# User Interface - Bug #3381

## Sub-menu drawing issues

11/22/2017 05:43 AM - Stanislav Lomany

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Eugenie Lyzenko | **% Done:** | 100% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |
| **billable:** | No | **case_num:** | |
| **vendor_id:** | GCD | | |

**Description**

**Related issues:**

| | | |
|---|---|---|
| Related to User Interface - Bug #3461: Remaining menu/sub-menu bugs or incorr... | **New** | **01/22/2018** |

### History

**#1 - 11/22/2017 08:12 AM - Stanislav Lomany**

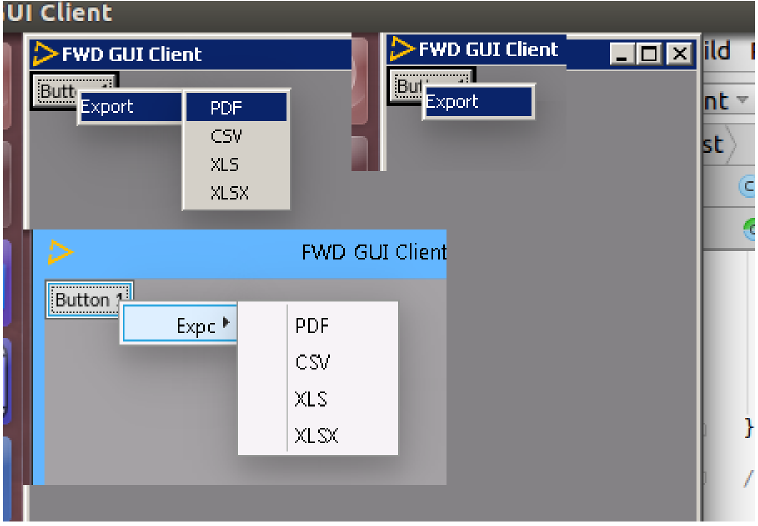
*- File sub-menus.p added*

*- File samples.png added*

*- File start.p added*

Testcase attached (sub-menus.p). The following scenarios have issues:

1. Right-click on the button: in swing the right arrow next to "Export" is missing, in win8 "Export" is cut to "Expo".
2. Click on "PDF" item: "Export" item doesn't disappear.



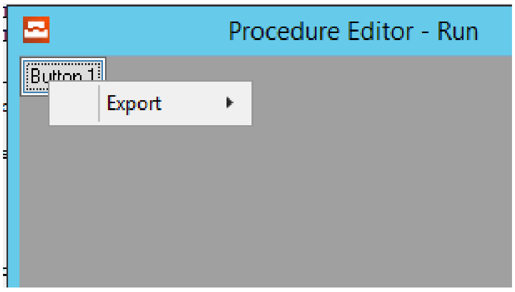
3. Hover over "Export". When you hover out of "Export", sub menu doesn't disappear while it should (4GL has delay after which sub-menu disappears).
4. Click on "Export". Sub menu disappears while it shouldn't. If you hover out of "Export", in swing the color of "Export" text changes from white to black.
5. Run start.p and press Esc. "Static handle type MENU-ITEM given to DELETE PROCEDURE or DELETE OBJECT statement. (5428)" error appears.

**#2 - 12/08/2017 10:01 PM - Eugenie Lyzenko**

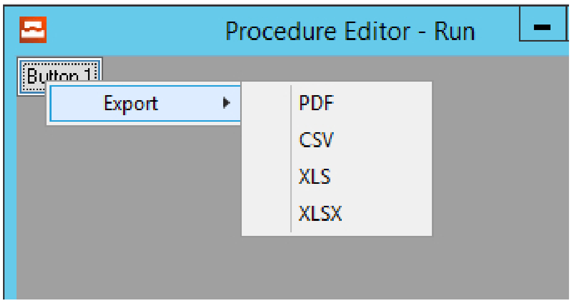
*- File sub_menu1_level1.jpg added*

*- File sub_menu1_level2.jpg added*

4GL level1 menu opening:



4GL level2 menu opening:

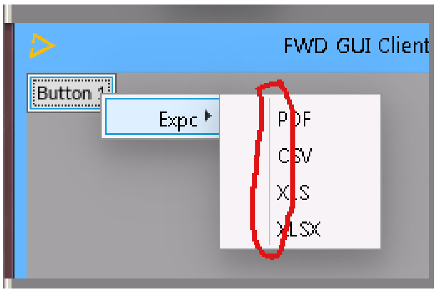


**#3 - 12/11/2017 12:07 PM - Eugenie Lyzenko**

*- File missing_vert_win10.jpg added*

Stanislav,

From your picture I can see vertical line in Windows10 theme:



No there is no such line in FWD. Can you remember when you saw it last time? I need to find out when it disappeared.

**#4 - 12/11/2017 02:17 PM - Stanislav Lomany**

> No there is no such line in FWD. Can you remember when you saw it last time?

I guess somewhere between revisions and 11208 and 11230.

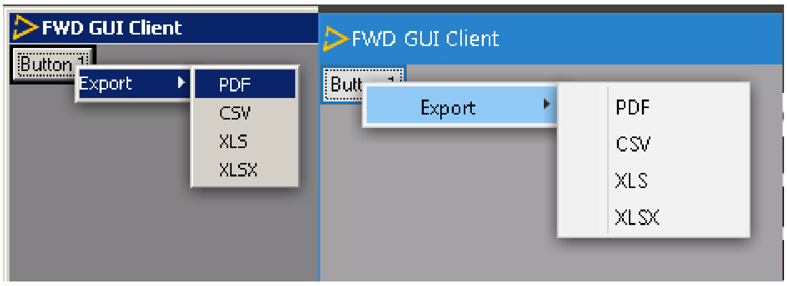
**#5 - 12/11/2017 07:46 PM - Eugenie Lyzenko**

*- File sub_menu1_fwd_20171211a.jpg added*

Task branch 3394a has been updated for review to revision 11255.

This update include several painting issues including:
1. Missing submenu arrow for Classic theme.
2. Wrong width calculation for Windows10 theme and truncated menu text.
3. Missing vertical menu separator for Windows10 theme.

The result now looks:



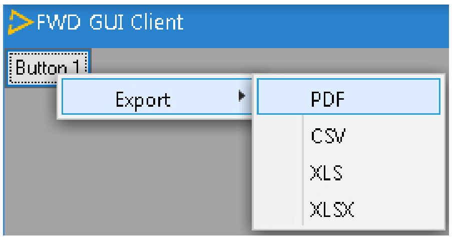
Continue with selected item background color issue and messages processing for menu hiding and selecting.

**#6 - 12/11/2017 09:08 PM - Eugenie Lyzenko**

*- File sub_menu1_fwd_20171211b.jpg added*

Task branch 3394a has been updated for review to revision 11256.

The fix for Windows10 selected menu item background to be:



**#7 - 12/12/2017 09:31 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11259.

This is the fix for "Export" item doesn't disappear issue. We need to unfocus and hide the root menu container.

Continue working on another event processing issue when click on "Export" causes the submenu items to close when it shouldn't. The logic in SubMenuItemGuiImpl looks correct but the further processing of the event in ThinClient causes the negative reaction. We need to find out the way to stop processing (simple event.consume() does not work) of this event once it reaches the SubMenuItemGuiImpl.mousePressed(), or may be even we need to use separate event processing loop for whole pop-up menu to fix another remaining issues with menu to be hidden when mouse became not over the pop-up menu. Something like the handling we used to control tooltip life cycle. Continue investigation.

**#8 - 12/13/2017 03:44 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11264.

Added fix for sub menu hiding issues as reaction on mouse click. Including fixes for notes 2 and 4 in original note. The change is not obvious so any notes/objections can be considered after review.

Continue with hover out of the main sub menu item issue.

**#9 - 12/13/2017 10:09 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11265.

This update adds hover base show/hide pop-up menu implementation. The only thing left in this area is to implement the delay between menu status change and mouse enter/leave the submenu. Now the menu is hiding immediately when the mouse leaving respective submenu root.

I'll continue working if we need the exact match to 4GL behavior. But if it is OK to have such a bit different implementation please let me know.

**#10 - 12/14/2017 05:20 AM - Greg Shah**

> The only thing left in this area is to implement the delay between menu status change and mouse enter/leave the submenu.

Is the 4GL actually having a "delay" or is it just ensuring the ordering of the events only occurs in a known pattern?

My guess is that the ordering is what is important. We need to match the 4GL in this way. I do not want to add a hard coded delay in our code.

Hynek: Please review the changes associated with this task.

**#11 - 12/14/2017 06:04 AM - Hynek Cihlar**

There is a regression caused by the changes. When the popup is displayed and another window (or the title bar of the same window) is clicked the popup stays visible. The expected behavior is that the popup should be closed.

There is another problem with multiple selected items. Open the submenu click on an item in the leaf submenu, reopen the submenu, the previously clicked item is now highlighted. The expected behavior is that the opened sub menu should not have any items highlighted. This one also occurs in trunk.

**#12 - 12/14/2017 07:04 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> There is a regression caused by the changes. When the popup is displayed and another window (or the title bar of the same window) is clicked the popup stays visible. The expected behavior is that the popup should be closed.

What is the testcase and recreate instructions to see this? With sub-menus.p I do not see the regression.

**#13 - 12/14/2017 07:06 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
>> There is a regression caused by the changes. When the popup is displayed and another window (or the title bar of the same window) is clicked the popup stays visible. The expected behavior is that the popup should be closed.
>
> What is the testcase and recreate instructions to see this? With sub-menus.p I do not see the regression.

I am able to recreate this with sub-menus.p.

1. Right click on the button.
2. Navigate to a submenu item (do not click on it) with a mouse cursor.
3. Move the mouse cursor out to the window's title bar (go there directly without entering the parent submenu).
4. Click in the title bar.

**#14 - 12/14/2017 07:39 AM - Hynek Cihlar**

Code review the #3381 changes in task branch 3394a.

MenuItemGuiImpl.selectMenuItem

```
// if root menu exists for submenu - we need to hide it as well
MenuGuiImpl menu = (MenuGuiImpl) Menu.getMenu(sm);
if (menu != null)
{
   menu.setFocusInt(null);
   menu.setVisible(false);
}
else
{
   ThinClient.getInstance().eventBracket(() -> sm.hideBody(true));
}
```

In the code above, shouldn't the root menu always exist for the submenu? If so the else branch is never executed.

What is the case that requires mouseEntered = true; in MenuItemGuiImpl.onFocusGained()? Focus gained can be achieved also by keyboard navigation, so turning this flag on when the mouse isn't actually entered can cause problems.

