

User Interface - Feature #3908

implement browse column VIEW-AS COMBO-BOX

02/06/2019 12:26 PM - Greg Shah

Status: Closed	Start date:
Priority: Normal	Due date:
Assignee: Stanislav Lomany	% Done: 100%
Category:	Estimated time: 0.00 hour
Target version:	vendor_id: GCD
billable: No	
Description	
Related issues:	
Related to User Interface - Feature #3811: more misc UI features	Closed
Related to User Interface - Feature #4049: implement mouse wheel for combo-boxes	New
Related to User Interface - Bug #4075: Combo-box columns 4GL quirks	New 05/08/2019
Related to User Interface - Bug #4415: combo-box item selection via screen-va...	New

History

#1 - 02/06/2019 12:28 PM - Greg Shah

- Related to Feature #3811: more misc UI features added

#2 - 03/18/2019 04:12 PM - Greg Shah

- Assignee set to Stanislav Lomany

#4 - 04/08/2019 11:32 AM - Stanislav Lomany

What branch should I use for this task?

#5 - 04/08/2019 11:36 AM - Greg Shah

You can create a new 3908a or you can use 3811a if you think it will make merging easier.

#6 - 04/08/2019 02:37 PM - Stanislav Lomany

Created task branch 3908a from P2J trunk revision 11301.

#7 - 04/15/2019 07:59 AM - Stanislav Lomany

Rebased task branch 3908a from P2J trunk revision 11302.

#8 - 04/16/2019 08:49 AM - Greg Shah

Please summarize status here.

When will this be complete and stable?

#9 - 04/16/2019 10:24 AM - Stanislav Lomany

Please summarize status here.

I'm working on rendering/screen-value part. Because combo-box cells behavior differs from other widget types. Other issues seem to be less significant.

When will this be complete and stable?

The best estimate is the day after tomorrow, the worst - the end of this week.

#10 - 04/16/2019 04:27 PM - Stanislav Lomany

FYI. When a combo-box cell gains focus, it displays the value which corresponds the cell value. If there is no corresponding value in combo box-list, it displays the last used combo-box value for this column. So, from a user's perspective if a cell has no corresponding value in the combo-box, a random value will be displayed. I love Progress.

#11 - 04/16/2019 08:04 PM - Greg Shah

So, from a user's perspective if a cell has no corresponding value in the combo-box, a random value will be displayed.

I'm trying to imagine the insane code inside the 4GL that causes this. It makes no sense at all.

I love Progress.

Do you love horror movies too? :)

#12 - 04/17/2019 03:46 AM - Stanislav Lomany

I'm trying to imagine the insane code inside the 4GL that causes this. It makes no sense at all.

I guess that's the lack of code that sets "empty" combo-box value if there is no corresponding one. Nevertheless I assume we need to support this. One more thing: this random value is not saved when we leave the cell. In order to save it, a user has to explicitly re-select it from the drop-down list.

#13 - 04/17/2019 07:58 PM - Stanislav Lomany

Currently browse cells are rendered at the server side: value --srv--> screen-value. Combo columns add a new layer at the client side value --srv--> screen-value --client--> item description. This is NOT value --srv--> item description, it is supported by the following evidences:

1. Normally SCREEN-VALUE attribute contains cell text, but because of [#3908-10](#) it's not quite usable. However for combo columns it returns item values rather than descriptions.
2. Item description is chosen basing on the formatted screen value.
3. We can modify the list of items (add/remove items) and the column is supposed to be redrawn basing on the new value -> description mapping. So it is redrawn without re-fetching the rows.

I wrote that it is *supposed* to be redrawn because this functionality is buggy in 4GL:

1. When a new item is added, the cells matching the new item are redrawn, but only if the browse is scrolled or, if nearby cells are navigated, it is partially or fully redrawn.
2. When an existing item is removed, the cells matching the removed item are not redrawn until you select an another item for this cell. 4GL doesn't seem to invalidate some kind of cache.

I'm not going to support these two quirks.

#14 - 04/18/2019 09:01 AM - Greg Shah

I'm trying to understand your syntax in the last note.

I assume that value means the raw data in the backing lvalue (variable or database field). Correct?

