User Interface - Feature #6764

Improve trigger selection for keyboard events.

09/15/2022 08:15 AM - Marian Edu

Status:	Review	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Marian Edu	**% Done:**	80%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		

Description

History

#1 - 09/15/2022 08:17 AM - Marian Edu

For keyboard events there is a specific order on which 4GL is looking for the right trigger to fire:
- key label
- key function
- any-printable (if appropriate)
- any-key

It is using that order first for exact widget, then for the widget with anywhere option and then it goes up the widget's parents tree and as last resort it looks for anywhere without any widget (default handler).

Current implementation is using recursion to lookup the trigger, this might be improved to call for all key events in a single pass (with or without anywhere) and there is still one case we've found where FWD is picking a different trigger compared to the 4GL.

#2 - 09/28/2022 08:11 AM - Marian Edu

- File 6764.01.patch added

- Status changed from New to WIP

- Assignee set to Marian Edu

- % Done changed from 0 to 80

Since #6528 is already closed I'll use this task to upload the patch that tries to clean-up/improve the event lookup especially for keyboard events (label/function/any printable/any key).

There are a few differences still for combo-box when the drop-down list is being shown but otherwise the event lookup seems to match the 4GL side now - the 4GL test used in testcases/redmine/6528/frame_insert.w.

#5 - 09/28/2022 08:13 AM - Marian Edu

- Status changed from WIP to Review

Please review the patch and let me know if you see anything wrong or something I've missed - doing a smoke test on <large_gui_app> atm.

#6 - 09/28/2022 08:20 AM - Greg Shah

- Description updated

Constantin/Hynek: Please review.

#7 - 09/29/2022 02:56 AM - Hynek Cihlar

Code review 6764.01.patch.

It is very nice that the result is a lot cleaner. I didn't find any particular issue from looking at the code.

I'm only worried about the removed browser-specific behavior and "frame/window propagation" in EventDefintion.lookup. Will these be missing for their specific use cases?

Also UiContainer.locateContainer javadoc deserves clarification. Sometimes it mentions "container", sometimes "frame", but it never mentions browse which can be returned, too. Also the statement "the given widget is itself a container" is also not accurate.

#8 - 09/29/2022 07:07 AM - Marian Edu

- File 6764.02.patch added

Hynek Cihlar wrote:

> Code review 6764.01.patch.
>
> I'm only worried about the removed browser-specific behavior and "frame/window propagation" in EventDefintion.lookup. Will these be missing for their specific use cases?

No, the browse use-case is handled by using the locateContainer in ThinClient when looking-up the event.

> Also UiContainer.locateContainer javadoc deserves clarification. Sometimes it mentions "container", sometimes "frame", but it never mentions browse which can be returned, too. Also the statement "the given widget is itself a container" is also not accurate.

Agreed, copy/paste is evil... tried to fix the comment, attached the new patch.

#9 - 09/29/2022 08:11 AM - Greg Shah

Stanislav: Please review.

#10 - 09/29/2022 08:14 AM - Hynek Cihlar

Marian Edu wrote:

> Hynek Cihlar wrote:
>
> > Code review 6764.01.patch.
> >
> > I'm only worried about the removed browser-specific behavior and "frame/window propagation" in EventDefintion.lookup. Will these be missing for their specific use cases?
>
> No, the browse use-case is handled by using the locateContainer in ThinClient when looking-up the event.

OK. What about the "frame/window propagation"?

> > Also UiContainer.locateContainer javadoc deserves clarification. Sometimes it mentions "container", sometimes "frame", but it never mentions browse which can be returned, too. Also the statement "the given widget is itself a container" is also not accurate.
>
> Agreed, copy/paste is evil... tried to fix the comment, attached the new patch.

It looks good now.

One more thing. In UiUitls.getTopLevelWindow, top-level window for dialogs depends on the visibility of the immediate top-level window. It seems confusing, the window is still there, it's just not visible, why shouldn't it be returned? If you do need to consider the visibility state I would do it in the caller of the method.

#11 - 09/29/2022 08:30 AM - Marian Edu

Hynek Cihlar wrote:

> OK. What about the "frame/window propagation"?

That is handled in lookupWorker, first walking through parent frames (UiUtils.getParentFrames) and then the 'top level window' (see below).

> One more thing. In UiUitls.getTopLevelWindow, top-level window for dialogs depends on the visibility of the immediate top-level window. It seems confusing, the window is still there, it's just not visible, why shouldn't it be returned? If you do need to consider the visibility state I would do it in the caller of the method.