ScrollPopupGuiImpl
spcify => specify

ScrollPopupGuiImpl.getPhysOrigLocation()
"physical location of the pop-up stored at the time the menu become visible" is confusing, it should state that the stored location is the location of the click event that caused the popup to appear.

```
scrollHereOpt.addKeyListener(new KeyListener()
      {
         @Override
         public void onKeyPressed(KeyInput keyEvent)
         {
            // this is the handling place for menu selection event
         }
The body of @onKeyPressed()@ is empty.
```

ScrollBarGuiImpl
coodinate => coordinate

SubMenuGuiImpl.mouseEntered

```
         wnd.enableActivationChange(false);
         OverlayWindow owParent = (OverlayWindow)topLevelWindow();
         if (owParent != null)
         {
            owParent.enableActivationChange(false);
         }
```

In the code above, shouldn't the wnd and topLevelWindow() be equal?

Also it is not clear from the code which use case is the OverlayWindow.enableActivation addressing. Can you put more comments in the code to help the readers?

**#15 - 12/14/2017 07:43 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
> > Hynek Cihlar wrote:
> >
> > > There is a regression caused by the changes. When the popup is displayed and another window (or the title bar of the same window) is clicked the popup stays visible. The expected behavior is that the popup should be closed.
> >
> > What is the testcase and recreate instructions to see this? With sub-menus.p I do not see the regression.
>
> I am able to recreate this with sub-menus.p.
>
> 1. Right click on the button.
> 2. Navigate to a submenu item (do not click on it) with a mouse cursor.
> 3. Move the mouse cursor out to the window's title bar (go there directly without entering the parent submenu).
> 4. Click in the title bar.

What is your testing environment? Swing client, Web client or both?

**#16 - 12/14/2017 07:52 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> What is your testing environment? Swing client, Web client or both?

I can see this only in Swing client. The thing is that web client currently provides some more window logic on top of the Java client, which it shouldn't. Thanks to the extra logic the overlay window does go away and so it masks the problem there.

#### #17 - 12/14/2017 07:53 AM - Hynek Cihlar

Another problem is that the popup menu cannot be navigated by keyboard. However this probably exists in trunk, too.

#### #18 - 12/14/2017 07:58 AM - Hynek Cihlar

Another regression with the window menu bar. Run the uast/menu/Menubar.p try to navigate the menu, an exception below occurs.

```
java.lang.ClassCastException: com.goldencode.p2j.ui.client.gui.WindowGuiImpl cannot be cast to com.goldencode.
p2j.ui.client.gui.OverlayWindow
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl.showBody(SubMenuGuiImpl.java:1348)
    at com.goldencode.p2j.ui.client.SubMenu.processKeyEvent(SubMenu.java:407)
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl.processKeyEvent(SubMenuGuiImpl.java:817)
    at com.goldencode.p2j.ui.client.widget.TitledWindow.processKeyEvent(TitledWindow.java:295)
    at com.goldencode.p2j.ui.client.widget.AbstractWidget.processEvent(AbstractWidget.java:1618)
    at com.goldencode.p2j.ui.client.widget.TitledWindow.processEvent(TitledWindow.java:264)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.processEvent(WindowGuiImpl.java:1365)
    at com.goldencode.p2j.ui.chui.ThinClient.processEventsWorker(ThinClient.java:16744)
    at com.goldencode.p2j.ui.chui.ThinClient.pop(ThinClient.java:15562)
    at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:15545)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:15469)
    at com.goldencode.p2j.ui.chui.ThinClient.applyWorker(ThinClient.java:15230)
    at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:12269)
    at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:11753)
    at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:11696)
    at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:11650)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:124)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:757)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:412)
    at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:348)
    at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1170)
    at com.goldencode.p2j.net.Queue.transact(Queue.java:641)
    at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
    at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
```

**#19 - 12/14/2017 08:56 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Code review the #3381 changes in task branch 3394a.
> [...]
> In the code above, shouldn't the wnd and topLevelWindow() be equal?

No. The topLevelWindow() gets the overlay for submenu related window while wnd is the window used to display submenu children. And these are two different windows.

**#20 - 12/14/2017 08:59 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
> > What is your testing environment? Swing client, Web client or both?
>
> I can see this only in Swing client. The thing is that web client currently provides some more window logic on top of the Java client, which it shouldn't. Thanks to the extra logic the overlay window does go away and so it masks the problem there.

Please rebuild all. I still unable to recreate this. Pressing title bar caused the menu to disappear. Can you provide screenshot for regression wrong picture?

**#21 - 12/14/2017 09:17 AM - Hynek Cihlar**

*- File Peek 2017-12-14 15-12.gif added*

Eugenie Lyzenko wrote:

> Please rebuild all. I still unable to recreate this. Pressing title bar caused the menu to disappear. Can you provide screenshot for regression wrong picture?

Please see the attached animated gif.

**#22 - 12/14/2017 04:09 PM - Eugenie Lyzenko**

Greg Shah wrote:

> > The only thing left in this area is to implement the delay between menu status change and mouse enter/leave the submenu.
>
> Is the 4GL actually having a "delay" or is it just ensuring the ordering of the events only occurs in a known pattern?
>
> My guess is that the ordering is what is important.  We need to match the 4GL in this way.  I do not want to add a hard coded delay in our code.

I'm not having exact answer for question if there is an delay between mouse go out of menu and menu hiding. My guess and facts:
1. I think the 4GL uses native Windows libraries to draw menu so we see how the OS itself handles it.
2. There is a registry key HKEY_CURRENT_USER\Control Panel\Desktop\MenuShowDelay to specify delay before menu show after mouse is over item.
3. Some experiments: in Windows if menu is on screen and mouse left menu but **moving** - the menu is on screen. As soon as mouse stopped - the menu disappear in a 1 second, feels like there is a delay not 100% sure here. If mouse is returning to submenu that is root for opened menu - the opened menu is not hiding.

So looks like for now we can stay with reaction provided by correct event ordering processing. In this case we will have some delay from CPU used by event processing.

**#23 - 12/14/2017 06:35 PM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Code review the #3381 changes in task branch 3394a.
> ...
> The body of onKeyPressed() is empty.

We used message box for testing and issue resolution purpose, isn't it? It is not a part of the FWD code, right?

**#24 - 12/14/2017 06:50 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11266.

The review notes resolution. The code in OverlayWindow for activation condition was removed because now it is not required, other change does necessary work.

The remaining opened question is the our "delay" strategy for submenu open/close.

#### #25 - 12/14/2017 11:19 PM - Eugenie Lyzenko

Task branch 3394a has been updated for review to revision 11267.

Greg,

This is the approach I suggest to use for menu show/hide delays. This does not make delay for whole UI, we can use Java 5 schedulers and pretty simple with implementation. The single millisecond delay value is used for both show and hide. The execution can be cancelled when necessary.

I have kept the old code to safely and fast restore if the approach is not acceptable. If OK I'll cleanup the code from commented out lines.

#### #26 - 12/15/2017 06:12 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> We used message box for testing and issue resolution purpose, isn't it? It is not a part of the FWD code, right?

The new code with menu items opening modal windows will use its own menu items, not the Scroll Here.

#### #27 - 12/15/2017 06:22 AM - Greg Shah

Code Review Task Branch 3394a Revision 11267

The delay implementation seems good. I'm still not sure I like having it there. Some questions:

1. Does the delay implementation improve the usability of the menus?
2. Is this always there by default on all versions of Windows?
3. I assume a customer might have disabled this or customized the wait time in Windows. If so, we would probably need to implement a similar configurable capability.

#### #28 - 12/15/2017 06:37 AM - Hynek Cihlar

Another exception when menu bar menu item is mouse-highlighted and arrow keys pressed.

```
Caused by: java.lang.NullPointerException
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.cycleFocus(AbstractContainer.java:1849)
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl$SubMenuBody.focusWorker(SubMenuGuiImpl.java:1493)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.nextFocus(AbstractContainer.java:948)
    at com.goldencode.p2j.ui.client.MenuItem.processKeyEvent(MenuItem.java:302)
    at com.goldencode.p2j.ui.client.gui.MenuItemGuiImpl.processKeyEvent(MenuItemGuiImpl.java:473)
    at com.goldencode.p2j.ui.client.widget.TitledWindow.processKeyEvent(TitledWindow.java:280)
    at com.goldencode.p2j.ui.client.widget.AbstractWidget.processEvent(AbstractWidget.java:1616)
    at com.goldencode.p2j.ui.client.widget.TitledWindow.processEvent(TitledWindow.java:249)
    at com.goldencode.p2j.ui.client.gui.OverlayWindow.processEvent(OverlayWindow.java:633)
    at com.goldencode.p2j.ui.chui.ThinClient.processEventsWorker(ThinClient.java:16648)
    at com.goldencode.p2j.ui.chui.ThinClient.pop(ThinClient.java:15471)
    at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:15454)
```

#### #29 - 12/15/2017 06:39 AM - Hynek Cihlar

Eugenie, I think there is another regression in the menu bar. The delayed submenu hiding doesn't seem to apply there in the native 4GL.

1. Run menu/menubar.p
2. Open a submenu.
3. Do the same routine you do in the popup menu to replicate the delay behavior.
4. The submenu will be closed while it should stay on.

#### #30 - 12/15/2017 11:58 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Another exception when menu bar menu item is mouse-highlighted and arrow keys pressed.
>
> [...]

I think this exception is related to tab item changes, since focus traversal widgets are from the tab items of the containing field group. I lost this menu widgets when implemented the corresponding changes. It seems that for menus

```
  protected void cycleFocus(boolean direction)
   {
      List<Widget<O>> widgets = getTabItemList();

      Widget<O> anchor = focus();
      if (anchor instanceof Frame)
      {
         anchor = ((Frame) anchor).getContentPane();
      }

      Iterable<Widget<O>> iter = (direction) ?
                                  Iterables.directFrom(widgets, anchor) :
                                  Iterables.reverseFrom(widgets, anchor);

      for (Widget<O> widget : iter)
      {
         if (checkWidget(widget, direction))
            return;
      }

      // cyclic transition to the first or last element
      Widget<O> nextWidget = direction ? widgets.get(0) : widgets.get(widgets.size() - 1);
      checkWidget(nextWidget, direction);
```

```
    }
```

getTabItemList() should return all child menus. I didn't develop this case. May be if we override getTabItemList() for menus as widgetsAsList(), then this exception will be fixed.