I assume that screen-value is the character instance that is returned from the combo-box-cell-widget:screen-value attribute. Correct?

The item-description is clear, though this only occurs for the list-item-pairs mode. I assume that in the simple list mode, it works as expected (the value is always displayed)?

What do you mean by --srv--> and --client-->?

I'm not going to support these two quirks.

I'm OK with that. Please add tasks for these and make them related to [#3908](#) and [#2596](#).

#15 - 04/18/2019 09:29 AM - Stanislav Lomany

I assume that value means the raw data in the backing lvalue (variable or database field). Correct?

Yes.

I assume that screen-value is the character instance that is returned from the combo-box-cell-widget:screen-value attribute. Correct?

Yes, it normally matches text of any kind of cell.

The item-description is clear, though this only occurs for the list-item-pairs mode. I assume that in the simple list mode, it works as expected (the value is always displayed)?

Yes.

What do you mean by --srv--> and --client-->?

I mean the that rendering happens on the client or on the server side.

#16 - 04/18/2019 04:02 PM - Stanislav Lomany

Strange, but only character combo columns display the item description instead of the item value in a cell. However if you assign screen-value manually in a trigger (NOT row-display trigger), it works for all data types.

#17 - 04/22/2019 04:28 AM - Stanislav Lomany

If there is no corresponding value in combo box-list, it displays the last used combo-box value for this column. So, from a user's perspective if a cell has no corresponding value in the combo-box, a random value will be displayed.

Moreover, for non-character columns, when a combo cell gains focus, a random (last selected for this column) value is displayed as selected. So, basically, non-character combo columns are unusable.

#18 - 04/22/2019 07:37 AM - Greg Shah

Stanislav Lomany wrote:

If there is no corresponding value in combo box-list, it displays the last used combo-box value for this column. So, from a user's perspective if a cell has no corresponding value in the combo-box, a random value will be displayed.

Moreover, for non-character columns, when a combo cell gains focus, a random (last selected for this column) value is displayed as selected. So, basically, non-character combo columns are unusable.

I'm not sure this aspect needs to be duplicated. This seems like a bug in the 4GL, but we should create a task and make it related to [#3908](#) and [#2596](#).

Can we do something simple to allow non-character fields to be functional without breaking compatibility?

Another question: if you don't go into editing mode, does the combo-box work as expected?

#19 - 04/22/2019 08:31 AM - Stanislav Lomany

Can we do something simple to allow non-character fields to be functional without breaking compatibility?

Character columns are rendered as a cell and behave as combo-box as one would expect. 4GL rendering should be preserved for non-character columns because of compatibility. While we can afford to stick to correct behavior for all columns, I don't think it affects compatibility. The issue described in [#3908-10](#) is questionable. Personally, I would like to implement a normal behavior. Because supporting incorrect one requires efforts. Thoughts on this one?

Another question: if you don't go into editing mode, does the combo-box work as expected?

In non-editing mode it is a cell with text. There is a slight chance that it's not a bug, it's a feature, but character and non-character columns rendered in different ways. I'll leave it as is.

Overall I think that the behavior is buggy because combo columns are not widely used, and Progress Corp is fine with keeping this feature in an OKay state and not putting efforts into fixing it.

#20 - 04/22/2019 09:32 AM - Greg Shah

The issue described in [#3908-10](#) is questionable. Personally, I would like to implement a normal behavior. Because supporting incorrect one requires efforts.

I agree, this can be treated as a "quirk" (create a task, link it here and make it related to [#3908](#) and [#2596](#)).

#21 - 04/22/2019 01:12 PM - Stanislav Lomany

BTW scrolling with mouse wheel is not implemented for regular combo-boxes.

#22 - 04/22/2019 06:09 PM - Stanislav Lomany

Correction: 4GL behaves correctly for non-character columns regarding cell rendering and item selection when a cell gains focus **if** the value is declared in a character form rather than original data type. I.e.

```
"opt 1", "01/01/01"
```

not

```
"opt 1", 01/01/01
```

However, as I noted before, if we manually assign SCREEN-VALUE to a cell in a non-row-display trigger, then comparison can be performed properly which leads to correct rendering and selection. Combo-box value is converted into a string for comparison using the export format.

