

## User Interface - Feature #2423

### add proper key-to-event mappings in GUI

10/21/2014 01:17 PM - Constantin Asofiei

<b>Status:</b>	Closed	<b>Start date:</b>	06/08/2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Constantin Asofiei	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	GUI Support for a Complex ADM2 App	<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			
<b>Subtasks:</b>			
Bug # 2582: Some key labels are not supported by p2j			<b>New</b>
<b>Related issues:</b>			
Related to User Interface - Bug #2432: investigate and fix collisions related...			<b>New</b>

#### History

##### #1 - 10/21/2014 01:18 PM - Constantin Asofiei

From [#2252](#) note 391:

Greg Shah wrote:

I don't understand how is GUI mode special, so that it needs to treat the keyboard differently.

It isn't special. My only concern was to ensure that the GUI driver did in fact provide key events as required for widgets and READKEY.

OK, I think you are correct to be concerned, to some extent. Keys are read properly, but not all keys are interpreted the same way in GUI i.e.:

- in P2J, ALT-F4 doesn't show as the proper key code on my windows machine
- in 4GL, ESC is a replacement for what F4/END-ERROR does on ChUI
- in 4GL, F4 looks like it triggers CLOSE event
- in 4GL, F1 is linked to HELP and F2 is linked to GO event.

I think we should treat this in a separate task - we might need a way to make the Keyboard mappings configurable depending on the GUI/ChUI mode.

##### #2 - 10/22/2014 07:24 AM - Constantin Asofiei

- Status changed from New to WIP
- Assignee set to Constantin Asofiei

### #3 - 10/22/2014 11:36 AM - Constantin Asofiei

Greg, for this task, beside adding the key functions in Keyboard\$WorkArea, the SwingKeyboardReader needs to be split into ChUI and GUI implementations. Can you tell me what did you use as source for the original SwingKeyboardReader? Did you use the Programming Interfaces -> Handling User Input chapter or some other documentation?

### #4 - 10/22/2014 02:00 PM - Greg Shah

If I recall correctly, I started with a deep review of the 4GL docs and our current ChUI/NCURSES source code. I think I also used keys.p, keytest.p and key\_code\_report.p to review behavior of our current P2J implementation. At the time I did not have access to a 4GL dev system, so that is why some of the TODOs exist in the class. My objective was to get the swing ChUI client working like the NCURSES client for an interactive user.

### #5 - 10/23/2014 01:30 PM - Constantin Asofiei

- File chui-linux.txt added
- File gui.txt added
- File chui-win.txt added

After running key\_code\_report.p (for keys 0 to 2048), the key labels/functions for ChUI in linux and windows are different - this explains the collisions I saw for the MIDDLE-MOUSE-\* and MOUSE-MOVE-\* event codes.

I think we need the following:

- create Keyboard instances for each case - ChUI win/linux and GUI
- at runtime, P2J will determine which keyboard to use depending on the client type (ChUI or GUI) and the OS compatibility set in the directory (windows/linux).

Another issue is that I think on Windows, ALT has the same usage as ESC under linux - all key labels with a ESC- have ALT- instead.

So, for the current GUI project, I will focus on having correct key mappings for GUI usage, but I will search if I can find a smart way of "plugging in" this into the driver.

### #6 - 10/23/2014 04:15 PM - Greg Shah

Interesting findings. This is just one more place where Progress didn't spend the effort to make the platforms appear (the the 4GL programmer) to be more consistent. I'm OK with your approach.

### #7 - 10/29/2014 02:57 PM - Constantin Asofiei

Another frustrating fact about events: although the WINDOW-\* events are reporting LAST-EVENT:CODE values (other than -1), these values are not reported by the KEY-CODE("WINDOW-\*") function: this always returns -1... Progress either uses different tables for PROGRESS and KEYPRESS events or something else is happening. The collisions are (for GUI WINDOW-\* events):

1. WINDOW-CLOSE: 1078 - ALT-6
2. WINDOW-MAXIMIZED: 1094 - ALT-F
3. WINDOW-MINIMIZED: 1095 - ALT-G
4. WINDOW-RESTORED: 1096 - ALT-H
5. WINDOW-RESIZED: 1120 - ALT-`

**#8 - 10/29/2014 02:57 PM - Constantin Asofiei**

- File *ca\_upd20141029b.zip* added

This is a implementation of keyboards compatible with ChUI (linux/windows) and GUI (windows). When working on GUI mode, is important to set the `server/runtime/default/opsys/override` to `win32` (in `directory.xml`) and the `client/chui/windows=true` attribute in the client's bootstrap config.

Note that some refactoring was needed related to how the `SwingKeyboardReader` instance is used: because the client's compatibility mode (linux/windows) is required to be known BEFORE the client's session is established, all simulators require a `SwingKeyboardInstance` as parameter; it's the caller's responsibility to ensure that the same instance is shared by multiple simulators (i.e. in GUI mode, all windows will share the same instance). In ChUI mode, as only one window exists, is OK to pass a new instance directly.

Regression testing was not performed yet; will be done after the merge with the [#2229](#) / *ca\_upd20141029a.zip* update.

**#9 - 10/29/2014 02:58 PM - Constantin Asofiei**

PS: the issue in note 7 is not yet fixed.