**#31 - 12/15/2017 03:23 PM - Eugenie Lyzenko**

Greg Shah wrote:

> Code Review Task Branch 3394a Revision 11267
> ...
> 1. Does the delay implementation improve the usability of the menus?

As to me I can say having the delay does not improve usability, is an annoying point for me instead. Especially when I need to leave the menu and enter it again to show submenu. We have some delay anyway due to event processing usage.

> 2. Is this always there by default on all versions of Windows?

Yes, found the same reaction on Windows XP and Windows 7.

> 3. I assume a customer might have disabled this or customized the wait time in Windows. If so, we would probably need to implement a similar configurable capability.

Agreed if we decide to implement this.

As alternative I can offer to use the following: when ShubMenuGuiImpl needs to show/hide menu items it post the special message to itself instead of immediate show/hide. This will produce some delay to route and receive message say SE_MENU_(SHOW|HIDE). On receiving this message the ShubMenuGuiImpl will do actual show/hide. The disadvantage is the extra load to the event processing loop. As I noted above my preference is not to use any delay at all.

**#32 - 12/15/2017 03:30 PM - Greg Shah**

Please comment out the delay code to disable it.  I want to ensure it can be added later if needed, but for now I prefer to avoid the delay.  If customers need the delay, we will deal with it later.

**#33 - 12/15/2017 03:48 PM - Eugenie Lyzenko**

Greg Shah wrote:

> Please comment out the delay code to disable it.  I want to ensure it can be added later if needed, but for now I prefer to avoid the delay.  If customers need the delay, we will deal with it later.

OK.

**#34 - 12/15/2017 04:52 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11268.

Undo for menu show/hide delay implementation. Continue with menu related issues/regressions fix.

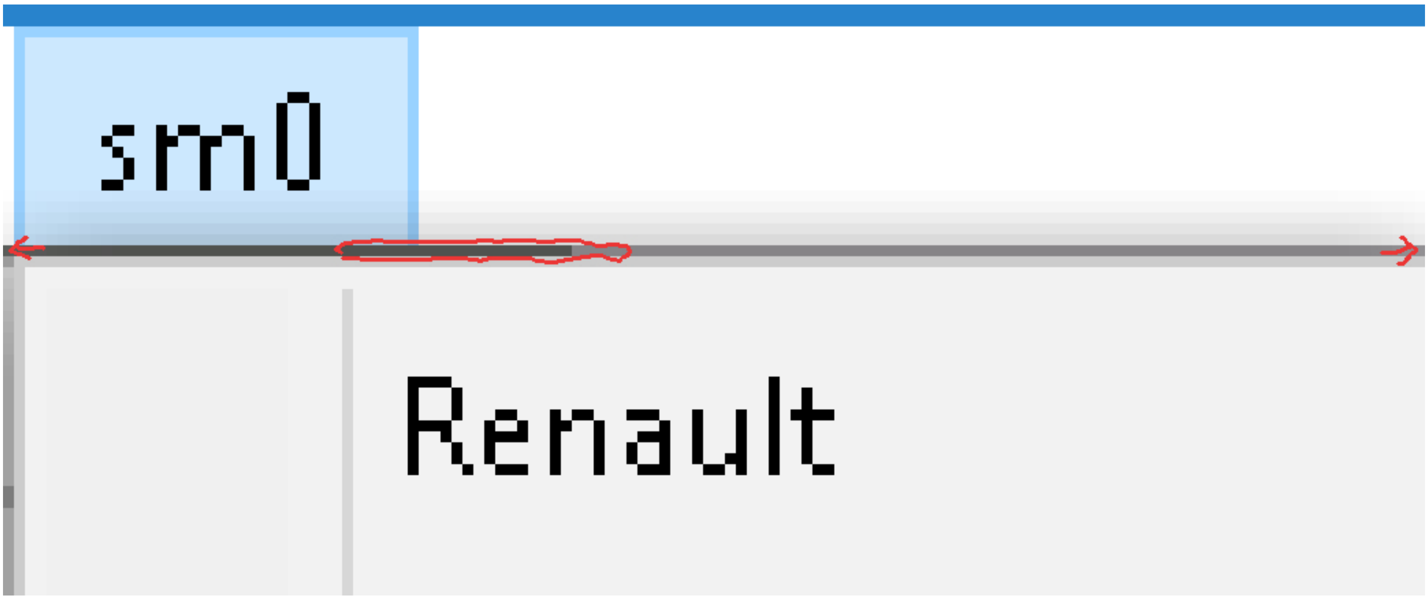
**#35 - 12/15/2017 05:57 PM - Eugenie Lyzenko**

*- File menubar_fwd_issue_20171215a.jpg added*

Hynek Cihlar wrote:

> Eugenie, I think there is another regression in the menu bar. The delayed submenu hiding doesn't seem to apply there in the native 4GL.
>
> 1. Run menu/menubar.p
> 2. Open a submenu.
> 3. Do the same routine you do in the popup menu to replicate the delay behavior.
> 4. The submenu will be closed while it should stay on.

I think the root cause here is the 1-pixel hole between main menu bar and submenu. This horizontal line is neither belonging to main bar not to submenu. So when moving mouse to submenu we got into space that out of menu area, get mouseLeave event and submenu is closing. Looks like we need to fix the submenu location one pixel up to get rid of this. See the 8x scaled picture with area I mean  marked in red:



**#36 - 12/15/2017 11:07 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11272.

This is the fix for menu regressions/issues/notes resolution. I have commented out the code that stops correct working of the menu/menubar.p testcase. Here it is SubMenuGuiImpl.hideBody(), line 739:

```
//       if (!MenuGuiImpl.isKeepOpenOnBodyClose() && Menu.isMenubarElement(this))
//       {
//          MenuGuiImpl.resetMbarPressedState();
//       }
```

Can someone clarify what this code was added for? What is the case the code become working. Asking because this code stops proper working of the menu/menubar.p testcase.

**#37 - 12/17/2017 04:57 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > Code Review Task Branch 3394a Revision 11267
> > ...
> > 1. Does the delay implementation improve the usability of the menus?
>
> As to me I can say having the delay does not improve usability, is an annoying point for me instead. Especially when I need to leave the menu and enter it again to show submenu. We have some delay anyway due to event processing usage.

Guys, the delay improves usability in cases when you navigate the menu by mouse. Without the delay you must trace the opened menu bodies exactly, with the delay you can relax and afford to leave the menu bodies temporarily.

**#38 - 12/17/2017 05:00 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
> > Eugenie, I think there is another regression in the menu bar. The delayed submenu hiding doesn't seem to apply there in the native 4GL.
> >
> > 1. Run menu/menubar.p
> > 2. Open a submenu.
> > 3. Do the same routine you do in the popup menu to replicate the delay behavior.
> > 4. The submenu will be closed while it should stay on.
>
> I think the root cause here is the 1-pixel hole between main menu bar and submenu. This horizontal line is neither belonging to main bar not to submenu. So when moving mouse to submenu we got into space that out of menu area, get mouseLeave event and submenu is closing. Looks like we need to fix the submenu location one pixel up to get rid of this. See the 8x scaled picture with area I mean marked in red:

But this doesn't occur in trunk. Was the extra space introduced by the task branch changes?

**#39 - 12/17/2017 05:53 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Task branch 3394a has been updated for review to revision 11272.
>
> This is the fix for menu regressions/issues/notes resolution. I have commented out the code that stops correct working of the menu/menubar.p testcase. Here it is SubMenuGuiImpl.hideBody(), line 739:
> [...]
>
> Can someone clarify what this code was added for? What is the case the code become working. Asking because this code stops proper working of the menu/menubar.p testcase.

There is a case (or possibly cases) when the root menu bar item highlight must be preserved when the submenu body is closed. Which I think the piece of code you mention covers. The case goes as follows.

1. Run menubar.p and make the window menu appear.
2. Open a submenu.
3. Press Esc, the submenu body will close but the menu bar item will keep highlighted.

**#40 - 12/17/2017 09:14 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
> > Hynek Cihlar wrote:
> > I think the root cause here is the 1-pixel hole between main menu bar and submenu. This horizontal line is neither belonging to main bar not to submenu. So when moving mouse to submenu we got into space that out of menu area, get mouseLeave event and submenu is closing. Looks like we need to fix the submenu location one pixel up to get rid of this. See the 8x scaled picture with area I mean  marked in red:
>
> But this doesn't occur in trunk. Was the extra space introduced by the task branch changes?

Yes, but because in trunk there is no functionality to hide submenu once mouse is left the parent menu. New feature just highlights the hidden issue.

**#41 - 12/17/2017 11:18 AM - Greg Shah**

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
> > Greg Shah wrote:
> >
> > > Code Review Task Branch 3394a Revision 11267
> > > ...
> > > 1. Does the delay implementation improve the usability of the menus?
> >
> > As to me I can say having the delay does not improve usability, is an annoying point for me instead. Especially when I need to leave the menu and enter it again to show submenu. We have some delay anyway due to event processing usage.
>
> Guys, the delay improves usability in cases when you navigate the menu by mouse. Without the delay you must trace the opened menu bodies exactly, with the delay you can relax and afford to leave the menu bodies temporarily.

OK, Eugenie please put it back in.

