
 - listItemPairsConfig.usedScreenValue equal to false means that getScreenValue was **not** called before setting LIST-ITEM-PAIRS.
 - listItemPairsConfig.usedScreenValue equal to true means that getScreenValue was called before setting LIST-ITEM-PAIRS.
- Update the usedScreenValue field when calling getScreenValue.

2. When adding/inserting/deleting in/from LIST-ITEM-PAIRS, SCREEN-VALUE takes into account the first label used. [ShowHide](#)

Test (extracted from testcases/uast/combo_box/screen_value/combo-box12.p):

```
DEFINE VARIABLE cbType1 AS CHARACTER
    VIEW-AS COMBO-BOX.
DEFINE VARIABLE cbType2 AS CHARACTER
    VIEW-AS COMBO-BOX.

DEFINE FRAME f-main cbType1 cbType2.
DO WITH FRAME f-main:
    cbType1:LIST-ITEM-PAIRS = "A, ,B, ".
    cbType1:DELETE(""). // Items are "B, "
    MESSAGE cbType1:SCREEN-VALUE. // "?"

    cbType2:LIST-ITEM-PAIRS = "A, ,B, ".
    cbType2:LIST-ITEM-PAIRS = "B, ". // Items are "B, "
    MESSAGE cbType2:SCREEN-VALUE. // ""
END.
```

Solution:

- When setting the LIST-ITEM-PAIRS, check if we are in append mode (ADD, INSERT, DELETE) and the label that we are trying to set is the same as the first one ever set.

```
(!append || listItemPairsConfig.labelWhenFirstListItemPairs.equals(labelText)) // if this is true, the program will continue to evaluate if empty value is valid
```

3. By setting multiple LIST-ITEM-PAIRS, SCREEN-VALUE can remain empty string only if it is not accessed (through a getter) in between the time. [ShowHide](#)

Test:

```
DEFINE VARIABLE cbType1 AS CHARACTER
    VIEW-AS COMBO-BOX.
DEFINE VARIABLE cbType2 AS CHARACTER
    VIEW-AS COMBO-BOX.

DEFINE FRAME f-main cbType1 cbType2.
DO WITH FRAME f-main:
    cbType1:LIST-ITEM-PAIRS = "A, ".
    MESSAGE cbType1:SCREEN-VALUE. // ""
    cbType1:LIST-ITEM-PAIRS = "B, ".
    MESSAGE cbType1:SCREEN-VALUE. // "?"

    cbType2:LIST-ITEM-PAIRS = "A, ".
    cbType2:LIST-ITEM-PAIRS = "B, ".
    MESSAGE cbType2:SCREEN-VALUE. // ""
```

Solution:

- Even if LIST-ITEM-PAIRS is set multiple times, consider each one as the first time as long as there are no SCREEN-VALUE gets in the meantime.
- When going from one value to another, listItemPairsConfig.firstSetListItemPairs will become false (meaning that we have set LIST-ITEM-PAIRS more than once) only if we have got SCREEN-VALUE earlier. Otherwise, it will be considered true (with the meaning that it is the first LIST-ITEM-PAIRS):

```
else if (listItemPairsConfig.firstSetListItemPairs != null &&
        listItemPairsConfig.firstSetListItemPairs == true &&
        listItemPairsConfig.usedScreenValue != null)
    {
        listItemPairsConfig.firstSetListItemPairs = false;
    }
```

4. Using append-type methods (such as INSERT, ADD-FIRST, ADD-LAST) instead of directly setting LIST-ITEM-PAIRS will not influence SCREEN-VALUE. [ShowHide](#)

Test:

```
DEFINE VARIABLE cbType1 AS CHARACTER
    VIEW-AS COMBO-BOX.
DEFINE VARIABLE cbType2 AS CHARACTER
    VIEW-AS COMBO-BOX.

DEFINE FRAME f-main cbType1 cbType2.
DO WITH FRAME f-main:
    cbType1:LIST-ITEM-PAIRS = "B, ".
    MESSAGE cbType1:SCREEN-VALUE. // ""
    cbType1:LIST-ITEM-PAIRS = "A,,B, ".
    MESSAGE cbType1:SCREEN-VALUE. // "?"

    cbType1:LIST-ITEM-PAIRS = "B, ".
    MESSAGE cbType1:SCREEN-VALUE. // ""
    cbType1:HANDLE:ADD-FIRST("A", "").
    MESSAGE cbType1:SCREEN-VALUE. // ""
END.
```

Solution:

- If we previously worked with an empty value, and setting the current LIST-ITEM-PAIRS is through an append-type method, then empty value (for the SCREEN-VALUE) can still be considered valid and further analyzed.
- The correspondent in ComboBoxWidget.checkItems is the following part of the if statement:

```
append &&
listItemPairsConfig.firstSetListItemPairs == null &&
listItemPairsConfig.initialEmptyValue != null &&
listItemPairsConfig.initialEmptyValue == true
```

5. When setting SCREEN-VALUE, "" is not treated the same as " " (shocking, I know). [ShowHide](#)

Test (extracted from testcases/uast/combo_box/screen_value/combo-box3.p):

```
DEFINE VARIABLE cbType1 AS CHARACTER
VIEW-AS COMBO-BOX.
DEFINE VARIABLE cbType2 AS CHARACTER
VIEW-AS COMBO-BOX.

DEFINE FRAME f-main cbType1 cbType2.
ENABLE ALL WITH FRAME f-main.
DO WITH FRAME f-main:
  cbType1:LIST-ITEM-PAIRS = "A, ".
  cbType2:LIST-ITEM-PAIRS = "A, ".

  cbType1:SCREEN-VALUE = "".
  cbType2:SCREEN-VALUE = " ".

  MESSAGE cbType1:SCREEN-VALUE. // "?"
  MESSAGE cbType2:SCREEN-VALUE. // " "
END.
```

Solution:

- This was solved in #7806 by adding an unparsedValue parameter to ComboBoxWidget.validateScreenValue and comparing, for the second case (with " "), the unparsedValue and the parsedValue and determining if empty string is a valid case for the SCREEN-VALUE.

Most likely there are some cases that I've missed, but I will keep everyone up to date as soon as I find them. Also, 7806b reached rev. 14840 after I added the fix for point 3.

Currently, all 34 test files that I have created are working as in Progress. Also, navigating through a large customer application raised no errors. However, this needs extensive testing.