

User Interface - Bug #3382

Client-side popup menus abend

11/22/2017 09:18 AM - Stanislav Lomany

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Hynek Cihlar	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to User Interface - Feature #3261: enhanced browse that can optionall...			Closed

History

#1 - 11/22/2017 09:41 AM - Stanislav Lomany

The problem with popup menus created at client side is that they can cause abend under a certain circumstances, e.g. if a message box was displayed. Reproduction:

1. In ScrollPopupGuiImpl.create replace the code which creates "Scroll Here" item with this one, which displays a message when we press "Scroll Here":

```
// Creating items

MenuItemGuiImpl scrollHereOpt = MenuItemGuiImpl.createDynamicItem("Scroll Here", true,
    new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        ThinClient.getInstance().messageBox(new Object[]{"test"}, null,
            LogicalTerminal.ALERT_MESSAGE,
            LogicalTerminal.BTN_OK,
            " Error ",
            null,
            WindowConfig.RESOLVE_WINDOW,
            null);
    }
});
```

2. Run a browse testcase. E.g.:

```
DEF TEMP-TABLE tt FIELD f1 AS INTEGER
                FIELD f2 AS character format "x(20)"
                FIELD f3 AS LOGICAL.

def var i as integer.
repeat i = 1 to 100:
    CREATE tt. tt.f1 = i.
end.

DEFINE QUERY q FOR tt SCROLLING.
OPEN QUERY q FOR EACH tt.

DEF BROWSE br QUERY q
DISPLAY
    tt.f1
    tt.f2
    tt.f3 VIEW-AS TOGGLE-BOX
```

```
WITH size-chars 50 by 10 TITLE "Static browse" separators.
```

```
DEF FRAME fr br SKIP(1)  
WITH TITLE "Frame" SIZE 70 BY 15 NO-LABELS.
```

```
ENABLE ALL WITH FRAME fr.
```

```
WAIT-FOR WINDOW-CLOSE OF DEFAULT-WINDOW.
```

Press right button on the right browse scrollbar to show the popup menu, select "Scroll here", click "OK" on the message box.
3. You will get the abend because the menu item (actually the parent menu) is not attached to a window:

```
Caused by: java.lang.IllegalStateException: Widget -51 is not attached to a TopLevelWindow instance.  
  at com.goldencode.p2j.ui.client.widget.AbstractWidget.topLevelWindow(AbstractWidget.java:385)  
  at com.goldencode.p2j.ui.chui.ThinClient.invokeTriggers(ThinClient.java:18238)  
  at com.goldencode.p2j.ui.chui.ThinClient.invokeTriggers(ThinClient.java:18100)  
  at com.goldencode.p2j.ui.chui.ThinClient.processProgressEvent(ThinClient.java:17416)  
  at com.goldencode.p2j.ui.chui.ThinClient.processEventsWorker(ThinClient.java:16481)  
  at com.goldencode.p2j.ui.chui.ThinClient.pop(ThinClient.java:15453)  
  at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:15436)  
  at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:15360)  
  at com.goldencode.p2j.ui.chui.ThinClient.applyWorker(ThinClient.java:15121)  
  at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:12193)  
  at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:11677)  
  at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:11620)  
  at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:11574)  
  ... 21 more
```

#2 - 11/23/2017 11:45 AM - Hynek Cihlar

- Assignee set to Hynek Cihlar

#3 - 11/23/2017 11:45 AM - Hynek Cihlar

- Status changed from New to WIP

#4 - 11/23/2017 12:01 PM - Hynek Cihlar

Stanislav, do you remember the actual case when the exception happened? I would rather have a real recreate for this.

#5 - 11/23/2017 12:09 PM - Stanislav Lomany

I've created a client-side popup menu for browse (using the same approach used as in ScrollPopupMenuImpl) and called TC.chooseFile.