**#42 - 12/17/2017 12:43 PM - Eugenie Lyzenko**

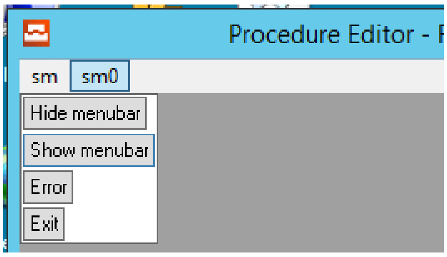
*- File menubar_after_ESC_fwd_20171217a.jpg added*

*- File menubar_after_ESC_4gl.jpg added*

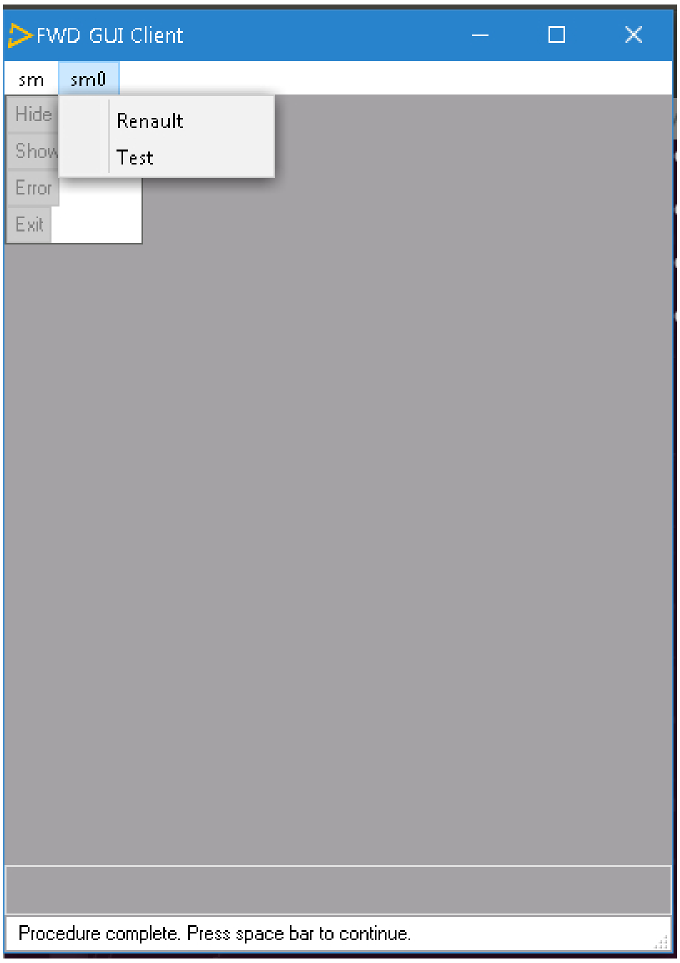
Hynek Cihlar wrote:

> > Can someone clarify what this code was added for? What is the case the code become working. Asking because this code stops proper working of the menu/menubar.p testcase.
>
> There is a case (or possibly cases) when the root menu bar item highlight must be preserved when the submenu body is closed. Which I think the piece of code you mention covers. The case goes as follows.
>
> 1. Run menubar.p and make the window menu appear.
> 2. Open a submenu.
> 3. Press Esc, the submenu body will close but the menu bar item will keep highlighted.

And here is what we have in 4GL on Windows:



Do you expect different? BTW in FWD reaction to ESC key is the converted menubar.p application finishes working:



Procedure complete. Press space bar to continue.

Do you have another result?

**#43 - 12/17/2017 05:50 PM - Eugenie Lyzenko**

Greg Shah wrote:

> > Guys, the delay improves usability in cases when you navigate the menu by mouse. Without the delay you must trace the opened menu bodies exactly, with the delay you can relax and afford to leave the menu bodies temporarily.
>
> OK, Eugenie please put it back in.

Done.

Task branch 3394a has been updated for review to revision 11273.

Working on adding directory base support for this feature as configurable one.

**#44 - 12/18/2017 03:17 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Yes, but because in trunk there is no functionality to hide submenu once mouse is left the parent menu. New feature just highlights the hidden issue.

Just to be sure this doesn't get lost. The main bar doesn't have this feature in native 4GL - it doesn't hide its submenu once mouse is left the parent menu.

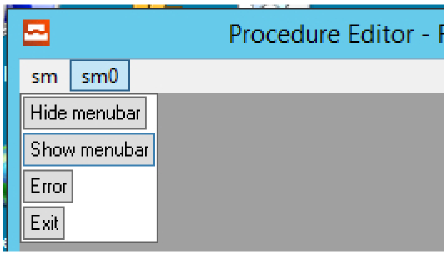
**#45 - 12/18/2017 03:25 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
> > Can someone clarify what this code was added for? What is the case the code become working. Asking because this code stops proper working of the menu/menubar.p testcase.
>
> There is a case (or possibly cases) when the root menu bar item highlight must be preserved when the submenu body is closed. Which I think the piece of code you mention covers. The case goes as follows.

1. Run menubar.p and make the window menu appear.
2. Open a submenu.
3. Press Esc, the submenu body will close but the menu bar item will keep highlighted.

And here is what we have in 4GL on Windows:



Do you expect different?

I expect this to happen, the root menu item should be kept highlighted. And hence I think the commented out code in note 36 might be needed.

Do you have another result?

I get the same ill behavior.

**#46 - 12/18/2017 12:58 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11276, 11277.

Added the ability to configure menu delay. The section is under UI themes configuration:

```
        <node class="container" name="theme">
...
          <node class="integer" name="submenu-delay">
```

```
            <node-attribute name="value" value="IntegerValueHere"/>
          </node>
...
        </node>
```

It is possible to set up 0 delay. This means we do not use the show/hide delay and moreover the Java class SubMenuGuiImpl uses old code style for handling show/hide and task scheduler is not used.

**#47 - 12/18/2017 03:11 PM - Greg Shah**

Code Review Task Branch 3394a Revisions 11276/11277

I like the approach.

**#48 - 12/19/2017 07:47 AM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11279.

This change contains the fix for ClassicTheme painting for submenu item that is not hovered. In this case the menu background should not have highlight background.

Also I have returned the change to control OverlayWindow activation/deactivation ability. After removing we have the issue for mouse click on sub-menu when items body is opened. Need to suppress overlay activation to reflect 4GL behavior. However the implementation is changed to apply only for pop-up menus and avoid class cast issue.

Will continue testing with different menu testcases to be sure all is working as expected.

**#49 - 12/19/2017 08:38 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11286.

The minor changes to resolve menu related issues and regressions. For both pop-up and menu-bar related menus. The focus change and mouse enter flag are independent. Also the activation change logic must be limited to pop-up menus only.

**#50 - 12/19/2017 09:28 PM - Eugenie Lyzenko**

*- File sub_menu_classic_fwd_pop-up_20171219.jpg added*

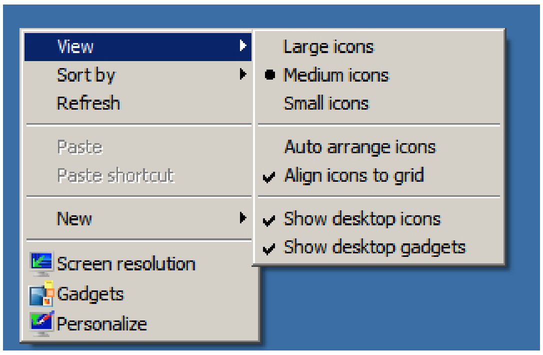
*- File sub_menu_classic_pop-up.jpg added*

*- File sub_menu_classic_menu-bar.jpg added*

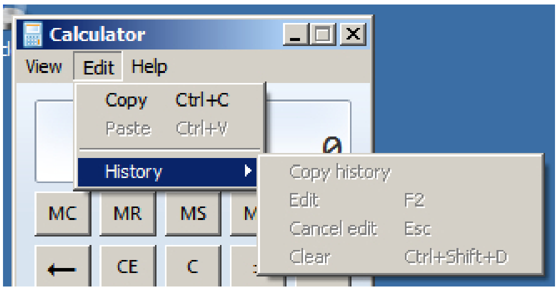
*- File sub_menu_classic_fwd_menu-bar_20171219.jpg added*

There are some UI painting deviations for classic theme implementation. Consider the following native Windows pictures for menu usage:

Pop-up submenu:

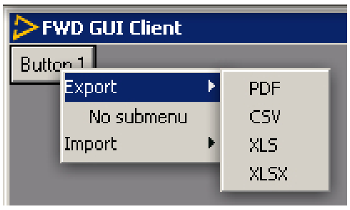


Menu-bar example:

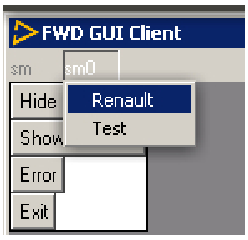


The similar cases for FWD usage:

Pop-up submenu:



Menu-bar example:



We need to fix this later or sooner. The question is how urgent this to complete? Do I need to work on this now?

#### #51 - 12/20/2017 12:50 AM - Eugenie Lyzenko

Task branch 3394a has been updated for review to revision 11288.

This is the fix for classic theme menu painting deviations.

#### #52 - 12/20/2017 03:10 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Task branch 3394a has been updated for review to revision 11288.
>
> This is the fix for classic theme menu painting deviations.

With the revision there are still some issues with the main bar. Please see the attached gif. The submenu body still closes when the cursor moves out of the root menu item (no click). Also note the multiple highlighted items.

Another point is the way the menu bar should handle Esc key. On the first Esc press the submenu bodies should close and the root menu item should stay highlighted and focused, om the second Esc press, the menu item should loose the highlight and focus.  You can test this with uast/menu/menubar_no_endkey.p.

#### #53 - 12/20/2017 03:38 AM - Hynek Cihlar

*- File Peek 2017-12-20 09-04.gif added*

Hynek Cihlar wrote:

> With the revision there are still some issues with the main bar. Please see the attached gif.

And the gif attached.

#### #54 - 12/20/2017 11:32 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

> Another point is the way the menu bar should handle Esc key. On the first Esc press the submenu bodies should close and the root menu item should stay highlighted and focused, om the second Esc press, the menu item should loose the highlight and focus.  You can test this with uast/menu/menubar_no_endkey.p.

The file is not able to be even compiled in 4GL:

```
...
repeat on next on endkey undo, next:
    wait-for choose of ext.
end.
```

is wrong. Did you use instead here:

```
repeat on endkey undo, next:
    wait-for choose of ext.
end.
```

Or another version? Please chech and let me know what is the uast/menu/menubar_no_endkey.p file that is working in 4GL?

**#55 - 12/20/2017 12:41 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11291.

Fixes for issue except ESC key press reaction, for which I still do not have the 4GL test to run because not clear how to use the current uast/menu/menubar_no_endkey.p, this test is not able to be stopped otherwise than killing from task manager.

**#56 - 12/20/2017 01:53 PM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Fixes for issue except ESC key press reaction, for which I still do not have the 4GL test to run because not clear how to use the current uast/menu/menubar_no_endkey.p, this test is not able to be stopped otherwise than killing from task manager.

The compilation error is fixed and checked in, you can try now.

**#57 - 12/20/2017 10:18 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11293.

This is suggested fix for ESC key processing in GUI menu widget. The idea is to intercept ESC generated END-ERROR event and perform respective actions for GUI SubMenu and MenuItem. I think in 4GL the ESC handling is not even in Progress event queue because it is hidden in Windows native menu related library API internals.

Continue testing with multiple selection issue regression.

**#58 - 12/21/2017 06:56 AM - Greg Shah**

Code Review Task Branch 3394a Revision 11293

I'm OK with the changes.

> I think in 4GL the ESC handling is not even in Progress event queue because it is hidden in Windows native menu related library API internals.

