FWD - Bug #3378

Date input fields don't get focus by clicking on them

11/21/2017 07:45 AM - Sergey Ivanovskiy

Status:	Closed	**Start date:**	11/21/2017
Priority:	Normal	**Due date:**	
Assignee:	Sergey Ivanovskiy	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

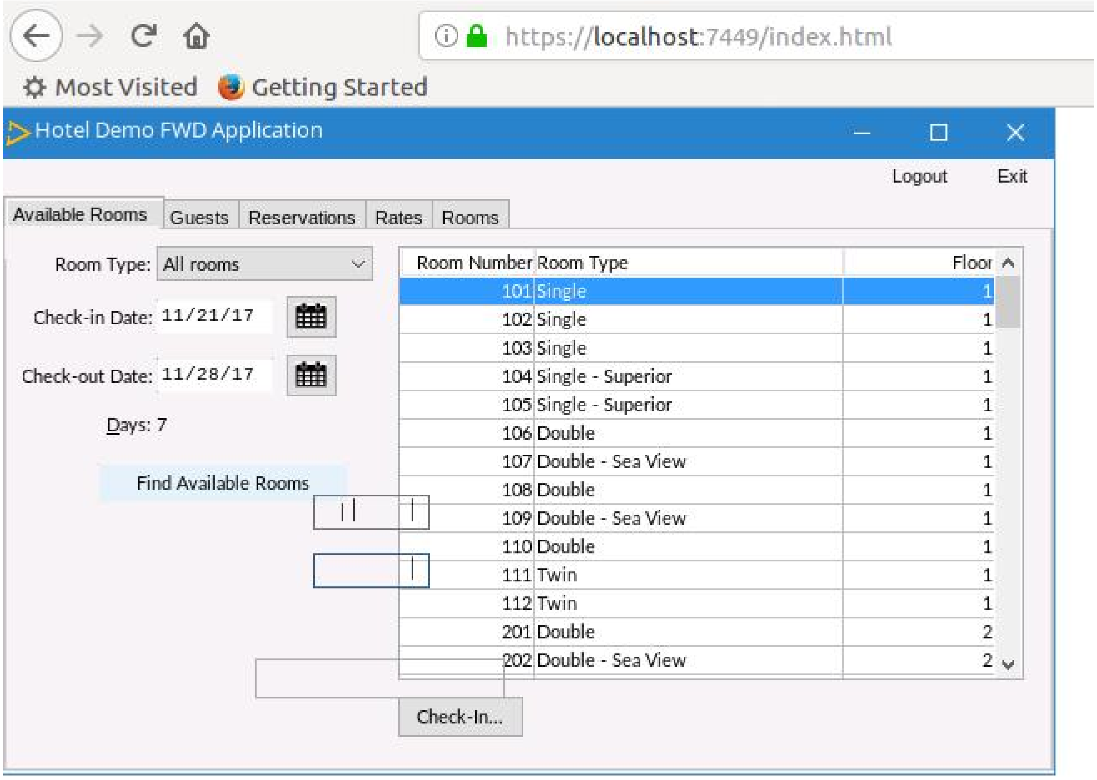
#1 - 11/21/2017 07:49 AM - Sergey Ivanovskiy

Open Hotel GUI with Swing client. The main screen is displayed with its first available tab "Available Rooms". Clicking on "Checkin Date" fillin doesn't move focus to this field.

#2 - 11/21/2017 07:56 AM - Sergey Ivanovskiy

- File BrokenScreenAfterRandomClicks.png added

On the web client after some random clicks involving the date input dialog I got this broken screen. The web client logs have no errors and the server's logs have no related errors except "UNSUPPORTED: READ-ONLY attribute is undefined ...".