#6 - 12/01/2017 04:38 PM - Eugenie Lyzenko

The root cause for this issue is the code in pop up menu item:

```
public TopLevelWindow<O> topLevelWindow()
{
    @SuppressWarnings("unchecked")
    TopLevelWindow<O> wnd = parentOrSelf(TopLevelWindow.class);

    if (!(wnd instanceof TopLevelWindow))
    {
        int wid = UiUtils.getWidgetIdAsInt(this);
        throw new IllegalStateException("Widget " + name() + " " + wid
            + " is not attached to a TopLevelWindow instance.");
    }
}
```

In this code `parentOrSelf(TopLevelWindow.class)` gives null for parent widget. Because some time before this call the popup menu is detaching and hiding. And this means there is no parent window. So instead of exception generation I would suggest to get the default window for this case:

```
public TopLevelWindow<O> topLevelWindow()
{
    @SuppressWarnings("unchecked")
    TopLevelWindow<O> wnd = parentOrSelf(TopLevelWindow.class);

    // sometimes the widget (like popup menu) has been detached at this time
    // so we need to get the default window instead
    if (wnd == null)
    {
        wnd = WindowManager.getDefaultWindow();
    }

    if (!(wnd instanceof TopLevelWindow))
    {
        int wid = UiUtils.getWidgetIdAsInt(this);
        throw new IllegalStateException("Widget " + name() + " " + wid
            + " is not attached to a TopLevelWindow instance.");
    }
}
```

This change resolve the issue. Any objections or notes?

#7 - 12/01/2017 04:56 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

This change resolve the issue. Any objections or notes?

The exception thrown from `topLevelWindow()` is there for a purpose. The method expects the widget to be connected to a top-level window, the exception indicates that the code has a problem and must be resolved.

If in the menu-item it is expected that the top-level window may be gone, use `parentOrSelf()` instead of `topLevelWindow()`.

#8 - 12/01/2017 05:46 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

Eugenie Lyzenko wrote:

If in the menu-item it is expected that the top-level window may be gone, use `parentOrSelf()` instead of `topLevelWindow()`.

I do not understand what you mean.

The call `pupUpMenu.topLevelWindow()` executes from `ThinClient.invokeTriggers()`, line 18245 at the time when pop up menu already detached from parent, so `parentOrSelf()` will also return null. We could change `ThinClient.invokeTriggers()` from:

```
WidgetId id = widget.getId();
if (widget != null    &&
    widget.parent() != null &&
    WidgetId.virtualWidget(id))
{
    // all keys from virtual sources (still attached to a window) are sent directly to the
    // associated window...
    widget = widget.topLevelWindow();
}
```

to:

```
WidgetId id = widget.getId();
if (widget != null    &&
    widget.parent() != null &&
    WidgetId.virtualWidget(id))
{
    // all keys from virtual sources (still attached to a window) are sent directly to the
    // associated window...
    Widget widParent = widget.parentOrSelf(TopLevelWindow.class);
    if (widParent != null)
    {
        widget = widParent;
    }
}
```

This also eliminates the exception for widgets that detached from parent, like pop-up menus.

#9 - 12/02/2017 11:44 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

This also eliminates the exception for widgets that detached from parent, like pop-up menus.

Yes, this is what I meant. To use `parentOrSelf()` and check the result for null, however only when the state of the widget allows the parent to be null. That is in your particular case: (1) is the trigger expected for the menu-item? (2) If it is, how come the widget is detached from its window, is this correct? Should it stay connected at this point?

#10 - 12/04/2017 06:36 PM - Eugenie Lyzenko

Debugging result. The current events data flow:

1. After the user clicks on Scroll here pop up menu from scrollbar the `MenuItemGuiImpl.mouseClicked()` runs `mouseClicked` registered with `ScrollPopupGuiImpl` which causes the message box modal execution request.
2. Before modal message box showing the pop-up menu is hiding and detaching from parent overlay window.
3. After user press OK button in message box the control is returning to the pop-up menu `ScrollPopupGuiImpl.mouseClicked()`. The continue processing cause the `MenuItemGuiImpl.mouseClicked()` to continue with `MenuItemGuiImpl.selectMenuItem()`.
4. This in turn causes the CHOOSE event generated for `ScrollPopupGuiImpl`.
5. This CHOOSE event for `ScrollPopupGuiImpl` is handled by `ThinClient` event processing. And here we have got an issue while trying to detect the parent for `ScrollPopupGuiImpl` which has been hidden and detached on step 2 above.

So the CHOOSE event either must be handled when pop-up is not hidden or we need to defer the pop-up detach until complete destruction or we can not use calls that affects pop-up visibility inside mouse handler for pop-up. Or we have to generate pop-up menu CHOOSE event before finishing mouse click, say at processing of mouse press event coming.