I'm sure you are right. It would be a better implementation if we had the a menu-specific event loop and could hide this. But with our implementation, it is not a simple match to do that. Your approach is the best way based on how our code works.

**#59 - 12/21/2017 01:03 PM - Eugenie Lyzenko**

Task branch 3394a has been updated for review to revision 11295.

This is new approach to control menu item selection. Now for both MenuItem and SubMenu classes the highlighted attribute is used to find out whether the item to be drawn selected or not. The new thing is now we have static method in MenuItemGuiImpl to make the menu item highlighted. The key point is once some item to become highlighted, all other items in the same menu container must be de-highlighted. Because only one selected item is possible for given menu.

The difference between old approach where we used mouseEntered flag is inconsistent because it is possible to have two menu items with mouseEntered = true it we change this flag when focus changed. This can happen when we simultaneously control the menu with both mouse and keyboard for example.

Hynek, you saw the multiple selections more often than I so please if with this change this issue happen - please let me know with instructions (and animated gif is a great help).

**#60 - 12/21/2017 04:13 PM - Hynek Cihlar**

*- File Peek 2017-12-21 22-04.gif added*

With the latest 3394a the multiple selection issue is fixed and the Esc key works properly. However there are some new ones:

1. The root menu items sometime keep highlighted even when mouse-clicked outside the menu.
2. There is a noticable lag when navigating from one root submenu to another, in native 4GL the submenus are switched instantly.
3. When a menu item is clicked, the menubar stops to draw itself.
4. Menu item is highlighted when its submenu is opened, it should not be.

You can see all the points in the attached gif.

**#61 - 12/21/2017 04:20 PM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> With the latest 3394a the multiple selection issue is fixed and the Esc key works properly. However there are some new ones:
>
> 2. There is a noticable lag when navigating from one root submenu to another, in native 4GL the submenus are switched instantly.

This is side effect of the show/hide delay introduced. Disable delay in server.xml by setting time to 0 - the submenus will be opened as fast as it can. I do not think we need to add special logic to differentiate this case from other hide/show cases.

**#62 - 12/21/2017 04:22 PM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> > 2. There is a noticable lag when navigating from one root submenu to another, in native 4GL the submenus are switched instantly.
>
> This is side effect of the show/hide delay introduced. Disable delay in server.xml by setting time to 0 - the submenus will be opened as fast as it can. I do not think we need to add special logic to differentiate this case from other hide/show cases.

The delay should not be applied to (root) submenus. Please compare it to the native app.

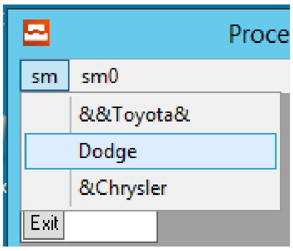
**#63 - 12/21/2017 04:32 PM - Eugenie Lyzenko**

*- File menubar_opened_submenu_4gl.jpg added*

Hynek Cihlar wrote:

> 4. Menu item is highlighted when its submenu is opened, it should not be.

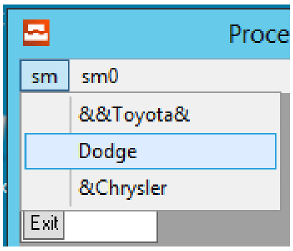
I do not understand this point. Please see the attached opened 4GL menu case:



Is it what you mean telling "...when its submenu is opened"?

**#64 - 12/21/2017 04:35 PM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
> > 4. Menu item is highlighted when its submenu is opened, it should not be.
>
> I do not understand this point. Please see the attached opened 4GL menu case:
>
> 
>
>
> Is it what you mean telling "...when its submenu is opened"?

Yes, this is it. You can see it in the gif close to the end of the playback.

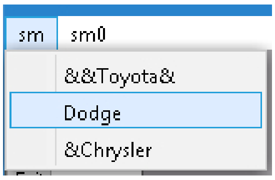
**#65 - 12/21/2017 04:42 PM - Eugenie Lyzenko**

*- File menubar_opened_submenu_fwd_20171221a.jpg added*

Hynek Cihlar wrote:

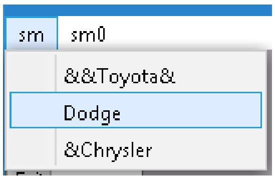
> Yes, this is it. You can see it in the gif close to the end of the playback.

But in FWD we have the same:



**#66 - 12/21/2017 04:47 PM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
> > Yes, this is it. You can see it in the gif close to the end of the playback.
>
> But in FWD we have the same:
>
> 
>

I'll try to rephrase what the problem is. When you open a submenu body, none of the menu items in it must be already highlighted. The items must highlight only when you navigate to them. In the gif you can see that when a submenu sm0 was opened the menu item Test was already highlighted.

**#67 - 12/21/2017 05:09 PM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> I'll try to rephrase what the problem is. When you open a submenu body, none of the menu items in it must be already highlighted. The items must highlight only when you navigate to them. In the gif you can see that when a submenu sm0 was opened the menu item Test was already highlighted.

OK. Understood.

**#68 - 12/21/2017 08:24 PM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> With the latest 3394a the multiple selection issue is fixed and the Esc key works properly. However there are some new ones:
>
> 1. The root menu items sometime keep highlighted even when mouse-clicked outside the menu.
> 2. There is a noticable lag when navigating from one root submenu to another, in native 4GL the submenus are switched instantly.
> 3. When a menu item is clicked, the menubar stops to draw itself.
> 4. Menu item is highlighted when its submenu is opened, it should not be.

Task branch 3394a has been updated for review to revision 11300.

The update includes notes 1-4 resolution. And minor fix for Windows10 theme menu-bar painting for hovered/pressed item state. The colors are not the same as used for menu item selected state.

**#69 - 12/28/2017 11:14 AM - Greg Shah**

The changes are included in task branch 3394a which has been merged to trunk as revision 11214.

Hynek: Do you have any further concerns?

Can I close this task or are there any more open issues?

**#70 - 12/28/2017 11:19 AM - Hynek Cihlar**

Greg Shah wrote:

> The changes are included in task branch 3394a which has been merged to trunk as revision 11214.

> Hynek: Do you have any further concerns?

I don't know yet. I will do another round of functional changes and will do a review of the last changes, if I will be able to locate them in trunk now.

**#71 - 12/28/2017 12:29 PM - Hynek Cihlar**

*- File Peek 2017-12-28 18-26.gif added*

*- File Peek 2017-12-28 18-12.gif added*

*- File Peek 2017-12-28 18-22.gif added*

Hynek Cihlar wrote:

> Greg Shah wrote:
>
>> The changes are included in task branch 3394a which has been merged to trunk as revision 11214.
>>
>> Hynek: Do you have any further concerns?
>
> I don't know yet. I will do another round of functional changes and will do a review of the last changes, if I will be able to locate them in trunk now.

I tested trunk and still found some functional issues.

1. Run menu/menubar.p, press Alt key to make the menu shortcuts to appear (you may need to press the key multiple times), an NPE is thrown:

```
Caused by: java.lang.NullPointerException
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl.mouseExited(SubMenuGuiImpl.java:974)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.lambda$processEvent$4(WindowGuiImpl.java:1177)
    at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:15547)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:15491)
    at com.goldencode.p2j.ui.chui.ThinClient.independentEventDrawingBracket(ThinClient.java:15361)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.processEvent(WindowGuiImpl.java:1164)
    at com.goldencode.p2j.ui.chui.ThinClient.checkForSystemEvent(ThinClient.java:14655)
    at com.goldencode.p2j.ui.chui.ThinClient.waitForEvent(ThinClient.java:14551)
```

2. When root menu item in a menu-bar is highlighted, navigating to another root menu item sometimes doesn't make its body open.
Peek%202017-12-28%2018-12.gif

3. Submenu body isn't closed when it should be, this seems to be an intermittent issue.
Peek%202017-12-28%2018-22.gif

4. Item 3 from note 60 is still an issue.

5. Esc handling - sometimes when menu is exited the Esc key doesn't cause the endkey condition. This seems to be somehow related to item 3.
Peek%202017-12-28%2018-26.gif

**#72 - 01/02/2018 02:48 PM - Eugenie Lyzenko**

Created task branch 3381a from trunk revision 11214.

**#73 - 01/03/2018 08:53 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> 4. Item 3 from note 60 is still an issue.

I do not have this one. If you mean: 3. When a menu item is clicked, the menubar stops to draw itself..

Otherwise please provide the scenario to recreate.

**#74 - 01/03/2018 02:47 PM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
>> 4. Item 3 from note 60 is still an issue.
>
> I do not have this one. If you mean: 3. When a menu item is clicked, the menubar stops to draw itself..
>
> Otherwise please provide the scenario to recreate.

Use the latest uast/menu/menubar.p, click on the item in the sm2 or sm3 submenu.

**#75 - 01/03/2018 07:45 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11215.

This is the fix for notes 2 and 4 in remaining issues list. Continue with the rest.

**#76 - 01/04/2018 04:48 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11216.

The update contains fixes to resolve menu related notes/issues (1, 3, 5). The NPE was caused by artificial mouseExited() call with null as parameter. Now taking into account. So if there will be no new notes the issues resolution is completed and branch is ready to be merged (only GUI code affected, the harness cycle is useless I guess).

**#77 - 01/08/2018 12:58 PM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Task branch 3381a has been updated for review to revision 11216.
>
> The update contains fixes to resolve menu related notes/issues (1, 3, 5). The NPE was caused by artificial mouseExited() call with null as parameter. Now taking into account. So if there will be no new notes the issues resolution is completed and branch is ready to be merged (only GUI code affected, the harness cycle is useless I guess).

Code review 3381a/11216.

The code changes seem to be reasonable. I just checked in some typos fixes.

I found the following runtime issues.

1. Abend when navigating the root menu-bar items with a keyboard:

- Run menu/menubar.p and make the menu bar visible.
- Select a menu item from the root submenu then press the right or left arrow key. The exception below is thrown and the client abends.

```
Caused by: java.lang.NullPointerException
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.cycleFocus(AbstractContainer.java:1849)
    at com.goldencode.p2j.ui.client.Menu.focusWorker(Menu.java:376)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.prevFocus(AbstractContainer.java:957)
    at com.goldencode.p2j.ui.client.MenuItem.processKeyEvent(MenuItem.java:341)
    at com.goldencode.p2j.ui.client.gui.MenuItemGuiImpl.processKeyEvent(MenuItemGuiImpl.java:512)
    at com.goldencode.p2j.ui.client.widget.TitledWindow.processKeyEvent(TitledWindow.java:295)
    at com.goldencode.p2j.ui.client.widget.AbstractWidget.processEvent(AbstractWidget.java:1619)
    at com.goldencode.p2j.ui.client.widget.TitledWindow.processEvent(TitledWindow.java:264)
    at com.goldencode.p2j.ui.client.gui.OverlayWindow.processEvent(OverlayWindow.java:642)
```

2. Root submenus can't be navigated by left/right keys until a menu item is highlighted.

- Run menu/menubar.p and make the menu bar visible.
- Press Alt to make the first menu bar submenu label highlighted.
- Pressing left or right keys doesn't change the label highlighted.