**#10 - 10/29/2014 03:49 PM - Constantin Asofiei**

- File *ca\_upd20141029c.zip* added

This is replacement for 1029b.zip (merged with rev 10638) and is going into regression testing. Note 7 still applies.

**#11 - 10/30/2014 02:41 AM - Constantin Asofiei**

- File *ca\_upd20141030b.zip* added

Fixed a regression and javadoc issues.

**#12 - 10/30/2014 09:42 AM - Greg Shah**

Code Review *ca\_upd20141030b.zip*

I'm good with the changes. It is a decent approach. The unavoidable 4GL nastiness makes the solution pretty tricky.

What do you suggest in regard to note 7?

**#13 - 10/30/2014 09:49 AM - Constantin Asofiei**

Greg Shah wrote:

Code Review *ca\_upd20141030b.zip*

I'm good with the changes. It is a decent approach. The unavoidable 4GL nastiness makes the solution pretty tricky.

What do you suggest in regard to note 7?

I suggest to defer it... it requires more analysis on how the `PROGRESS/KEYPRESS` events are managed by 4GL. My theory is that they are using different event tables, but I don't know yet how we can prove this.

On a side note, runtime testing is almost passed, need one more run to clear false negatives. But I think I will release this after Marius's work, so that he doesn't have to merge again.

**#14 - 10/30/2014 09:53 AM - Greg Shah**

I suggest to defer it... it requires more analysis on how the PROGRESS/KEYPRESS events are managed by 4GL. My theory is that they are using different event tables, but I don't know yet how we can prove this.

OK. Please add a bug task referencing this task and note 7. Don't set a milestone, as I'm not sure this will impact us right now.

But I think I will release this after Marius's work, so that he doesn't have to merge again.

I appreciate your consideration here.

**#15 - 10/31/2014 05:25 AM - Constantin Asofiei**

- File *ca\_upd20141031a.zip* added

This one is merged with *mag\_upd20141030b.zip* from [#2252](#). Testing passed, release is pending.

**#16 - 11/05/2014 09:09 AM - Constantin Asofiei**

- File *ca\_upd20141105c.zip* added

This one fixes a regression related to keyboard (was not being re-initialized properly after a reset). Built on top of *ca\_upd20141105a.zip* from [#2252](#) (related to [#2416](#)).

One final regression testing (with *ca\_upd20141105a.zip*) with is running.

**#17 - 11/05/2014 02:57 PM - Constantin Asofiei**

*ca\_upd20141105c.zip* was committed to bzz rev 10646.

**#18 - 11/05/2014 03:06 PM - Greg Shah**

- Status changed from WIP to Closed

**#19 - 11/20/2014 06:27 AM - Constantin Asofiei**

In GUI, according to official manual, the CTRL-C equivalent is CTRL-BREAK. This has a key-code of 541, but pressing CTRL-BREAK in the windev01 emulator sends a CTRL-CTRL-S key (code 2067), which does raise a STOP condition. More, using the on-screen keyboard does not help - I've tried pressing CTRL-PAUSE, CTRL-Fn-PAUSE or other combinations (via the on-screen keyboard), but the CTRL-BREAK key was not emitted, nor STOP condition raised.

As a side note: pressing CTRL-C and restarting the client in GUI P2J is problematic. Currently, the windows are cleaned upon client restart in *TC.initializePrep*, but this should be done when the client is shutting down, after *standardEntry* API call is finished; if the window is still in a "responding" state when it needs to be cleared, events can be fired (like mouse or window activated) which can cause deadlocks. For know, I'll

assume that CTRL-BREAK will have the same behaviour as CTRL-C.

**#20 - 11/20/2014 07:35 AM - Constantin Asofiei**

More findings: on linux, the PAUSE/BREAK key emits the CTRL-S key (code 9). I think CTRL-BREAK (code 541) might be an event/function raised when pressing CTRL-CTRL-S... the weird part is that CTRL-S (code 9) is not raised when pressing PAUSE/BREAK, on the on-screen keyboard or directly; CTRL-PAUSE raises CTRL-CTRL-S...

We have two choices: either allow STOP condition to be raised by CTRL-CTRL-S (and maybe translate this into 541?) or assume that the documentation is wrong and in GUI CTRL-BREAK does not work... I'm inclined to believe the second part.

**#21 - 11/20/2014 08:45 AM - Greg Shah**

We have two choices: either allow STOP condition to be raised by CTRL-CTRL-S (and maybe translate this into 541?) or assume that the documentation is wrong and in GUI CTRL-BREAK does not work... I'm inclined to believe the second part.

Please send an email to Guy Mills. Ask him to check this for you, since he has "local" access to a Windows system with the 4GL running.

**#22 - 11/16/2016 12:13 PM - Greg Shah**

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

**Files**

---

gui.txt	53 KB	10/23/2014	Constantin Asofiei
chui-win.txt	59.4 KB	10/23/2014	Constantin Asofiei
chui-linux.txt	61.1 KB	10/23/2014	Constantin Asofiei
ca_upd20141029b.zip	203 KB	10/29/2014	Constantin Asofiei
ca_upd20141029c.zip	218 KB	10/29/2014	Constantin Asofiei
ca_upd20141030b.zip	220 KB	10/30/2014	Constantin Asofiei
ca_upd20141031a.zip	220 KB	10/31/2014	Constantin Asofiei
ca_upd20141105c.zip	221 KB	11/05/2014	Constantin Asofiei