#11 - 12/04/2017 09:35 PM - Eugenie Lyzenko

So the suggested fix approach can be the following change in `ThinClient.invokeTriggers(...)`, line starting at 18225:

```
WidgetId id = widget.getId();
if (widget != null      &&
    widget.parent() != null &&
    WidgetId.virtualWidget(id) &&
    (widget.parentOrSelf(TopLevelWindow.class) != null ||
     !(widget instanceof MenuItemGuiImpl)))
{
    // all keys from virtual sources (still attached to a window) are sent directly to the
    // associated window...
    widget = widget.topLevelWindow();
}
```

In this change we allow the widget to be detached (`widget.parentOrSelf(TopLevelWindow.class) == null`) if the widget is pop-up menu item. This allows the menu item select event to be delivered to respective menu item even if the menu has already been detached. Any comments, objections?

#12 - 12/05/2017 04:29 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

So the suggested fix approach can be the following change in `ThinClient.invokeTriggers(...)`, line starting at 18225:

[...]

In this change we allow the widget to be detached (`widget.parentOrSelf(TopLevelWindow.class) == null`) if the widget is pop-up menu item. This allows the menu item select event to be delivered to respective menu item even if the menu has already been detached. Any comments, objections?

Eugenie, I would rather not introduce the type cast in `ThinClient`. The mouse event being handled with the widget unattached seems to be a side-effect of something else. I think the core cause is the order of execution of the mouse handlers in the menu item widget. The `MouseAdapter` responsible for showing the modal window should be executed after `selectMenuItem()`.

Please try to move the execution of mouse adapters in `MenuItemGuiImpl.mouseClicked` before `selectMenuItem()`. If there are mouse adapters that rely on the current execution order, we can have two sets of mouse adapters, one set executed before the default handler, and the second set executed after.

#13 - 12/05/2017 05:55 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

Please try to move the execution of mouse adapters in `MenuItemGuiImpl.mouseClicked` before `selectMenuItem()`.

You mean execution `selectMenuItem()` first, then mouse adapters? Asking because currently the `MenuItemGuiImpl.mouseClicked` implements the order:

1. Mouse adapters.
2. `selectMenuItem()`.

My concern is the `selectMenuItem()` is the mandatory call for menu item, while mouse adapters are optional, so I think `selectMenuItem()` must be the first thing to do while handling click.

#14 - 12/05/2017 02:34 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

You mean execution `selectMenuItem()` first, then mouse adapters?

Exactly so.

#15 - 12/05/2017 09:49 PM - Eugenie Lyzenko

Unfortunately this is not enough to fix the issue. Moving `MenuItemGuiImpl.selectMenuItem()` before `ScrollPopupGuiImpl.mouseClicked()` does not change the issue condition if `ScrollPopupGuiImpl.mouseClicked()` runs `ThinClient.messageBox()`. The respective CHOOSE event is properly posted before message box but `ThinClient` does not have the ability to peek and process this message because `ThinClient.messageBox()` blocks it entering it's own message loop, and first hiding the pop-up menu (and detaching it from overlay window).

So we need another solution. Either we have to defer pop-up detaching until all related messages be processed or may be `MenuItemGuiImpl.selectMenuItem()` should directly run `ThinClient.processProgressMessage()` to ensure immediate CHOOSE event processing. I'm inclining to detaching defer.

#16 - 12/06/2017 12:04 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

Unfortunately this is not enough to fix the issue. Moving `MenuItemGuiImpl.selectMenuItem()` before `ScrollPopupGuiImpl.mouseClicked()` does not change the issue condition if `ScrollPopupGuiImpl.mouseClicked()` runs `ThinClient.messageBox()`.

The call to `ThinClient.messageBox()` must be executed in `ThinClient.invokeLater` so that the message box is serialized with the other events posted on the event queue.

#17 - 12/06/2017 12:20 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

The call to `ThinClient.messageBox()` must be executed in `ThinClient.invokeLater` so that the message box is serialized with the other events posted on the event queue.

What I do not understand here is why we are committed to `ThinClient.messageBox()`? This is not the real code to be left in FWD. Another word you mean every possible code (we will remove `ThinClient.messageBox()` from final code) processing in `ScrollPopupGuiImpl.mouseClicked()` must be executed inside `ThinClient.invokeLater()`. Correct?

#18 - 12/06/2017 01:31 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

What I do not understand here is why we are committed to `ThinClient.messageBox()`? This is not the real code to be left in FWD.