3. A menu item still stays highlighted when cursor moves out.

- Run menu/menubar.p and make the menu bar visible.
- Open a submenu body and highlight a menu item by hovering the cursor over it.
- Move the cursor out quickly, the item stays highlighted.

4. Esc key still doesn't work as expected.

- Run menu/menubar.p and make the menu bar visible.
- Press Alt and click on the highlighted root menu bar item.
- Press Esc, the client terminates instead the menu submenu body should be closed.

#### #78 - 01/08/2018 09:42 PM - Eugenie Lyzenko

Task branch 3381a has been updated for review to revision 11218.

This is the fixes for notes 1-4. As to note/issue 3. Sometimes the mouse moving become too fast. This causes the input events queue overflow and the mouse leave event is missed. That's why the item can be highlighted even if mouse actually not over menu item. I guess this artifact we need to accept because if we are not receiving event we can not know when to turn the highlight off. Otherwise we have to start another watching thread to ask if the mouse over the menu. Please check if you can get wrong behavior now for point 3.

#### #79 - 01/09/2018 04:18 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

> Task branch 3381a has been updated for review to revision 11218.
>
> This is the fixes for notes 1-4. As to note/issue 3. Sometimes the mouse moving become too fast. This causes the input events queue overflow and the mouse leave event is missed.

Eugenie, which queue are you referring to? Can you point me in the code?

#### #80 - 01/09/2018 05:46 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
> > Task branch 3381a has been updated for review to revision 11218.
> >
> > This is the fixes for notes 1-4. As to note/issue 3. Sometimes the mouse moving become too fast. This causes the input events queue overflow and the mouse leave event is missed.
>
> Eugenie, which queue are you referring to? Can you point me in the code?

I mean the main event queue in ThinClient event loop where the mouse events are posting from SwingMouseHandler.mouseExited() interceptor installed in SwingEmulatedWindow(). I think the true solution can be the creation of dedicated mouse/key event processing queue for menus separating them from main event queue. The only we need to know from menu is if we made item selection (and what item selected) or menu cancelled without any selection.

**#81 - 01/09/2018 05:51 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
> > Eugenie Lyzenko wrote:
> >
> > > Task branch 3381a has been updated for review to revision 11218.
> > >
> > > This is the fixes for notes 1-4. As to note/issue 3. Sometimes the mouse moving become too fast. This causes the input events queue overflow and the mouse leave event is missed.
> >
> > Eugenie, which queue are you referring to? Can you point me in the code?
>
> I mean the main event queue in ThinClient event loop where the mouse events are posting from SwingMouseHandler.mouseExited() interceptor installed in SwingEmulatedWindow().

The mouse events are posted to EventManager.WorkArea.osEvents. This is an unbounded queue, so there should not be an overflow. Or am I missing something?

**#82 - 01/09/2018 06:06 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> > I mean the main event queue in ThinClient event loop where the mouse events are posting from SwingMouseHandler.mouseExited() interceptor installed in SwingEmulatedWindow().
>
> The mouse events are posted to EventManager.WorkArea.osEvents. This is an unbounded queue, so there should not be an overflow. Or am I missing something?

Yes, it is true. However I think the mouse exited event is not properly posted in this case. Not sure, may be it is just not coming from JVM. I mean back-end Swing component eats the event if mouse is too fast.

**#83 - 01/09/2018 06:07 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek Cihlar wrote:
>
> > > I mean the main event queue in ThinClient event loop where the mouse events are posting from SwingMouseHandler.mouseExited() interceptor installed in SwingEmulatedWindow().
> >
> > The mouse events are posted to EventManager.WorkArea.osEvents. This is an unbounded queue, so there should not be an overflow. Or am I missing something?
>
> Yes, it is true. However I think the mouse exited event is not properly posted in this case. Not sure, may be it is just not coming from JVM. I mean back-end Swing component eats the event if mouse is too fast.

Does web client work ok?

**#84 - 01/09/2018 08:19 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> > Yes, it is true. However I think the mouse exited event is not properly posted in this case. Not sure, may be it is just not coming from JVM. I mean back-end Swing component eats the event if mouse is too fast.
>
> Does web client work ok?

Yes, I can not reproduce the issue with as fast leave as I can.

Do you think it means some issue is Swing specific code? May be SwingMouseHandler.mouseExited() that uses TC.invokeLaterOS(()-> is the cause? Or something inside this block. I think it may be inability to find the mouse event source in:

```
      tc.invokeLaterOS(() ->
      {
         // make sure mouseExited is processed AFTER mouse-dragging is finished
         if (pressed)
         {
            delayedMouseExited = e;
            return;
         }

         Widget<?> mouseSource = findMouseSource(e);
         if (mouseSource != null)
         {
...
```

What do you think?

OK. Need to perform some debugging.

**#85 - 01/09/2018 08:24 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> OK. Need to perform some debugging.

Eugenie, I suggest you put a couple of trace statements in SwingMouseHandler, ThinClient and the subjected widgets and check whether the mouse event is properly posted and how far does it get.

**#86 - 01/09/2018 01:22 PM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
> > OK. Need to perform some debugging.
>
> Eugenie, I suggest you put a couple of trace statements in SwingMouseHandler, ThinClient and the subjected widgets and check whether the mouse event is properly posted and how far does it get.

Looks like I've found the place we have an issue. Consider both SwingMouseHandler.mouseMoved() and SwingMouseHandler.mouseExited(). The actual MouseExited event delivered to MenuItemGuiImpl is an artificial event generated in SwingMouseHandler.mouseMoved(), not by real SwingMouseHandler.mouseExited() (which does not even reach target MenuItemGuiImpl). It is not clear why we have so complicated implementation but this explains why MenuItemGuiImpl ignores fast mouse exit - MouseMove is generated not for every pixel change, for fast moving the distance between two move events can be more than one pixel, and we got to the location when mouse move become far away.

Looking for solution.

**#87 - 01/09/2018 03:10 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11219.

This is the fix for point 3 with mouse exited event for Swing client. If for some reasons the artificial event was not generated in SwingMouseHandler.mouseMoved() it is compensated by native MouseExited from SwingMouseHandler.mouseExited().

**#88 - 01/10/2018 01:15 AM - Hynek Cihlar**

Code review 3381a/11219.

Exiting the highlighted item now looks solid. Also I don't see anything wrong with the code changes, though the swing mouse handler will definitely require some regression testing.

However I still see some runtime issues.

1. When menu bar is hovered and then mouse-exited the highlight should go away, but it stays on. Can be replicated in both Swing and Web.

2. ESC causes the client to terminate instead of closing the menu. To replicate, click on the first menu bar item to open the submenu body, press ESC, the client terminates.

**#89 - 01/10/2018 03:18 PM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> 1. When menu bar is hovered and then mouse-exited the highlight should go away, but it stays on. Can be replicated in both Swing and Web.

Yes, this is just a simple painting refresh issue, fixed.

> 2. ESC causes the client to terminate instead of closing the menu. To replicate, click on the first menu bar item to open the submenu body, press ESC, the client terminates.

And this is not so easy to fix. The problem is when the sub-menu is opening the focus is really become null, we have disabled focus request for opened sub-menu because initially the body opening without any selection. And because current focus is null the END-ERROR is not delivering to menu and we have application stop instead of menu closing. So we need the approach that makes the menu body focused but not having any item highlighted.

Looking for solution, taking into account possible highlight change by keyboard navigation.

Regarding to the regression testing. I do agree we need to make exhaustive set of changes to make sure the current changes are safe. Can you recommend the test cases list to do except menubar.p?

**#90 - 01/10/2018 06:25 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11220.

Fixed the 1 and 2 notes resolution. The idea for point 2 is the highlighted is not the same as focused. When the sub-menu is shown for the first time the body is focused but not highlighted.

And I still need a help to gather the testcases list to verify the branch for regressions. Hynek, can you recommend some test set?

**#91 - 01/11/2018 04:43 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Task branch 3381a has been updated for review to revision 11220.
>
> And I still need a help to gather the testcases list to verify the branch for regressions. Hynek, can you recommend some test set?

Eugenie, I would choose the ones that are more complex or test a specific runtime behavior, see my shortlist below. The problem here is that this area requires high degree of input interaction and "creativity". In general for the regression testing you should check the following: all for menubar and popup menus for both swing and web clients: paint issues, ui flow (do the menu bodies react the same way as in the native app?), mouse input, keyboard input, keyboard shortcuts, ESC handling, legacy event processing. Also the ChUI should be checked if the changes affect the ChUI logic.

```
in uast/menu:
alt.p
ambiguity.p
children.p
empty_labels.p
like_simple.p
menubar.p
menuevents.p
menu_mouse.p
menu_sizes.p
mnemonic.p
popup_ext.p
sm_mnemonic.p
submenus.p
togglebox.p
win_menu*.p
```

**#92 - 01/12/2018 05:40 PM - Eugenie Lyzenko**

Running the testcases I have not found the regressions.

And the ChUI code was not touched in this branch.

**#93 - 01/13/2018 07:20 AM - Greg Shah**

Hynek: Can we merge this to trunk?  Or should we merge this into 3435a?

**#94 - 01/15/2018 04:04 AM - Hynek Cihlar**

I found two more issue in the latest 3381a.

To reproduce 1:

- run menu/menubar.p
- open the menu and navigate by keyboard to sm/sm2/Toyota 2 and press right key. sm0 root menu item will highlight and the sm menu body will stay on forever.

To reproduce 2:

- run menu/menubar.p
- open the menu and navigate by keyboard to sm/sm2/Toyota 2. Notice that when the submenu is opened for the first time Toyota 2 is not highlighted, the highlight will start again after down key is pressed multiple times.

**#95 - 01/15/2018 04:28 AM - Hynek Cihlar**

*- File Peek 2018-01-15 10-27.gif added*

Some more issue in the latest 3381a.