Room Type: All rooms

Check-in Date: 11/21/17

Check-out Date: 11/28/17

Days: 7

Room Number	Room Type	Floor
101	Single	1
102	Single	1
103	Single	1
104	Single - Superior	1
105	Single - Superior	1
106	Double	1
107	Double - Sea View	1
108	Double	1
109	Double - Sea View	1
110	Double	1
111	Twin	1
112	Twin	1
201	Double	2
202	Double - Sea View	2

#3 - 11/22/2017 05:53 AM - Sergey Ivanovskiy

Debugged the Swing client for Hotel GUI. If the target "Checkin Date" fillin was clicked, then at the certain point of the message processing in TC.processProgressEvent(event) there were encountered the state in which the focus event was not delivered because of the old focus was not set. The program flow was in this point

```
           if (newFocus != oldFocus && newFocus.isFocusable() && newFocus.isEnabled())
            {
               // Send LEAVE to oldFocus.  If trigger returns NO-APPLY, then focus is switched
               // back to oldFocus. Note that 4GL first GIVES focus to newFocus (as the mouse
               // event is an OS event) and if the old focus doesn't accept LEAVE, it moves focus
               // back to the old one.  If old focus accepts LEAVE, ENTRY is sent to new focus.

               if (oldFocus != null && sendLeave(oldFocus, true))
               {
                  // if new and old focus are in different frames, send LEAVE/ENTRY to them
                  Widget newFrame = UiUtils.locateFrame(newFocus);
                  boolean sendFocusLost = false;
                  boolean sendFocusGained = false;
                  if (oldFrame != newFrame && oldFrame != null)
                  {
                     // prevent double-posting of LEAVE event if FRAME is the focused widget
                     if (oldFrame != oldFocus)
                     {
                        sendLeave(oldFrame, true);
                     }

                     sendFocusLost = true;
                  }

                  boolean newFocusEntry = true;
                  if (oldFrame != newFrame && newFrame != null)
                  {
                     newFocusEntry = sendEntry(newFrame, true);
                     sendFocusGained = true;
                  }

                  if (newFocusEntry)
                  {
                     // prevent double-posting of SEND event when FRAME is the target widget
                     if (newFrame != newFocus)
                     {
                        sendEntry(newFocus, true);
                     }

                     if (sendFocusLost && oldFrame != null)
                     {
                        EventManager.postEvent(new FocusEvent(oldFrame, EventType.FOCUS_LOST));
                     }
                     if (sendFocusGained)
                     {
                        EventManager.postEvent(new FocusEvent(newFrame, EventType.FOCUS_GAINED));
                     }
                     newFocus.requestFocus();
                  }
               }
               else if (oldFocus != null)
               {
                  // no ENTRY is generated when switching focus back to it
                  oldFocus.requestFocus();
               }
```

where oldFocus = null and newFocus is the "Checkin Date" widget. Could you help what is supposed to be for oldFocus at the moment when the main Hotel GUI window is displayed on the screen?

#4 - 11/22/2017 06:32 AM - Ovidiu Maxiniuc

Sergey Ivanovskiy wrote:

> On the web client after some random clicks involving the date input dialog I got this broken screen. The web client logs have no errors and the server's logs have no related errors except "UNSUPPORTED: READ-ONLY attribute is undefined ...".

The UNSUPPORTED: READ-ONLY attribute is being fixed in #3370. I played around with swing client and I couldn't reproduce rendering errors similar to your screen-shot. Apparently not only the date pickers are affected but the buttons as well: the grey rectangle is supposed to be around the Find Available Rooms. The strange thing is, the button fill and border are rendered in a single place, 3 lines apart, within the same clipping/translation.

#5 - 11/22/2017 01:03 PM - Sergey Ivanovskiy

I would like to propose this diff for the fix

```
=== modified file 'src/com/goldencode/p2j/ui/client/gui/FillInGuiImpl.java'
--- src/com/goldencode/p2j/ui/client/gui/FillInGuiImpl.java    2017-10-27 17:04:40 +0000
+++ src/com/goldencode/p2j/ui/client/gui/FillInGuiImpl.java    2017-11-22 18:00:07 +0000
@@ -546,6 +546,7 @@
          // focus management is done at the TC.processProgressEvent
          bypassFocusInit = true;
          invalidateSelection();
+         this.requestFocus();
       }

       String stxt = getText();
```

because all another GUI widgets request focus explicitly if a mouse click event is handled.

#6 - 11/22/2017 03:10 PM - Sergey Ivanovskiy

Committed revision 11203 (3355a) applied this fix.

#7 - 11/22/2017 03:53 PM - Greg Shah

3355a is going to be archived because its changes are already in 3369a, right?

This change should go into 3369a.

#8 - 11/22/2017 04:08 PM - Sergey Ivanovskiy

OK, merged this change into 3369a, revision 11221. Planning to archive 3355a.