The message box example is a pattern that will be needed for displaying some of the system dialogs.

Another word you mean every possible code (we will remove `ThinClient.messageBox()` from final code) processing in `ScrollPopupGuiImpl.mouseClicked()` must be executed inside `ThinClient.invokeLater()`. Correct?

Correct for any code that expects the dispatch of the enqueued events in the order they were enqueued.

#19 - 12/06/2017 01:55 PM - Hynek Cihlar

Hynek Cihlar wrote:

Another word you mean every possible code (we will remove `ThinClient.messageBox()` from final code) processing in `ScrollPopupGuiImpl.mouseClicked()` must be executed inside `ThinClient.invokeLater()`. Correct?

Correct for any code that expects the dispatch of the enqueued events in the order they were enqueued.

In fact I don't see any reason the ordering should not be enforced for all mouse adapters in `MenuItemGuiImpl`. So instead of wrapping the method body of `MouseAdapter.mouseClicked()` with `ThinClient.invokeLater()`, the call itself to `MA.mouseClicked()` in `MIGI.mouseClicked()` should be wrapped.

#20 - 12/06/2017 03:32 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

Hynek Cihlar wrote:

In fact I don't see any reason the ordering should not be enforced for all mouse adapters in `MenuItemGuiImpl`. So instead of wrapping the method body of `MouseAdapter.mouseClicked()` with `ThinClient.invokeLater()`, the call itself to `MA.mouseClicked()` in `MIGI.mouseClicked()` should

be wrapped.

Actually the situation is a bit more complex. This is very event order sensitive area so we have restore 4GL processing. In windows the order is following for pop-up menu selection:

1. Mouse press
2. Mouse release
3. Mouse click
4. Menu item selection
5. Menu dismiss
6. Execute action for menu item if registered

If we change the some points in this order the issues start. For example when we display message box on click, step 3 we dismiss menu before item selection processing where the original trap is happening.

But if we postpone click processing to the end - we have dismissed menu when it should exist as in the original ScrollPopupGuiImpl.mouseClicked() handler. The problematic code there become:

```
NativePoint popupPos = menu.screenPhysicalLocation();
```

In this case we have null parent again because invokeLater() defer click handler to the moment when menu has been detached.

So we need something more smart than hacking with invokeLater() call. And first we need to preserve original event ordering. May be we must change action body from click to item selection or distribute processing between events, something like gathering info on press or click and do action on item select.

#21 - 12/06/2017 04:17 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

Hynek Cihlar wrote:

Hynek Cihlar wrote:

In fact I don't see any reason the ordering should not be enforced for all mouse adapters in MenuItemGuiImpl. So instead of wrapping the method body of MouseAdapter.mouseClicked() with ThinClient.invokeLater(), the call itself to MA.mouseClicked() in MIGI.mouseClicked() should be wrapped.

Actually the situation is a bit more complex. This is very event order sensitive area so we have restore 4GL processing. In windows the order is following for pop-up menu selection:

1. Mouse press
2. Mouse release
3. Mouse click
4. Menu item selection
5. Menu dismiss
6. Execute action for menu item if registered

If we change the some points in this order the issues start. For example when we display message box on click, step 3 we dismiss menu before item selection processing where the original trap is happening.

But if we postpone click processing to the end - we have dismissed menu when it should exist as in the original ScrollPopupGuiImpl.mouseClicked() handler. The problematic code there become:

[...]

In this case we have null parent again because invokeLater() defer click handler to the moment when menu has been detached.

I didn't mean to enqueue the body of MenuItemGuiImpl.mouseClicked() but rather only the call to MouseAdapter.mouseClicked().

#22 - 12/06/2017 04:27 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

But if we postpone click processing to the end - we have dismissed menu when it should exist as in the original ScrollPopupGuiImpl.mouseClicked() handler. The problematic code there become:

[...]

In this case we have null parent again because invokeLater() defer click handler to the moment when menu has been detached.

Btw. using the popup menu for determining the click offset in the scroll bar is incorrect. The offset must be calculated from the mouse event location and not from the popup location. What if the popup location is adjusted in order to be fully visible on the screen? For example when the click location is close to the screen boundary. In this case the popup location will be adjusted.