To reproduce 1:

- in menu/menubar.p
- click on a root menu item
- click into another fwd window, the menu will go away as expected
- however when switched back to the window the menu item previously highlighted will come on as highlighted again. The expected behavior is for the menu to stay un-highlighted.

To reproduce 2:

- in menu/menubar.p
- Open a submenu body sm and highlight sm2
- Move quickly out of the sm submenu body, the menu will stay on and will not go away even after switched to another window, see the attached gif showing this in the web client.

To reproduce 3:

- The submenu highlight is remembered.
- in menu/menubar.p click on sm/sm2/Toyota 2
- navigate to sm/sm2 and notice Toyota 2 is highlighted even though the mouse cursor is not hovering over it.

**#96 - 01/15/2018 04:33 AM - Hynek Cihlar**

Greg Shah wrote:

> Hynek: Can we merge this to trunk?  Or should we merge this into 3435a?

Yes I think it can be eventually merged directly to trunk.

**#97 - 01/15/2018 06:56 AM - Eugenie Lyzenko**

Hynek,

Are the problem in notes 94 and 95 the regressions from branch changes or new found issues?

**#98 - 01/15/2018 08:12 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Hynek,
>
> Are the problem in notes 94 and 95 the regressions from branch changes or new found issues?

I think they are from branch changes, but I didn't compare with trunk.

**#99 - 01/15/2018 12:41 PM - Eugenie Lyzenko**

Interesting finding for original 4GL behavior. On 4GL capable Windows system.

When navigating menu via keyboard if submenu is opened for left-right keys in menubar the first item of the submenu body is highlighted, this is not happening when opening submenu via mouse click. So we have to consider this while implementing/testing.

**#100 - 01/15/2018 10:24 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11221.

The update is a starting point for keyboard navigation issues. Some of them a regressions, I hope fixed, some are the missing functionality in original implementation. Must say the keyboard navigation has many options to handle, so it takes more time than I have expected. Some keys handling are not obvious, like reaction to right key when submenu item is highlighted or left-right key when menu-bar item is highlighted. So this change is not the last and debugging code is still here.

Continue testing and fixing.

**#101 - 01/16/2018 04:19 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11222.

Added changes to fix some GUI issues/regression for menu-bar or sub-menus. As to me the current behaviour is a good stable state.

Unfortunately it is not so easy to predict any possible scenario to say if the current code is 100% perfect as we want it to be. One important point I have discovered is when using/handling keyboard the mouse events are suspended and vice-versa. This means the menu does not handle keyboard and mouse simultaneously (or at least not all mouse/keyboard events). This means having standalone event queue for menu and implementation can take a time we currently do not have for this task. Or may be we can do this development now?

I would suggest to commit this change to be able to use fixes implemented. Then we can wok on more implementation for menu depending on time schedule and general task requirements, may be we we need to even rework menu handling model to make it more able to be fixed or improved. What do you think?

**#102 - 01/17/2018 08:47 AM - Greg Shah**

I haven't done a full code review but the TC changes are fine to merge to trunk.

Hynek: Please do a code review. If this is significantly better than trunk for menuing without creating any regressions that make it unusable, then I think merging to trunk is a good idea.

In regard to the next steps, Eugenie: please create a new task that describes the remaining issues in enough detail that we can pick up the work later. It may need posting some screencasts to properly demonstrate the issues.

My sense is it is probably time to focus on other changes and come back to this.

**#103 - 01/17/2018 09:29 AM - Hynek Cihlar**

Code review 3381a/11222. The code changes look OK to me.

I also did a quick retest and found:

(1) Note 94 point 1 is still an issue.
(2) When a submenu is navigated by keyboard and then another (FWD) window activated, the opened menu will stay on. When a non-FWD window is activated the opened menu will obscure the new focused window.
(3) A root menu-bar item keeps highlighted when focus switched from the window with menu to another window and back.

I think (1) and (2) should be cleared before merging to trunk as these are easy to replicate by the end-user and can negatively impact the workflow (i.e. by obscuring the window content).

**#104 - 01/17/2018 10:31 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Code review 3381a/11222. The code changes look OK to me.
>
> I also did a quick retest and found:
>
> (1) Note 94 point 1 is still an issue.
> (2) When a submenu is navigated by keyboard and then another (FWD) window activated, the opened menu will stay on. When a non-FWD window is activated the opened menu will obscure the new focused window.
> (3) A root menu-bar item keeps highlighted when focus switched from the window with menu to another window and back.
>
> I think (1) and (2) should be cleared before merging to trunk as these are easy to replicate by the end-user and can negatively impact the workflow (i.e. by obscuring the window content).

OK. Please provide demo for (2) and (3) cases above or more detailed instructions or specific *.p test file/code. I can not reproduce them.

**#105 - 01/17/2018 11:40 AM - Hynek Cihlar**

*- File Peek 2018-01-17 17-40.gif added*

Eugenie Lyzenko wrote:

> OK. Please provide demo for (2) and (3) cases above or more detailed instructions or specific *.p test file/code. I can not reproduce them.

See the attached gif running menu/menubar.p.

**#106 - 01/17/2018 11:52 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
>> OK. Please provide demo for (2) and (3) cases above or more detailed instructions or specific *.p test file/code. I can not reproduce them.
>
> See the attached gif running menu/menubar.p.

Great, thanks. But how did you get two FWD windows with menu/menubar.p testcase?

**#107 - 01/17/2018 11:56 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Great, thanks. But how did you get two FWD windows with menu/menubar.p testcase?

Make sure to update your uast working copy.

**#108 - 01/17/2018 11:58 AM - Hynek Cihlar**

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
> > Great, thanks. But how did you get two FWD windows with menu/menubar.p testcase?
>
> Make sure to update your uast working copy.

Sorry, I apparently failed to commit my changes in menubar.p. I am just checking them in.

**#109 - 01/17/2018 09:56 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11223.

This is the fixes for point (1) and (2). Working on the fix for point (3). The issue is related to window activation/deactivation handling with menubar wrong cleanup. So hope to have a fix in a day. I think it is better to fix all issues we now see if possible in a reasonable time.

**#110 - 01/18/2018 03:49 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11224.

This is the fix for (3) issue with root menu bar item wrong highlight. All required flags are now returning to the original state when window become inactive. Also new interface MenuGuiElement has been introduced to accumulate GUI specific methods for possible further refactoring and improvements. For now it has the method to reset enable highlight flag.

Hynek,

please let me know if you see some issues to fix with this update. The items 1-3 from note 103 should be resolved.

I'm performing additional testing on my side to verify the changes are safe.

**#111 - 01/18/2018 05:10 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11225.

This is the small change to separate GUI menu specific methods used in this branch with new interface for menu GUI items.

**#112 - 01/18/2018 07:08 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11226.

More fixes for menu handling issues with de-activation. One for reset keyboard navigation flag and another - with reset sub-menu activation blocker.

**#113 - 01/18/2018 08:55 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11227.

Final testing discovered new regression for mouse press on already opened submenu item. This is the fix. So for now I can consider the branch as stable enough.

**#114 - 01/19/2018 06:42 AM - Hynek Cihlar**

Code review 3381a/11227.

In ThinClient.processEventsWorker()

```
            if (ki.actionCode() == Keyboard.KA_END_ERROR && ki.keyCode() == Key.VK_ESCAPE &&
                ((src instanceof SubMenuGuiImpl || src instanceof MenuItemGuiImpl) &&
                 ((MenuGuiElement)src).acceptEndError()))
```

should be changed to

```
            if (ki.actionCode() == Keyboard.KA_END_ERROR && ki.keyCode() == Key.VK_ESCAPE &&
                ((src instanceof MenuGuiElement) &&
                 ((MenuGuiElement)src).acceptEndError()))
```

SubMenuGuiImpl.wndBody should be private

If I interpret the logic related to MenuGuiImpl.keyNavigation correctly, it is used for processing of the overlay window deactivation event to prevent submenu bodies to close in certain cases. Can you describe these "certain" cases and document it in the code? Since it is not obvious form the source I am afraid it could easily break with future changes.

Here are the runtime issues I found:
1. Two menu items can be highlighted at a time

- In menu/menubar.p
- Navigate to sm/Dodge, press s key. Notice two items get highlighted.

2. Menu item highlight is remembered when submenu is opened

- In menu/menubar.p
- Navigate by keys to sm/sm2/Dodge 2, Esc to close all submenus.
- Now navigate to sm/sm2 by mouse and notice the sm2 submenu has already an item highlighted.

3. Menu bodies stay on when switched to another top-level window

- In menu/menubar.p in Swing client
- Navigate to sm/sm2 by mouse
- Alt-tab to another window (FWD or non-FWD)
- The menu will stay on covering content of the newly focused window.

4. Wrong focus after menu closed

- In menu/menubar.p in Swing client
- Open submenu sm
- Close it by pressing Esc
- Focus is not transferred back to the original widget when the menu was opened

I think number 3 should be fixed before merging to trunk. 1, 2 and 4 can be postponed till later.

**#115 - 01/19/2018 11:01 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Code review 3381a/11227.
>
> In ThinClient.processEventsWorker()
> [...]
> should be changed to
> [...]
>
> SubMenuGuiImpl.wndBody should be private
>
> If I interpret the logic related to MenuGuiImpl.keyNavigation correctly, it is used for processing of the overlay window deactivation event to prevent submenu bodies to close in certain cases. Can you describe these "certain" cases and document it in the code? Since it is not obvious form the source I am afraid it could easily break with future changes.
>
> Here are the runtime issues I found:
> 1. Two menu items can be highlighted at a time
>
> - In menu/menubar.p
> - Navigate to sm/Dodge, press s key. Notice two items get highlighted.
>
> 2. Menu item highlight is remembered when submenu is opened
>
> - In menu/menubar.p
> - Navigate by keys to sm/sm2/Dodge 2, Esc to close all submenus.
> - Now navigate to sm/sm2 by mouse and notice the sm2 submenu has already an item highlighted.
>
> 3. Menu bodies stay on when switched to another top-level window
>
> - In menu/menubar.p in Swing client
> - Navigate to sm/sm2 by mouse
> - Alt-tab to another window (FWD or non-FWD)
> - The menu will stay on covering content of the newly focused window.
>
> 4. Wrong focus after menu closed
>
> - In menu/menubar.p in Swing client
> - Open submenu sm
> - Close it by pressing Esc
> - Focus is not transferred back to the original widget when the menu was opened
>
> I think number 3 should be fixed before merging to trunk. 1, 2 and 4 can be postponed till later.

Task branch 3381a has been updated for review to revision 11228.

