

User Interface - Bug #2775

Bug # 2677 (New): fix drawing and functional differences between P2J GUI and 4GL GUI

alert-box key processing is disabled when no main window is visible

10/22/2015 05:52 PM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Hynek Cihlar	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	case_num:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to User Interface - Bug #2773: simpler_alert_box.p fails in web GUI c...		Closed	
Related to User Interface - Bug #2787: implement mnemonics/accelerator key su...		Closed	
Related to User Interface - Bug #2789: implement generic ALT+mnemonic acceler...		New	
Related to User Interface - Bug #2790: implement generic for button groups		New	

History

#1 - 10/22/2015 05:54 PM - Greg Shah

The problem is documented in [#2773-9](#)

Recreate it with simpler_alert_box.p. If there is a visible owner window, then the key processing events get routed in Swing. If that window is not visible, the keys are not routed to the child window, even though that child window is visible.

Could this have something to do with the alert-box window not having focus at the OS level?

#2 - 10/22/2015 06:01 PM - Hynek Cihlar

Greg Shah wrote:

The problem is documented in [#2773-9](#)

Recreate it with simpler_alert_box.p. If there is a visible owner window, then the key processing events get routed in Swing. If that window is not visible, the keys are not routed to the child window, even though that child window is visible.

Could this have something to do with the alert-box window not having focus at the OS level?

Yes, this was the reason. A fix is checked in 1811s revision 11025.

#3 - 10/23/2015 08:01 AM - Greg Shah

Please note that the key processing is only partial. In my testing of 1811s rev 11032, I see the following issues:

1. LEFT and UP cursor keys and SHIFT-TAB should change the focus between buttons in a right to left direction. None of these keys work.
2. RIGHT and DOWN cursor keys and TAB should change the focus between buttons in a left to right direction. Only TAB seems to partially work.
3. When focus is already on the rightmost button, a right direction move will cycle back around and leave focus on the leftmost button. Likewise moving left from the leftmost button will leave focus on the rightmost button. Cycling doesn't work at all.

Please make sure all of the above is working on both GUI and ChUI.

#4 - 10/23/2015 08:20 AM - Hynek Cihlar

Greg Shah wrote:

Please note that the key processing is only partial. In my testing of 1811s rev 11032, I see the following issues:

2. RIGHT and DOWN cursor keys and TAB should change the focus between buttons in a left to right direction. Only TAB seems to partially work.

The focus doesn't move, only the highlight does. You can confirm this by pressing enter, you will always receive result of the first button.

Please make sure all of the above is working on both GUI and ChUI.

Yes.

#5 - 10/23/2015 08:23 AM - Greg Shah

The focus doesn't move, only the highlight does. You can confirm this by pressing enter, you will always receive result of the first button.

Understood, thanks.

#6 - 10/23/2015 02:28 PM - Hynek Cihlar

Task branch 2677a revision 10951 resolves GUI ALERT-BOX keyboard input.

Now button focus moves from the last button to the first and vice versa; button focus can be changed with TAB, SHIFT-TAB and arrow keys; added "underscore shortcuts for Yes, No and Retry buttons (other buttons do not have shortcuts in native 4GL), fixed setting of default button depending on the UPDATE expression.

Known issues:

- ButtonGuilImpl doesn't implement visual aids for the underscore shortcuts.
- The shortcuts can be invoked only with the character key. The native ALERT-BOX also allows the Alt key combination. I think the support for Alt button shortcuts should be handled globally, but that would require more risky changes.
- The reference to AlertBox from the button widgets is not optimal.

#7 - 10/23/2015 02:34 PM - Constantin Asofiei

Hynek Cihlar wrote:

- The shortcuts can be invoked only with the character key. The native ALERT-BOX also allows the Alt key combination. I think the support for Alt button shortcuts should be handled globally, but that would require more risky changes.

Yes, ALT+key as shortcut can be applied to other widgets, too. So we need a more general approach for it.

#8 - 10/24/2015 10:45 AM - Greg Shah

- File *gui_btn_accel0.p_accelerator_keys_in_buttons_in_progress_20151024.png* added

ButtonGuilImpl doesn't implement visual aids for the underscore shortcuts.

I have tested this using testcases/uast/button/gui_btn_accel0.p. It turns out that Progress does generically support defining the Windows accelerator key for a button. The **undocumented** approach is the same as for menus:

```
def button b_something label "Somet&hing".
def button b_quit label "Q&uit" AUTO-ENDKEY.

def frame fr
  b_something
  b_quit
  with centered size 50 by 5 title "Buttons Can Have Accelerators".

ON 'choose':U OF b_something
DO:
  MESSAGE "Something!".
  RETURN.
END.
```