#23 - 04/22/2019 06:12 PM - Greg Shah

- Related to Feature #4049: implement mouse wheel for combo-boxes added

#24 - 04/23/2019 09:54 AM - Stanislav Lomany

It is interesting than for integer columns spaces are taken into consideration for comparison. So

```
tt.fl format ">>>9"  
  view-as combo-box LIST-ITEM-PAIRS  
    "opt 1", "1",  
    "opt 2", "2",  
    "opt 3", "3" DROP-DOWN-LIST
```

won't properly work. It should be

```
tt.fl format ">>>9"
  view-as combo-box LIST-ITEM-PAIRS
    "opt 1", " 1",
    "opt 2", " 2",
    "opt 3", " 3" DROP-DOWN-LIST
```

#25 - 04/23/2019 09:57 AM - Stanislav Lomany

It is interesting than for integer columns

For decimal columns too.

#26 - 04/23/2019 06:35 PM - Stanislav Lomany

Side issue: a regular combo in FWD with decimals should round decimals according to the format precision before comparison. E.g. value 2.222 won't select an item 2.22 in FWD. Testcase:

```
def var v as decimal
  view-as combo-box list-item-pairs "item 1", 1, "item 2", 2.22, "item 3", 3 size 15 by 5 drop-down-list.

v = 2.222.
form v with frame f1.

update v with frame f1.
```

#27 - 05/01/2019 05:37 PM - Stanislav Lomany

Rebased task branch 3908a from P2J trunk revision 11306.

#28 - 05/03/2019 04:49 PM - Stanislav Lomany

Please review 3908a rev 11312, especially the changes for standalone combos.

#29 - 05/06/2019 03:30 PM - Hynek Cihlar

Code review 3908a.

- `setViewAsComboBox(boolean)` - the boolean parameter is redundant and should be removed (false value is a no-op), also I didn't find code that

- would allow to change the column type to fill-in (hColumn:VIEW-AS = "FILL-IN").
- setModeDropDownList(boolean) - same as above
- setModeDropDown(boolean) - same as above
- BrowseColumnWidget.delete is not implemented, is this expected?
- BrowseColumnWidget.setHonorFormat - missing Override annotation, calling the method on a non-combo column will cause an unexpected exception, rather do if (editor instanceof ComboBoxWidget) instead of if (editor != null)
- comboBoxItemsCache - shouldn't the cache be invalidated when the combo items change?
- ComboBoxGuiImpl.setValue - the condition config.mode == Mode.DROP_DOWN || config.mode == Mode.SIMPLE is redundant and should be removed, the original condition entryField != null already covers both cases. The check for modes duplicates the connection between modes and entryField being set.
- ComboBoxGuiImpl.destroy - shouldn't entryField be destroyed even when browse == null?

#30 - 05/08/2019 05:27 AM - Stanislav Lomany

- Related to Bug #4075: Combo-box columns 4GL quirks added

#31 - 05/08/2019 01:43 PM - Stanislav Lomany

The issues are addressed in 3908a rev 11313

- setViewAsComboBox(boolean) - the boolean parameter is redundant and should be removed (false value is a no-op),

Fixed.

also I didn't find code that would allow to change the column type to fill-in (hColumn:VIEW-AS = "FILL-IN").

We can add that in the future.

- setModeDropDownList(boolean) - same as above

Fixed.

- setModeDropDown(boolean) - same as above

Fixed.

- BrowseColumnWidget.delete is not implemented, is this expected?

List modification functions do not work properly in 4GL, so we can fix this in the future.

- BrowseColumnWidget.setHonorFormat - missing Override annotation, calling the method on a non-combo column will cause an unexpected exception, rather do if (editor instanceof ComboBoxWidget) instead of if (editor != null)

Fixed.

- comboBoxItemsCache - shouldn't the cache be invalidated when the combo items change?

Yes, but because list modification functions do not work properly in 4GL, I left it for the future.

- `ComboBoxGuiImpl.setValue` - the condition `config.mode Mode.DROP_DOWN || config.mode Mode.SIMPLE` is redundant and should be removed, the original condition `entryField != null` already covers both cases. The check for modes duplicates the connection between modes and `entryField` being set.