I have added some documentation to usage of the activation control method when it is called. It is not directly related to MenuGuiImpl.keyNavigation but the MenuGuiImpl.keyNavigation value can be taken into account. Let me know if comments are not enough to have understanding.

The fix for case (3) is also included. Confirm if it works or not. There are the cases when it will not work I guess. If the FWD is not receiving any events from ALT-TAB - I think we need to make more investigation for how to handle this case. We can not react if nothing is happening inside FWD for external actions like ALT-TAB.

Continue working on rest issues.

**#116 - 01/19/2018 12:55 PM - Hynek Cihlar**

*- File Peek 2018-01-19 18-54.gif added*

Eugenie Lyzenko wrote:

> I have added some documentation to usage of the activation control method when it is called. It is not directly related to MenuGuiImpl.keyNavigation but the MenuGuiImpl.keyNavigation value can be taken into account. Let me know if comments are not enough to have understanding.

Yes, I think the comments helped to explain the case.

> The fix for case (3) is also included. Confirm if it works or not. There are the cases when it will not work I guess. If the FWD is not receiving any events from ALT-TAB - I think we need to make more investigation for how to handle this case. We can not react if nothing is happening inside FWD for external actions like ALT-TAB.

It is still an issue, see the attached gif. The Alt-Tab will cause the top-level window to receive a WindowActivated event. We should focus on handling this activation event and not care how we got it - there are multiple ways how the window can get deactivated.

**#117 - 01/19/2018 07:17 PM - Eugenie Lyzenko**

Task branch 3381a has been updated for review to revision 11229.

This is the fix for 1-3 notes. The case (3) must now be fixed, please confirm. The note (4) can really be postponed because can require serious focus handling subsystem investigation. Just because it can take a time to keep track for correct focus route in your scenario.

**#118 - 01/22/2018 09:19 AM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Task branch 3381a has been updated for review to revision 11229.

Code review 3381a/11229.

The code changes seem to be OK. I found a runtime issue though causing the client to abend.

To replicate:

- run menu/menubar.p
- navigate to sm/sm2 by mouse (do not enter the submenu sm2)
- press right arrow key
- note that the submenu sm2 isn't entered, instead sm0 submenu is opened

- navigate to sm/sm2 by mouse, notice the submenu body will not open
- navigate to sm/sm3 by mouse and click on Toyota 3, the client abends

**#119 - 01/22/2018 10:22 AM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Eugenie Lyzenko wrote:
>
> > Task branch 3381a has been updated for review to revision 11229.
>
> Code review 3381a/11229.
>
> The code changes seem to be OK. I found a runtime issue though causing the client to abend.
>
> To replicate:
>
> - run menu/menubar.p
> - navigate to sm/sm2 by mouse (do not enter the submenu sm2)
> - press right arrow key
> - note that the submenu sm2 isn't entered, instead sm0 submenu is opened

The same as 4gl recation.

- navigate to sm/sm2 by mouse, notice the submenu body will not open
- navigate to sm/sm3 by mouse and click on Toyota 3, the client abends

Task branch 3381a has been updated for review to revision 11230.

Fixed abend with menu selection.

**#120 - 01/22/2018 12:31 PM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Task branch 3381a has been updated for review to revision 11230.
>
> Fixed abend with menu selection.

Code review 3381a/11230. The changes are ok. The abend is resolved. I found the following exception during retest, but I am not sure how to replicate it. I only remember switching rapidly the root menu bar items in menu/menubar.p with keyboard left and right arrows.

```
java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:124)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:757)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:412)
    at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:348)
    at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1170)
    at com.goldencode.p2j.net.Queue.transact(Queue.java:641)
    at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
    at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
    at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
    at com.sun.proxy.$Proxy4.standardEntry(Unknown Source)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:361)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:158)
    at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:250)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
    at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
    at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at com.intellij.uiDesigner.snapShooter.SnapShooter.main(SnapShooter.java:59)
Caused by: java.lang.NullPointerException
    at com.goldencode.p2j.ui.client.WindowManager.resolveWindow(WindowManager.java:893)
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl.getLongestItemAccelWidth(SubMenuGuiImpl.java:261)
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl.titleWidth(SubMenuGuiImpl.java:806)
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl.getMaxSubMenuItemWidth(SubMenuGuiImpl.java:426)
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl.bodyDimension(SubMenuGuiImpl.java:1446)
    at com.goldencode.p2j.ui.client.gui.SubMenuGuiImpl.findMouseSource(SubMenuGuiImpl.java:1177)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.findMouseSource(AbstractContainer.java:450)
    at com.goldencode.p2j.ui.client.gui.MenuGuiImpl.findMouseSource(MenuGuiImpl.java:727)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.findMouseSource(AbstractContainer.java:450)
    at com.goldencode.p2j.ui.client.widget.AbstractContainer.findMouseSource(AbstractContainer.java:450)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.findMouseSource(WindowGuiImpl.java:785)
    at com.goldencode.p2j.ui.client.gui.driver.swing.SwingMouseHandler.findMouseSource(SwingMouseHandler.java:
494)
    at com.goldencode.p2j.ui.client.gui.driver.swing.SwingMouseHandler.lambda$mouseMoved$8(SwingMouseHandler.j
ava:360)
    at com.goldencode.p2j.ui.chui.ThinClient.waitForEvent(ThinClient.java:14424)
    at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:12199)
    at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:11774)
    at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:11717)
    at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:11671)
```

Other than the above I don't see any other issue that prevents 3381a to be merged to trunk.

**#121 - 01/22/2018 01:25 PM - Eugenie Lyzenko**

Hynek Cihlar wrote:

> Code review 3381a/11230. The changes are ok. The abend is resolved. I found the following exception during retest, but I am not sure how to replicate it. I only remember switching rapidly the root menu bar items in menu/menubar.p with keyboard left and right arrows.
>
> [...]
>
> Other than the above I don't see any other issue that prevents 3381a to be merged to trunk.

Task branch 3381a has been updated for review to revision 11231.

This is the fix for menu navigation NPE. I have not reproduced the issue but the change must avoid using null widget.

**#122 - 01/22/2018 02:04 PM - Hynek Cihlar**

Eugenie Lyzenko wrote:

> Task branch 3381a has been updated for review to revision 11231.
>
> This is the fix for menu navigation NPE. I have not reproduced the issue but the change must avoid using null widget.

The change is OK.

**#123 - 01/22/2018 02:15 PM - Greg Shah**

Eugenie: please go forward with the following:

1. Merge 3381a to trunk.
2. Open a new task for the remaining menuing bugs.  Make that task "related" to this one and add both Hynek and myself as watchers.

**#124 - 01/22/2018 02:35 PM - Eugenie Lyzenko**

Greg Shah wrote:

> Eugenie: please go forward with the following:
>
> 1. Merge 3381a to trunk.
> 2. Open a new task for the remaining menuing bugs.  Make that task "related" to this one and add both Hynek and myself as watchers.

OK. Starting to merge.

**#125 - 01/22/2018 02:41 PM - Eugenie Lyzenko**

Task branch 3381a has been merged into trunk as revision 11215. The branch 3381a has been archived.

**#126 - 01/22/2018 04:37 PM - Eugenie Lyzenko**

*- Related to Bug #3461: Remaining menu/sub-menu bugs or incorrect behavior added*

**#127 - 01/22/2018 05:15 PM - Greg Shah**

*- Start date deleted (11/22/2017)*

*- Status changed from New to Closed*

*- Assignee set to Eugenie Lyzenko*

*- % Done changed from 0 to 100*

## Files

| | | | |
|---|---|---|---|
| samples.png | 27.4 KB | 11/22/2017 | Stanislav Lomany |
| start.p | 113 Bytes | 11/22/2017 | Stanislav Lomany |
| sub-menus.p | 528 Bytes | 11/22/2017 | Stanislav Lomany |
| sub_menu1_level1.jpg | 16.7 KB | 12/09/2017 | Eugenie Lyzenko |
| sub_menu1_level2.jpg | 23.7 KB | 12/09/2017 | Eugenie Lyzenko |
| missing_vert_win10.jpg | 26.6 KB | 12/11/2017 | Eugenie Lyzenko |
| sub_menu1_fwd_20171211a.jpg | 36.7 KB | 12/12/2017 | Eugenie Lyzenko |
| sub_menu1_fwd_20171211b.jpg | 19.1 KB | 12/12/2017 | Eugenie Lyzenko |
| Peek 2017-12-14 15-12.gif | 122 KB | 12/14/2017 | Hynek Cihlar |
| menubar_fwd_issue_20171215a.jpg | 16.1 KB | 12/15/2017 | Eugenie Lyzenko |
| menubar_after_ESC_4gl.jpg | 21.5 KB | 12/17/2017 | Eugenie Lyzenko |
| menubar_after_ESC_fwd_20171217a.jpg | 35.3 KB | 12/17/2017 | Eugenie Lyzenko |
| sub_menu_classic_pop-up.jpg | 54.2 KB | 12/20/2017 | Eugenie Lyzenko |
| sub_menu_classic_menu-bar.jpg | 46.1 KB | 12/20/2017 | Eugenie Lyzenko |
| sub_menu_classic_fwd_pop-up_20171219.jpg | 19.4 KB | 12/20/2017 | Eugenie Lyzenko |
| sub_menu_classic_fwd_menu-bar_20171219.jpg | 17.5 KB | 12/20/2017 | Eugenie Lyzenko |
| Peek 2017-12-20 09-04.gif | 65.4 KB | 12/20/2017 | Hynek Cihlar |
| Peek 2017-12-21 22-04.gif | 251 KB | 12/21/2017 | Hynek Cihlar |
| menubar_opened_submenu_4gl.jpg | 19.9 KB | 12/21/2017 | Eugenie Lyzenko |
| menubar_opened_submenu_fwd_20171221a.jpg | 10.1 KB | 12/21/2017 | Eugenie Lyzenko |
| Peek 2017-12-28 18-12.gif | 55.5 KB | 12/28/2017 | Hynek Cihlar |
| Peek 2017-12-28 18-22.gif | 79.9 KB | 12/28/2017 | Hynek Cihlar |
| Peek 2017-12-28 18-26.gif | 207 KB | 12/28/2017 | Hynek Cihlar |
| Peek 2018-01-15 10-27.gif | 300 KB | 01/15/2018 | Hynek Cihlar |
| Peek 2018-01-17 17-40.gif | 189 KB | 01/17/2018 | Hynek Cihlar |
| Peek 2018-01-19 18-54.gif | 247 KB | 01/19/2018 | Hynek Cihlar |