#9 - 11/23/2017 06:53 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> I would like to propose this diff for the fix
> [...]
> because all another GUI widgets request focus explicitly if a mouse click event is handled.

I'm not sure why in this particular case the fillin doesn't receive focus, but it is already handled in TC.processProgressEvent (line 16942, search for if (event instanceof MouseEvt)). I think the idea is to transform the mouse event to an ENTRY event which (if not stopped by the app logic) is eventually transformed to a Widget.requestFocus call (in TC.processProgressEvent, search for if (evt.isSpecial() || triggerNesting > 0) { src.requestFocus()). On the way there are multiple conditions that must be satisfied in order for the widget to receive focus, you should use the debugger and see which condition is not met and proceed from there.

Thus requesting focus unconditionally from the fillin's mouse handler is not correct I believe.

#10 - 11/23/2017 07:41 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > I would like to propose this diff for the fix
> > [...]
> > because all another GUI widgets request focus explicitly if a mouse click event is handled.
>
> I'm not sure why in this particular case the fillin doesn't receive focus, but it is already handled in TC.processProgressEvent (line 16942, search for if (event instanceof MouseEvt)). I think the idea is to transform the mouse event to an ENTRY event which (if not stopped by the app logic) is eventually transformed to a Widget.requestFocus call (in TC.processProgressEvent, search for if (evt.isSpecial() || triggerNesting > 0) { src.requestFocus()). On the way there are multiple conditions that must be satisfied in order for the widget to receive focus, you should use the debugger and see which condition is not met and proceed from there.

According to #3378-3 this ENTRY event was not sent because oldFocus was null.

#11 - 11/23/2017 07:46 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> According to #3378-3 this ENTRY event was not sent because oldFocus was null.

Possibly the oldFocus being null is incorrect or the condition of checking oldFocus for null is incorrect. I think you will have to find out how Progress behaves in this case, what loses and what receives the focus, what messages are sent, etc., and based on that evidence do the code changes. If possible we want to fix this in TC, rather than in the widget itself.

#12 - 11/23/2017 08:03 AM - Sergey Ivanovskiy

OK, understand, planning to find another way. Sadly I couldn't setup Hotel GUI on VM Windows 12, but Hotel GUI is available on windev01.

#13 - 11/23/2017 08:32 AM - Constantin Asofiei

There is this code in TC.processProgressEvent:

```
               else if (oldFocus != null)
               {
                  // no ENTRY is generated when switching focus back to it
                  oldFocus.requestFocus();
               }
```

This is the alternative for oldFocus != null && sendLeave(oldFocus, true). I'm not sure that this is correct: we should focus back to oldFocus only if it's LEAVE was a return no-apply. But, if oldFocus is null, why don't we just send entry to newFocus?

I think the correct code should be:

```
if (oldFocus != null)
{
   if (sendLeave(oldFocus, true))
   {
      //... do the work 
   }
   else
   {
      // no ENTRY is generated when switching focus back to it
      oldFocus.requestFocus();
   }
}
else
{
    // send entry directly to newFocus
   sendEntry(newFocus, true);
}
```

#14 - 11/23/2017 08:46 AM - Sergey Ivanovskiy

It seems reasonable changes. Checked on windev01 that the native Hotel GUI places focus on the "Available Rooms" tab button when the main window appears first time but in this bug the current focus is null.

#15 - 11/23/2017 08:58 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Checked on windev01 that the native Hotel GUI places focus on the "Available Rooms" tab button when the main window appears first time but in this bug the current focus is null.

This is something else we should fix. I think the problem is in Frame.getFirstEnabledWidget - this doesn't treat the case of child frames. Please test this change:

```
### Eclipse Workspace Patch 1.0
#P p2j
Index: src/com/goldencode/p2j/ui/client/Frame.java
===================================================================
--- src/com/goldencode/p2j/ui/client/Frame.java    (revision 1332)
+++ src/com/goldencode/p2j/ui/client/Frame.java    (working copy)
@@ -7440,7 +7440,16 @@
       {
          comp = getField(widgets[i]);

-         if (UiUtils.hasConfig(comp) && comp.focusTraversable())
+         if (comp instanceof Frame)
+         {
+            Frame frame = (Frame) comp;
+            Widget<O> focus = frame.getFirstEnabledWidget();
+            if (focus != null)
+            {
+               return focus;
+            }
+         }
+         else if (UiUtils.hasConfig(comp) && comp.focusTraversable())
          {
             return comp;
          }
```