#23 - 12/06/2017 04:30 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

Eugenie Lyzenko wrote:

But if we postpone click processing to the end - we have dismissed menu when it should exist as in the original ScrollPopupGuiImpl.mouseClicked() handler. The problematic code there become:

[...]

In this case we have null parent again because `invokeLater()` defer click handler to the moment when menu has been detached.

Btw. using the popup menu for determining the click offset in the scroll bar is incorrect.

Yes, at least scrolling down by pop-up menu does not work.

#24 - 12/06/2017 04:36 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

Yes, at least scrolling down by pop-up menu does not work.

I think there are two options, (1) the correct one and the (2) quick one.

(1) Enqueue the call to mouse adapter on the event queue. This way the event ordering becomes as you mention in note 20. Unfortunately there is action code that relies on the menu item being attached to the menu and overlay window when it is executed. This code would have to be fixed, too.

(2) Enqueue only the call to `TC.messageBox()` in the mouse adapter body. Less impact on the existing code. But the expected ordering becomes less obvious.

#25 - 12/06/2017 04:39 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

In this case we have null parent again because `invokeLater()` defer click handler to the moment when menu has been detached.

I didn't mean to enqueue the body of `MenuItemGuiImpl.mouseClicked()` but rather only the call to `MouseAdapter.mouseClicked()`.

This is exactly how it is currently implemented. Pop-up `ScrollPopupGuiImpl.mouseClicked()` is called from `MenuItemGuiImpl.mouseClicked()` via `MouseAdapter` interface. So it does not matter whether we wrap `ScrollPopupGuiImpl.mouseClicked()` or call to `MouseAdapter.mouseClicked()`. The result will be the same - events ordering become inconsistent.

Eugenie Lyzenko wrote:

This is exactly how it is currently implemented. Pop-up ScrollPopupGuiImpl.mouseClicked() is called from MenuItemGuiImpl.mouseClicked() via MouseAdapter interface. So it does not matter whether we wrap ScrollPopupGuiImpl.mouseClicked() or call to MouseAdapter.mouseClicked(). The result will be the same - events ordering become inconsistent.

There is a difference.

```
MenuItemGuiImpl.mouseClicked(MouseEvent e)
{
    selectMenuItem();

    if (mouseListeners != null && mouseListeners.containsKey(MouseEvent.MOUSE_CLICKED))
    {
        for (MouseAdapter adapter : mouseListeners.get(MouseEvent.MOUSE_CLICKED))
        {
            adapter.mouseClicked(e);
        }
    }
}
```

The above will result in:

1. Menu body hidden.
2. SE_CHOOSSE enqueued.
3. Mouse adapter executed.
4. SE_CHOOSSE dispatched.

vs.

```
MenuItemGuiImpl.mouseClicked(MouseEvent e)
{
    selectMenuItem();

    if (mouseListeners != null && mouseListeners.containsKey(MouseEvent.MOUSE_CLICKED))
    {
        for (MouseAdapter adapter : mouseListeners.get(MouseEvent.MOUSE_CLICKED))
        {
            invokeLater(() -> adapter.mouseClicked(e));
        }
    }
}
```

The above will result in:

1. Menu body hidden.
2. SE_CHOOSSE enqueued.
3. Mouse adapter enqueued.
4. SE_CHOOSSE dispatched.
5. Mouse adapter executed.

#27 - 12/06/2017 04:57 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

(2) Enqueue only the call to `TC.messageBox()` in the mouse adapter body. Less impact on the existing code. But the expected ordering becomes less obvious.

Really I do not understand what `TC.messageBox()` do you mean. There is no `TC.messageBox()` in original code. How it is related to FWD scroll-bar pop-up? Can you provide the 4GL example? I have no objection, I just do not see from where this call is coming from.

#28 - 12/06/2017 05:05 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

Eugenie Lyzenko wrote:

This is exactly how it is currently implemented. Pop-up `ScrollPopupGuiImpl.mouseClicked()` is called from `MenuItemGuiImpl.mouseClicked()` via `MouseAdapter` interface. So it does not matter whether we wrap `ScrollPopupGuiImpl.mouseClicked()` or call to `MouseAdapter.mouseClicked()`. The result will be the same - events ordering become inconsistent.

There is a difference.

[...]

The above will result in:

1. Menu body hidden.
2. `SE_CHOOSE` enqueued.
3. Mouse adapter executed.
4. `SE_CHOOSE` dispatched.

vs.

[...]