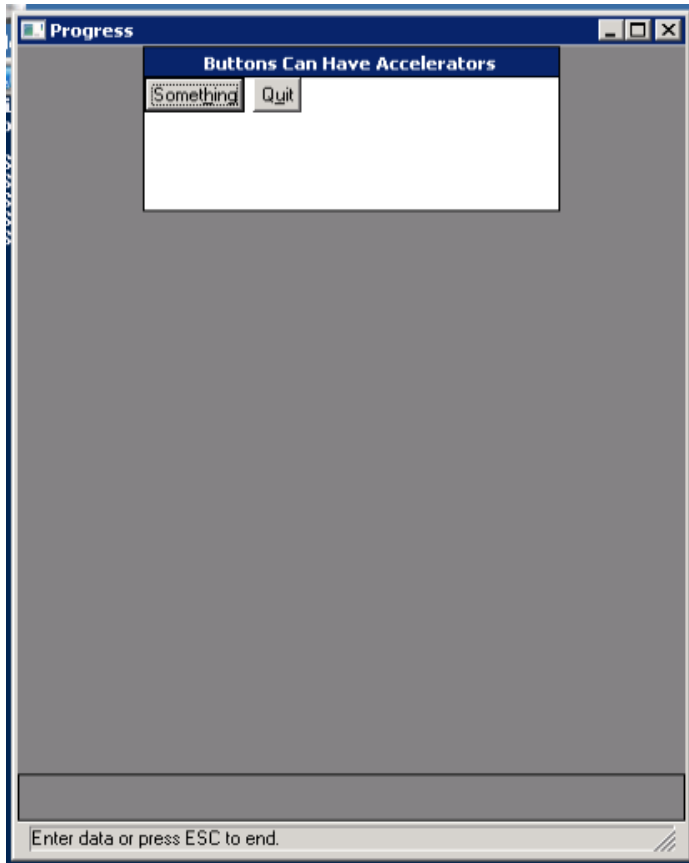
```
enable all with frame fr.
```

```
wait-for window-close of current-window or choose of b_quit in frame fr.
```

The & prefix marks the key to use as an accelerator. If the behavior is in shared code with menus, then it will likely also exhibit some other strangeness when we have multiple & characters and so forth.

I can also say there seems to be a bug in the 4GL. The accelerators only display sometimes on windev01 when running the same testcase multiple times. We are NOT planning to duplicate that bug.

Here is a screen capture from one of the times when the underlining is visible:



I've created a separate task for this in [#2787](#). It is bigger than just alert-box. We also need to check ChUI.

Interestingly, We do have some limited support in Button for this already. See `checkKey()`, `setUnderscoreOffset()` and `handleHotkeys()`. The only place it is used is in `AlertBoxImpl.doLayout()` which calls `button.setUnderscoreOffset(0)`; for each of the buttons.

Hynek: you could try to use this for GUI as a temporary solution.

Constantin: this brings up two questions:

- a) Do we consider the & character (and all its quiriness) when we capture `ui_strings.txt`?
- b) Do we place any of our "well known" strings which are embedded in the P2J source code (e.g. "Yes", "No", "Cancel"...) into `ui_strings.txt`?

However, we really need a more generic support. With that implemented, we could simply set the accelerator using the & in the button labels.

#9 - 10/24/2015 10:49 AM - Greg Shah

Constantin Asofiei wrote:

Hynek Cihlar wrote:

- The shortcuts can be invoked only with the character key. The native ALERT-BOX also allows the Alt key combination. I think the support for Alt button shortcuts should be handled globally, but that would require more risky changes.

Yes, ALT+key as shortcut can be applied to other widgets, too. So we need a more general approach for it.

Can you provide more information on how this works? I understand that in WIN32, one can define accelerator keys that are processed at the top level in the application's "window procedure" (low level WIN32 message processing loop). But I am unaware of the 4GL feature that exposes this as a general purpose feature for Progress apps.

#10 - 10/24/2015 10:50 AM - Greg Shah

The reference to AlertBox from the button widgets is not optimal.

What do you propose?

#11 - 10/24/2015 10:56 AM - Constantin Asofiei

Greg Shah wrote:

Constantin Asofiei wrote:

Hynek Cihlar wrote:

- The shortcuts can be invoked only with the character key. The native ALERT-BOX also allows the Alt key combination. I think the support for Alt button shortcuts should be handled globally, but that would require more risky changes.

Yes, ALT+key as shortcut can be applied to other widgets, too. So we need a more general approach for it.

Can you provide more information on how this works? I understand that in WIN32, one can define accelerator keys that are processed at the top level in the application's "window procedure" (low level WIN32 message processing loop). But I am unaware of the 4GL feature that exposes this as a general purpose feature for Progress apps.

From dvref.pdf Define Variable section:

In MS-Windows, you can designate a character within each label as a navigation mnemonic. Precede the character with an ampersand (&). When the variable is displayed with side labels, the mnemonic is underlined. The user can move focus to the variable by pressing ALT and the underlined letter. Navigation mnemonics operate only when you use side labels. If you specify more than one widget with the same mnemonic, the AVM transfers focus to each of these in tab order when you make a selection. Ending a label with an ampersand might produce unwanted behavior. To include a literal ampersand within a label, specify a double ampersand (&&).