#16 - 11/23/2017 09:28 AM - Sergey Ivanovskiy

It seems very close to be a fix. The Available Rooms tab button is selected in this case, and FocusManager.setFocusOn(Tab_Button) doesn't set the focus on this widget and old focus from the login window is returned by UiUtils.getCurrentFocus()

```
   public void setFocusOn(final Widget widget)
   {
      Widget oldFocus = UiUtils.getCurrentFocus();
      boolean ins = Window.getInsertMode() == Window.INSERT_MODE_INS;
      tc.eventDrawingBracket(widget, false, false, new Runnable()
      {
         @Override
         public void run()
         {
            FocusManager.setInitialFocus(widget);
         }
      });

      Widget newFocus = UiUtils.getCurrentFocus();    <------------------------ returns the old focus, but widget should be here
................................
}
```

#17 - 11/23/2017 11:11 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> It seems very close to be a fix. The Available Rooms tab button is selected in this case, and FocusManager.setFocusOn(Tab_Button) doesn't set the focus on this widget and old focus from the login window is returned by UiUtils.getCurrentFocus()
> [...]

OK, this is because the old login window is still assumed as the 'current focus window'. We are missing something... a focus_gained should be sent to the window, too, so the WindowManager.focusWnd is set correctly. Please look into it, maybe via a simple test with 'entry' and triggers on a newly created window and on a i.e. button in this window.

#18 - 11/23/2017 11:28 AM - Hynek Cihlar

Constantin Asofiei wrote:

> OK, this is because the old login window is still assumed as the 'current focus window'. We are missing something... a focus_gained should be sent to the window, too, so the WindowManager.focusWnd is set correctly. Please look into it, maybe via a simple test with 'entry' and triggers on a newly created window and on a i.e. button in this window.

Btw., is this reproducible on both Swing and Web drivers?

#19 - 11/23/2017 11:30 AM - Sergey Ivanovskiy

I checked the Swing client, but the original issue was reproduced with the web client too.

#20 - 11/23/2017 11:32 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> I checked the Swing client, but the original issue was reproduced with the web client too.

Ok, this could also be wrong timing. The code that checks the current focus could run at the time the old window is still active (before WindowManager gets notified by the driver that the login window loses focus and/or the new one gains focus).

#21 - 11/23/2017 11:39 AM - Constantin Asofiei

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > I checked the Swing client, but the original issue was reproduced with the web client too.
>
> Ok, this could also be wrong timing. The code that checks the current focus could run at the time the old window is still active (before WindowManager gets notified by the driver that the login window loses focus and/or the new one gains focus).

I think you are correct, the WindowActivated event is processed later, in the waitForWorker's main event processing loop (the event is posted in the event queue and is picked up by the waitForEvent); the focus is processed before that, at the beginning of waitForWorker.

#22 - 11/23/2017 11:57 AM - Sergey Ivanovskiy

Constantin, actually your patch fixes the issue (the tab button is not displayed active, but it has the current focus). The old focus becomes a valid widget with these changes. I think it makes sense to apply these changes instead of the current changes (rev 11221(3369a)).

#23 - 11/23/2017 12:04 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Constantin, actually your patch fixes the issue (the tab button is not displayed active, but it has the current focus). The old focus becomes a valid widget with these changes. I think it makes sense to apply these changes instead of the current changes (rev 11221(3369a)).

Yes, please back out 3369a/11221 and add the changes in #3378-15

#24 - 11/23/2017 12:45 PM - Sergey Ivanovskiy

Please look at the committed rev. 11223 (3369a).

#25 - 11/27/2017 03:47 PM - Greg Shah

- Assignee set to Sergey Ivanovskiy

- Status changed from New to Closed

- % Done changed from 0 to 100

Branch 3369a was merged to trunk as revision 11206.

#26 - 04/10/2018 03:35 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> OK, merged this change into 3369a, revision 11221. Planning to archive 3355a.

Sergey, I think you forgot to archive 3355a.

#27 - 04/10/2018 03:47 PM - Sergey Ivanovskiy

Thank you, I archived 3355a now.

Files

BrokenScreenAfterRandomClicks.png	44.5 KB	11/21/2017	Sergey Ivanovskiy