The code was 'extracted' as-is from the lookupWorker method, I can't really comment on why the window visibility for the dialog is being considered... it just looked like something that should be extracted out and UiUtils seemed like a good place to put it :(

#12 - 09/29/2022 09:02 AM - Hynek Cihlar

Marian Edu wrote:

> Hynek Cihlar wrote:
>
> > OK. What about the "frame/window propagation"?
>
> That is handled in lookupWorker, first walking through parent frames (UiUtils.getParentFrames) and then the 'top level window' (see below).

OK.

> > One more thing. In UiUitls.getTopLevelWindow, top-level window for dialogs depends on the visibility of the immediate top-level window. It seems confusing, the window is still there, it's just not visible, why shouldn't it be returned? If you do need to consider the visibility state I would do it in the caller of the method.
>
> The code was 'extracted' as-is from the lookupWorker method, I can't really comment on why the window visibility for the dialog is being considered... it just looked like something that should be extracted out and UiUtils seemed like a good place to put it :(

With this kind of specialized behavior it probably doesn't make sense to be placed in utils, what do you think?

#13 - 09/29/2022 09:55 AM - Marian Edu

Hynek Cihlar wrote:

> With this kind of specialized behavior it probably doesn't make sense to be placed in utils, what do you think?

You might be right, maybe I can just move it back to EventList as aa private method... looking at it it does look like it will return null for a dialog if it's top level window isn't visible but as said I really don't understand why it is like that :(

#14 - 09/29/2022 02:41 PM - Stanislav Lomany

- File dbl1.p added

> 6764.02.patch

There's a browse-specific piece of code which was dropped, and this part is regressed.
Testcase attached. Try double-clicking all columns - only this last one should produce an event.

#15 - 09/30/2022 04:12 AM - Marian Edu

- File dbl2.p added

Stanislav Lomany wrote:

> > 6764.02.patch
>
> There's a browse-specific piece of code which was dropped, and this part is regressed.
> Testcase attached. Try double-clicking all columns - only this last one should produce an event.

Thanks for that Stanislav, that is interesting and I think it makes sense... when looking up the parent's tree only events registered with anywhere should be considered.

If you try this test you will see FWD is (and was) behaving differently compared to 4GL, somehow it manages to avoid the fire on the browse if you double-click on en editable cell but that should not happen if the event was added with anywhere.

Looking at it and see if I can restrict the parents lookup to only anywhere triggers as it seems to happen in 4GL, otherwise the 'parent' for the browse column it is the browse as you can see when running the tests it's just that the current implementation is not filtering out parent triggers without anywhere.

#16 - 10/03/2022 02:48 AM - Marian Edu

- % Done changed from 80 to 90

I did a bit of additional clean-up, resolve function key code in lookupWorker and don't waste time looking up through widget's parent chain if there is no event registered for the event code (function code, any-printable, any-key as needed) and only consider anywhere triggers when going through the parents tree - this is solving the issue from Stanislav test.

Attached the new patch.

#17 - 10/03/2022 02:50 AM - Marian Edu

Marian Edu wrote:

> Attached the new patch.

Looks like attachments are not accepted, it gives an 'internal server error' instead :(

#18 - 10/03/2022 07:40 AM - Marian Edu

- File 6764.03.patch added

Marian Edu wrote:

> Marian Edu wrote:
>
>> Attached the new patch.

#19 - 10/04/2022 09:45 AM - Greg Shah

Hynek/Stanislav: Please review.

#20 - 10/05/2022 07:09 AM - Stanislav Lomany

> 6764.03.patch

I've tested misc related cases for the browse and found no regressions. I'm OK with the changes for the browse part.

#21 - 10/10/2022 04:14 PM - Greg Shah

Hynek: Are you OK with the proposed patch?

#22 - 10/11/2022 03:53 AM - Hynek Cihlar

Code review 6764.03.patch.

The changes look good, I didn't do any regression testing though. Just a few minor comments below.

In EventDefinition.lookup the condition for "global anywhere" excludes resource check, while resource check is in place in isAnywhereGlobally: widgetId == -1 && anywhere && !hasWidget(). Is this expected?

In EventList.lookupWorker:

```
         // lookup event for the widget and honor any-printable/any-key if needed (specific or anywhere)
         if (lookupEventHelper(wid, evCode, functionCode, trig, EventDefinition.SCOPE_SELF | EventDefinition.S
COPE_ANYWHERE, isKey, isPrintable,
                  result) != -1 || specificWidget)
         {
            return;
         }
```

move specificWidget before lookupEventHelper.

Change ThinClient.getInstance().getWidget(parentId); back to tc.getWidget(parentId); in EventList.lookupWorker. tc variable is available in the scope.