This allows any widget with a side-label to have a mnemonic. I've tested with a fill-in and it works.

#12 - 10/24/2015 11:09 AM - Greg Shah

OK, I created [#2789](#) for the generic ALT-mnemonics support.

#13 - 10/24/2015 11:14 AM - Hynek Cihlar

Greg Shah wrote:

ButtonGuiImpl doesn't implement visual aids for the underscore shortcuts.

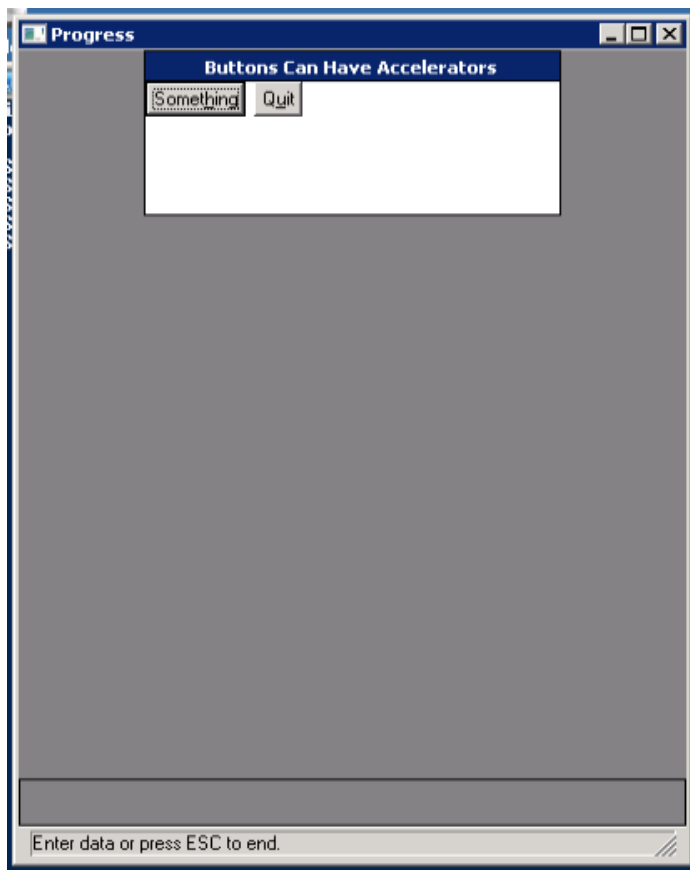
I have tested this using testcases/uast/button/gui_btn_accel0.p. It turns out that Progress does generically support defining the Windows accelerator key for a button. The **undocumented** approach is the same as for menus:

[...]

The & prefix marks the key to use as an accelerator. If the behavior is in shared code with menus, then it will likely also exhibit some other strangeness when we have multiple & characters and so forth.

I can also say there seems to be a bug in the 4GL. The accelerators only display sometimes on windev01 when running the same testcase multiple times. We are NOT planning to duplicate that bug.

Here is a screen capture from one of the times when the underlining is visible:



I've created a separate task for this in [#2787](#). It is bigger than just alert-box. We also need to check ChUI.

Interestingly, We do have some limited support in Button for this already. See `checkKey()`, `setUnderscoreOffset()` and `handleHotkeys()`. The only place it is used is in `AlertBoxImpl.doLayout()` which calls `button.setUnderscoreOffset(0)`; for each of the buttons.

Hynek: you could try to use this for GUI as a temporary solution.

I have done that. The GUI alert-box reacts on the shortcuts defined through `button.setUnderscoreOffset()` (with the exception of the Alt special key). Also as you mentioned the button already contains the shortcut logic, `ButtonGuiImpl` is just missing the drawing logic that would render the underscores.

#14 - 10/24/2015 11:18 AM - Greg Shah

ButtonGuiImpl is just missing the drawing logic that would render the underscores.

Good point. OK, the rest of the work really is just in [#2787](#).

#15 - 10/24/2015 11:24 AM - Hynek Cihlar

Greg Shah wrote:

The reference to AlertBox from the button widgets is not optimal.

What do you propose?

I propose to remove the reference to AlertBox from the button classes. The reason for its existence is a missing generic support for buttons grouped into logical units, like buttons in an alert box, so that tab-order cycling and lookup of a button based on a shortcut key event work as expected.

This generic support should be implemented.

#16 - 10/24/2015 11:47 AM - Greg Shah

- Status changed from New to Closed

#17 - 11/10/2015 05:10 AM - Greg Shah

- % Done changed from 0 to 100

#18 - 11/16/2016 12:12 PM - Greg Shah

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

Files

gui_btn_accel0.p_accelerator_keys_in_buttons_in_progress_20151024.92kB	10/24/2015	Greg Shah
--	------------	-----------