I would prefer to leave it as is because `entryField != null` may be not initialized yet at the point of the `config.mode Mode.DROP_DOWN || config.mode Mode.SIMPLE` check.

- `ComboBoxGuiImpl.destroy` - shouldn't `entryField` be destroyed even when `browse == null`?

Fixed.

Greg, should I run regression testing?

#32 - 05/08/2019 01:45 PM - Hynek Cihlar

Stanislav Lomany wrote:

- `ComboBoxGuiImpl.setValue` - the condition `config.mode Mode.DROP_DOWN || config.mode Mode.SIMPLE` is redundant and should be removed, the original condition `entryField != null` already covers both cases. The check for modes duplicates the connection between modes and `entryField` being set.

I would prefer to leave it as is because `entryField != null` may be not initialized yet at the point of the `config.mode Mode.DROP_DOWN || config.mode Mode.SIMPLE` check.

OK, makes sense.

#33 - 05/08/2019 01:54 PM - Greg Shah

Greg, should I run regression testing?

Yes, go ahead.

#34 - 05/12/2019 01:00 PM - Stanislav Lomany

Rebased task branch 3908a from P2J trunk revision 11308.
Passed regression testing basing on trunk revision 11306. Should I commit?

#35 - 05/13/2019 08:49 AM - Greg Shah

It passed ChUI conversion regression testing?

Also: the gap marking changes is not there. Please add those to 4069a.

#36 - 05/13/2019 09:02 AM - Stanislav Lomany

It passed ChUI conversion regression testing?

Yes.

Also: the gap marking changes is not there. Please add those to 4069a.

OK, I'll mark runtime support as partial.

#37 - 05/13/2019 09:29 AM - Greg Shah

I'll mark runtime support as partial.

I thought only the quirks were missing. Please post a detailed list of anything missing that is not a quirk.

#38 - 05/13/2019 11:53 AM - Stanislav Lomany

Missing conversion:

[MAX-CHARS characters] [AUTO-COMPLETION [UNIQUE-MATCH]]

Missing attributes:

AUTO-COMPLETION, MAX-CHARS, NUM-ITEMS, SUBTYPE, UNIQUE-MATCH

These functions may not work as expected (quirk).

ADD-FIRST, ADD-LAST, INSERT, REPLACE

Is not implemented because of the quirkiness of the functions above.

DELETE

#39 - 05/13/2019 12:34 PM - Greg Shah

Please rebase from 11309.

Constantin: Do you have an objection to this being merged to trunk?

#40 - 05/13/2019 12:39 PM - Constantin Asofiei

Greg Shah wrote:

Constantin: Do you have an objection to this being merged to trunk?

I'm OK with being merged to trunk, it looks safe.

#41 - 05/13/2019 01:13 PM - Greg Shah

OK, go ahead and merge once you've rebased.

#42 - 05/13/2019 01:58 PM - Stanislav Lomany

Rebased task branch 3908a from P2J trunk revision 11309.

3908a has been merged into the trunk as bzd revision 11310.

#43 - 11/14/2019 12:54 PM - Greg Shah

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

#44 - 11/14/2019 12:57 PM - Greg Shah

Stanislav Lomany wrote:

Missing conversion:
[MAX-CHARS characters] [AUTO-COMPLETION [UNIQUE-MATCH]]

Missing attributes:
AUTO-COMPLETION, MAX-CHARS, NUM-ITEMS, SUBTYPE, UNIQUE-MATCH

These functions may not work as expected (quirk).
ADD-FIRST, ADD-LAST, INSERT, REPLACE

Is not implemented because of the quiriness of the functions above.
DELETE

Is the gap marking up to date for these?

Where something is marked partial, is there a comment that explains the missing part?

Are there separate tasks which document the quirks?

#45 - 11/14/2019 12:58 PM - Greg Shah

- % Done changed from 0 to 100
- Status changed from New to WIP

#46 - 11/15/2019 01:53 PM - Stanislav Lomany

Gaps were updated in 3809e rev 11383. Quirks are documented in [#4075](#).

#47 - 11/15/2019 02:06 PM - Greg Shah

- Status changed from WIP to Closed

#48 - 01/21/2020 09:13 AM - Greg Shah

3809e was merged to trunk as revision 11340.