The above will result in:

1. Menu body hidden.
2. `SE_CHOOSE` enqueued.
3. Mouse adapter enqueued.
4. `SE_CHOOSE` dispatched.
5. Mouse adapter executed.

If adapter `adapter.mouseClicked(e)` calls body that has `invokeLater` wrapper the result will be the same. Anyway the mouse adapter execution (where the menu processing happening) become active when the menu is hidden and detached and this causes the exception issue.

#29 - 12/06/2017 05:06 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

Hynek Cihlar wrote:

(2) Enqueue only the call to `TC.messageBox()` in the mouse adapter body. Less impact on the existing code. But the expected ordering becomes less obvious.

Really I do not understand what `TC.messageBox()` do you mean. There is no `TC.messageBox()` in original code. How it is related to FWD scroll-bar pop-up? Can you provide the 4GL example? I have no objection, I just do not see from where this call is coming from.

The problematic part is the new event queue established in the adapter, like when `TC.messageBox()` is called. This isn't in the trunk yet, but soon will be (file chooser dialog shown when a menu item selected for example).

#30 - 12/06/2017 05:12 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

If adapter `adapter.mouseClicked(e)` calls body that has `invokeLater` wrapper the result will be the same. Anyway the mouse adapter execution (where the menu processing happening) become active when the menu is hidden and detached and this causes the exception issue.

The point is that when the call to the adapters invocation in `MIGI.mouseClicked()` is not wrapped in `invokeLater()` one can use `invokeLater()` in the adapter bodies when needed. So it would be used only for the code that displays modal windows.

#31 - 12/06/2017 11:52 PM - Eugenie Lyzenko

Well, looks like I have found alternative approach. So the set of changes:

1. No using `invokeLater()` to avoid change event current order.

2. Keep processing order for `mouseClicked` in `MenuItemGuiImpl`

- call the mouse listeners first

- call the `selectMenuItem()` second

This way we preserve the correct event order for regular cases.

3. Add calling `processKeyListeners()` in `processKeyEvent` of the `MenuItemGuiImpl`:

...

```

super.processKeyEvent(event);
// check for attached listeners
processKeyListeners(event);
}

```

This is the way we will call execute handler for pop-up menu selection event.

3. Install KeyListener in ScrollPopupGuiImpl:

```

scrollHereOpt.addListener(new KeyListener()
{
    @Override
    public void onKeyPressed(KeyInput keyEvent)
    {
        // this is the handling for menu selection event
        if (keyEvent.actionCode() == Keyboard.SE_CHOOSE)
        {
            ThinClient.getInstance().messageBox(new Object[]{"test"}, null,
                LogicalTerminal.ALERT_MESSAGE,
                LogicalTerminal.BTN_OK,
                " Error ",
                null,
                WindowConfig.RESOLVE_WINDOW,
                null);
        }
    }
});

@Override
public void onKeyTyped(KeyInput keyEvent)
{
    // stub for key typed evens
}
});

```

So we change the moment when we call messageBox() to show after actual menu selection. And keep old processing within mouseClicked().

The main idea is: 1) when processing is not critical to pop-up menu event ordering and must be executed early enough - we can call the code within mouseClicked(), like we currently do for scrolling code. 2) when processing can mess the pop-up menu event processing or no matter when to call - we execute such code with menu item selection handler with onKeyPressed() or onKeyTyped(), like for the messageBox() case. The menu item must properly register KeyListener().

This way we keep the correct event ordering and execute all we want with pop-up menu of the scrollbar or similar pop-ups.

I have tested this approach and it works fine with message box. Any comments/objections.

Also planning to debug scrolling issues with pop-up "Scroll Here" processing.

#32 - 12/07/2017 04:00 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

Well, looks like I have found alternative approach. So the set of changes:

I like that. This should also fix and allow the menu items to be selected by the keyboard. When you are at the changes can you make sure the key input works correctly? I am getting an exception when trying to navigate the menu, see the exception below. Also please test the check box menu item with your changes.

```
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.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)
```

#33 - 12/07/2017 01:29 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

When you are at the changes can you make sure the key input works correctly? I am getting an exception when trying to navigate the menu, see the exception below. Also please test the check box menu item with your changes.

[...]

Task branch 3394a has been updated for review to revision 11244.