#23 - 10/11/2022 06:54 AM - Marian Edu

- File 6764.04.patch added

Hynek Cihlar wrote:

> In EventDefinition.lookup the condition for "global anywhere" excludes resource check, while resource check is in place in isAnywhereGlobally: widgetId == -1 && anywhere && !hasWidget(). Is this expected?

Thanks for that Hynek, was an overlook - changed to use the isAnywhereGlobally instead.

> In EventList.lookupWorker:
> [...]
> move specificWidget before lookupEventHelper.

Actually that is like that on purpose, the idea is to return if we've found something or we don't need to check the parent's chain for anywhere triggers.

> Change ThinClient.getInstance().getWidget(parentId); back to tc.getWidget(parentId); in EventList.lookupWorker. tc variable is available in the scope.

Right, leftover from the move back from UiUtils.

Attached the new patch, made against #14277.

#24 - 10/11/2022 09:57 AM - Hynek Cihlar

Marian Edu wrote:

> Hynek Cihlar wrote:
>
> > move specificWidget before lookupEventHelper.
>
> Actually that is like that on purpose, the idea is to return if we've found something or we don't need to check the parent's chain for anywhere triggers.

I see, makes sense.

The changes look good, I have no more concerns.

#25 - 10/11/2022 10:04 AM - Greg Shah

Hynek: Please merge the changes into 3821a.

#26 - 10/11/2022 10:30 AM - Hynek Cihlar

Greg Shah wrote:

> Hynek: Please merge the changes into 3821a.

Will do, I assume you mean 3821c.

#27 - 10/11/2022 10:45 AM - Greg Shah

Yes, sorry.

#28 - 10/11/2022 10:59 AM - Hynek Cihlar

6764.04.patch checked in 3821c revision 14285.

#29 - 10/13/2022 08:15 PM - Greg Shah

Is there anything left open in this task?

#30 - 10/14/2022 01:42 AM - Marian Edu

Greg Shah wrote:

> Is there anything left open in this task?

I don't have anything left, all our tests in 4GL are passing so far although the only thing we've tested were UI triggers.

#31 - 10/14/2022 03:37 AM - Hynek Cihlar

Marian Edu wrote:

> Greg Shah wrote:
>
> > Is there anything left open in this task?
>
> I don't have anything left, all our tests in 4GL are passing so far although the only thing we've tested were UI triggers.

I have a potential regression, I will confirm shortly.

#32 - 10/14/2022 04:17 AM - Hynek Cihlar

Hynek Cihlar wrote:

> Marian Edu wrote:
>
> > Greg Shah wrote:
> >
> > > Is there anything left open in this task?

> > I don't have anything left, all our tests in 4GL are passing so far although the only thing we've tested were UI triggers.
>
> I have a potential regression, I will confirm shortly.

A false alarm. So nothing left on my side.

#33 - 10/14/2022 05:30 AM - Greg Shah

- Status changed from Review to Closed

- % Done changed from 90 to 100

#34 - 10/14/2022 05:30 AM - Greg Shah

- Status changed from Closed to Test

#35 - 10/14/2022 05:30 AM - Greg Shah

- Status changed from Test to Closed

#36 - 10/14/2022 10:35 AM - Hynek Cihlar

- File EventList.java.20221014a.diff added

There is a regression after all. The test case is as follows:

```
def var fi1 as char.
define frame f fi1.
enable all with frame f.

on any-key anywhere do:
  message "a key". // this isn't triggered after any key pressed
end.

wait-for go of frame f.
```

This causes issue #6840.

I'm attaching a fix candidate. Marian, please review.

#37 - 10/14/2022 11:02 AM - Marian Edu

Hynek Cihlar wrote:

> There is a regression after all. The test case is as follows:
>
> [...]
>
> This causes issue #6840.
>
> I'm attaching a fix candidate. Marian, please review.

This is interesting, running the test I was puzzled as for why the ANYWHERE is not true for that EventDefinition - it was defined as ANYWHERE and then did look at your patch :)

I don't see any reason to reset ANYWHERE there since I've dropped the 'GLOBAL_ANYWHERE@ - this is an ANYWHERE with no widget/resource set.

The patch is good, thank you.

#38 - 10/14/2022 11:05 AM - Hynek Cihlar

Marian Edu wrote:

> The patch is good, thank you.

I will run some more regression tests before I check it in.

#39 - 10/14/2022 11:34 AM - Marian Edu

Hynek Cihlar wrote:

> Marian Edu wrote:
>
> > The patch is good, thank you.
>
> I will run some more regression tests before I check it in.

The method addEvent(String[] event, boolean anywhere) seems to be broken as it disregard the value for anywhere and send false instead but it looks like is not used anywhere in code (at least not in <large_gui_application>). The constructor with same signature is also disregarding the value of input parameter and always use false - at least one use of this one in <large_gui_application> (some_program.w - on alt-page-up, alt-page-down anywhere).

