

User Interface - Bug #3233

button accelerators do not work

01/31/2017 09:59 AM - Greg Shah

Status: New	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	case_num:
billable: No	
vendor_id: GCD	
Description	
Related issues:	
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 - Feature #1790: implement menu support	Closed

History

#1 - 01/31/2017 09:59 AM - Greg Shah

- Related to Bug #2787: implement mnemonics/accelerator key support for button added

#2 - 01/31/2017 09:59 AM - Greg Shah

- Related to Bug #2789: implement generic ALT+mnemonic accelerator support for arbitrary focus changes in widgets added

#3 - 01/31/2017 10:00 AM - Greg Shah

- Related to Feature #1790: implement menu support added

#4 - 01/31/2017 10:07 AM - Greg Shah

I've looked into the missing accelerator processing for buttons. It is badly broken in 2 ways. The current implementation seems like it could never work except for the currently focused widget. Sadly, even this mode is not working. Worse, there seem to be no provisions for firing the accelerator on buttons that are not currently focused. This part is going to be trickier to solve. I will probably shift to something more urgent and we will try to talk around this in the demo.

See `./button/button_mnemonic_static_and_dynamic.p` as a simple testcase that shows the problem.

We need to write some testcases to check the following:

- Determine in what cases the accelerators can be triggered (what widgets can be in focus or not in focus).
- Confirm that non-enabled buttons cannot be triggered.
- Confirm that if a FILL-IN is focused it will eat the characters that would otherwise trigger an accelerator.

Most critically, I wonder at what level the accelerator processing occurs in the 4GL. Is it at the frame level? The window level? Both? The key here is to match the proper location of accelerator processing in the event processing loop.

Also, to make it more efficient we should do one of the following:

- update the event lists to add artificial trigger definitions for the accelerator keys (I think this is probably the wrong way to go because matching the definition exactly might be difficult or impossible)
- maintain a special list of currently active accelerators that is checked at the event processing point found above

#5 - 01/31/2017 10:09 AM - Greg Shah

The original problem can be seen in a customer app (see #3228-83).

#6 - 01/31/2017 07:52 PM - Greg Shah

From Constantin:

1. the 3111i was rebased and it fixes the radio-set mnemonic (but there is still an issue with its size). The button mnemonic in main window is not fixed (there were some notes from Hynek about a choose event). Also, do you know how mnemonics should be called? Simply pressing A or ALT-A? Or the buttons have triggers, which should be invoked when pressing i.e. A?

Please see testcases/uast/button/button_mnemonic_static_and_dynamic.p (make sure you use the one from rev 1604). I have tested this on windev01 and the results:

1. When the focus is on any of the widgets EXCEPT the FILL-IN, any press of a character associated with a mnemonic of an ENABLED button will cause the CHOOSE trigger to fire.
2. It doesn't matter if the key typed is upper or lower case, nor does it matter what the case is of the character following the ampersand in the label. The matching is case-insensitive.
3. A disabled button does not respond.
4. When focus is in the FILL-IN, the accelerators do not fire.
5. No use of the ALT key is needed.
6. I think this is equivalent to pressing ENTER or SPACE when a widget is focused.

I'm not sure if it is the window or the frame handling this. But otherwise the implementation is pretty straightforward.