This is the fix for exceptions generated by dialogBox() like show. Also fixes the NullPointerException from key navigation with pop-up. As you can now see the pop-up menu items drawing is wrong I guess, the text color is not the same as one we have from navigating by mouse move.

Working on the scrolling issue.

#34 - 12/07/2017 08:10 PM - Eugenie Lyzenko

Task branch 3394a has been updated for review to revision 11247.

Fixed the bar scrolling initiated from pop-up menu. The pop-up initial location must be stored separately because due to showing the physical location is resetting to (0,0). So the change makes pop-up local copy of physical location for reuse. Also fixed the pop-up menu drawing when using keyboard navigation. Tested with both Classic and Windows10 themes.

#35 - 12/08/2017 02:57 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

Task branch 3394a has been updated for review to revision 11247.

Fixed the bar scrolling initiated from pop-up menu. The pop-up initial location must be stored separately because due to showing the physical location is resetting to (0,0). So the change makes pop-up local copy of physical location for reuse. Also fixed the pop-up menu drawing when using keyboard navigation. Tested with both Classic and Windows10 themes.

Eugenie, why not to get the location from the mouse event? What if the popup isn't actually displayed at the mouse cursor location?

#36 - 12/08/2017 04:47 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

Eugenie, why not to get the location from the mouse event? What if the popup isn't actually displayed at the mouse cursor location?

I thought about mouse event as source. The Y coord will be a bit different comparing to original right-button click that caused the menu to show. And in the case of you noted above if popup displayed not at cursor location the menu item selection click will also be not at proper scrollbar area. So it is not clear what we need to do in this case.

#37 - 12/08/2017 06:57 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

Hynek Cihlar wrote:

Eugenie, why not to get the location from the mouse event? What if the popup isn't actually displayed at the mouse cursor location?

I thought about mouse event as source. The Y coord will be a bit different comparing to original right-button click that caused the menu to show. And in the case of you noted above if popup displayed not at cursor location the menu item selection click will also be not at proper scrollbar area. So it is not clear what we need to do in this case.

Yes, the coordinates must be taken from the mouse event causing the popup to come up and not from the menu item click event.

#38 - 12/08/2017 10:42 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

Yes, the coordinates must be taken from the mouse event causing the popup to come up and not from the menu item click event.

Agreed. And this is exactly how the coords are currently passing from ScrollBarGuiImpl to menu.showPopup() and further to ScrollPopupGuiImpl.doShowPopup().

#39 - 12/08/2017 12:40 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

Hynek Cihlar wrote:

Yes, the coordinates must be taken from the mouse event causing the popup to come up and not from the menu item click event.

Agreed. And this is exactly how the coords are currently passing from ScrollBarGuiImpl to menu.showPopup() and further to ScrollPopupGuiImpl.doShowPopup().

In your last change you used the menu popup location passed to Menu.showPopup(), which at this time does equal to the mouse click location, but will eventually change once somebody fixes the popup to always show on screen (the 4GL behavior). The mouse location should be passed as new argument to ScrollPopupGuiImpl.create() called in ScrollBarGuiImpl.showPopup() to prevent it to break in the future.

#40 - 12/08/2017 03:02 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

In your last change you used the menu popup location passed to Menu.showPopup(), which at this time does equal to the mouse click location, but will eventually change once somebody fixes the popup to always show on screen (the 4GL behavior). The mouse location should be passed as new argument to ScrollPopupGuiImpl.create() called in ScrollBarGuiImpl.showPopup() to prevent it to break in the future.

Task branch 3394a has been updated for review to revision 11248.

Done. Notes resolution change. The initial coords are not passing during pop-up create.

#41 - 12/08/2017 08:06 PM - Eugenie Lyzenko

Guys,

I'm just trying to run browse pop-up test in my Windows devbox. And has abend from inability to open temp table, RecordBuffer.openScope() call:

```
...
Caused by: java.lang.ExceptionInInitializerError
    at com.goldencode.p2j.persist.TemporaryBuffer.openMultiplexScope (TemporaryBuffer.java:3591)
    at com.goldencode.p2j.persist.TemporaryBuffer.openScope (TemporaryBuffer.java:3248)
    at com.goldencode.p2j.persist.RecordBuffer.openScope (RecordBuffer.java:3096)
    at com.goldencode.testcases.browse.gui.BrowseGuiPopup.lambda$execute$1 (BrowseGuiPopup.java:29)
    at com.goldencode.p2j.util.Block.body (Block.java:604)
    at com.goldencode.p2j.util.BlockManager.processBody (BlockManager.java:6985)
    at com.goldencode.p2j.util.BlockManager.topLevelBlock (BlockManager.java:6776)
    at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:343)
    at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:317)
    at com.goldencode.testcases.browse.gui.BrowseGuiPopup.execute (BrowseGuiPopup.java:25)
    at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke (Method.java:498)
    at com.goldencode.p2j.util.Utills.invoke (Utills.java:1341)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute (StandardServer.java:1981)
    at com.goldencode.p2j.main.StandardServer.invoke (StandardServer.java:1476)
    at com.goldencode.p2j.main.StandardServer.standardEntry (StandardServer.java:525)
    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.run (Conversation.java:232)
    at java.lang.Thread.run (Thread.java:748)
Caused by: java.lang.IllegalStateException: It is expected to have InMemoryClassLoader in the chain of system
class loaders.
    at com.goldencode.p2j.persist.DynamicTablesHelper.<clinit> (DynamicTablesHelper.java:204)
    at com.goldencode.p2j.persist.TemporaryBuffer.openMultiplexScope (TemporaryBuffer.java:3591)
    at com.goldencode.p2j.persist.TemporaryBuffer.openScope (TemporaryBuffer.java:3248)
    at com.goldencode.p2j.persist.RecordBuffer.openScope (RecordBuffer.java:3096)
    at com.goldencode.testcases.browse.gui.BrowseGuiPopup.lambda$execute$1 (BrowseGuiPopup.java:29)
    at com.goldencode.p2j.util.Block.body (Block.java:604)
    at com.goldencode.p2j.util.BlockManager.processBody (BlockManager.java:6985)
    at com.goldencode.p2j.util.BlockManager.topLevelBlock (BlockManager.java:6776)
    at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:343)
    at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:317)
    at com.goldencode.testcases.browse.gui.BrowseGuiPopup.execute (BrowseGuiPopup.java:25)
    at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke (Method.java:498)
    at com.goldencode.p2j.util.Utills.invoke (Utills.java:1341)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute (StandardServer.java:1981)
    at com.goldencode.p2j.main.StandardServer.invoke (StandardServer.java:1476)
    at com.goldencode.p2j.main.StandardServer.standardEntry (StandardServer.java:525)
    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.run(Conversation.java:232)
at java.lang.Thread.run(Thread.java:748)
```

The other tests that do not use temp tables working good. Something is missing in config? or server startup?

Is somebody able to run browse related testcases in Windows FWD?

#42 - 12/09/2017 04:19 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

Guys,

I'm just trying to run browse pop-up test in my Windows devbox. And has abend from inability to open temp table, RecordBuffer.openScope() call:

[...]

The other tests that do not use temp tables working good. Something is missing in config? or server startup?

Is somebody able to run browse related testcases in Windows FWD?

How do you run the test case? Do you have -Djava.system.class.loader=com.goldencode.p2j.classloader.MultiClassLoader defined for the server process?

#43 - 12/09/2017 07:43 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

Eugenie Lyzenko wrote:

Guys,

I'm just trying to run browse pop-up test in my Windows devbox. And has abend from inability to open temp table, RecordBuffer.openScope() call:

[...]

The other tests that do not use temp tables working good. Something is missing in config? or server startup?

Is somebody able to run browse related testcases in Windows FWD?

How do you run the test case? Do you have `-Djava.system.class.loader=com.goldencode.p2j.classloader.MultiClassLoader` defined for the server process?

Thanks. This is a key point here. After proper set up adding MultiClassLoader the browse test started and working fine.

#44 - 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.

Can I close this task?

#45 - 01/23/2018 08:10 AM - Greg Shah

I can close this, right?

#46 - 01/23/2018 08:22 AM - Hynek Cihlar

Greg Shah wrote:

I can close this, right?

Yes, I think so.

#47 - 01/23/2018 08:28 AM - Eugenie Lyzenko

I also think it can be closed.

#48 - 01/23/2018 08:34 AM - Greg Shah

- % Done changed from 0 to 100
- Status changed from WIP to Closed
- Start date deleted (11/22/2017)

#49 - 02/09/2018 11:55 AM - Greg Shah

- Related to Feature #3261: enhanced browse that can optionally selected as a replacement for the default ABL browse added