I think I need to add more combinations to the testcase I've used :(

#40 - 10/15/2022 04:10 AM - Hynek Cihlar

Marian Edu wrote:

> Hynek Cihlar wrote:
>
> > Marian Edu wrote:
> >
> > > The patch is good, thank you.
> >
> > I will run some more regression tests before I check it in.
>
> The method addEvent(String[] event, boolean anywhere) seems to be broken as it disregard the value for anywhere and send false instead but it looks like is not used anywhere in code (at least not in <large_gui_application>). The constructor with same signature is also disregarding the value of input parameter and always use false - at least one use of this one in <large_gui_application> (some_program.w - on alt-page-up, alt-page-down anywhere).

This looks suspicious and deserves attention. OTOH if it doesn't cause issues we can defer it. Looking in the history, it seems it's been in there since revision 11115.1.2.

#41 - 10/16/2022 07:15 AM - Greg Shah

> This looks suspicious and deserves attention. OTOH if it doesn't cause issues we can defer it. Looking in the history, it seems it's been in there since revision 11115.1.2.

At a minimum, please put a big TODO and BUG comment in that code so that we can't miss it later. If you can fix it now in a reasonable amount of work, go ahead.

#42 - 10/18/2022 06:29 AM - Hynek Cihlar

- File EventDefinition.java.20221018.diff added

There is a regression in a customer ChUI application. The cause was the following removed from one of the EventDefinition constructors:

```
      this.globalAnywhere = ((widget == null || widget == WidgetId.INVALID_WIDGET_ID) &&
                             (resource == null || resource == ResourceIdHelper.INVALID_RESOURCE));
```

Marian, please review the attached patch.

#43 - 10/18/2022 06:44 AM - Marian Edu

Hynek Cihlar wrote:

> There is a regression in a customer ChUI application. The cause was the following removed from one of the EventDefinition constructors:
>
> [...]
>
> Marian, please review the attached patch.

I guess this will work, problem is where is this ctor called with anywhere set to false and no widget nor resource? I think it's only from those addEvent method and ctor where anywhere parameter is ignored and false is sent instead. I've fixed those already but the patch is good as a safeguard, we can assume an event with no widget/resource is anywhere (globally) I guess.

#44 - 10/18/2022 07:00 AM - Hynek Cihlar

Marian Edu wrote:

> Hynek Cihlar wrote:
>
> > There is a regression in a customer ChUI application. The cause was the following removed from one of the EventDefinition constructors:
> >
> > [...]
> >
> > Marian, please review the attached patch.
>
> I guess this will work, problem is where is this ctor called with anywhere set to false and no widget nor resource? I think it's only from those addEvent method and ctor where anywhere parameter is ignored and false is sent instead. I've fixed those already but the patch is good as a safeguard, we can assume an event with no widget/resource is anywhere (globally) I guess.

This is the call stack where the definition gets created. It comes from the authentication plugin of the customer app.

```
<init>:306, EventDefinition {com.goldencode.p2j.ui}
addEvent:1314, EventList {com.goldencode.p2j.ui}
addEvent:907, EventList {com.goldencode.p2j.ui}
addEvent:1198, EventList {com.goldencode.p2j.ui}
<init>:332, EventList {com.goldencode.p2j.ui}
clientAuthHook:346, LoginClient {package censored}
```

#45 - 10/18/2022 07:01 AM - Hynek Cihlar

Hynek Cihlar wrote:

> This is the call stack where the definition gets created. It comes from the authentication plugin of the customer app.
>
> [...]

The line numbers match with 3821c revision 14284.

#46 - 10/18/2022 09:23 AM - Marian Edu

Hynek Cihlar wrote:

> The line numbers match with 3821c revision 14284.

Thanks for that Hynek, some more clean-up to do there...

#47 - 10/18/2022 09:24 AM - Marian Edu

- % Done changed from 100 to 80

#49 - 11/10/2022 07:05 AM - Marian Edu

- File 6764.05.patch added

It seems I can't re-open this one but this is a patch that fixes the global anywhere in event definition, universal events handling (parent frame lookup if not consumed), fixes for default help (only if no trigger fired) and some other small changes.

#51 - 11/10/2022 07:16 AM - Greg Shah

Hynek: Please review.

#52 - 11/10/2022 07:16 AM - Greg Shah

- Status changed from Closed to Review

#53 - 11/10/2022 07:55 AM - Marian Edu

Greg Shah wrote:

> Hynek: Please review.

Greg, this will kinda re-open #6195 - if conversion isn't fixed (#6901) the code in mainwinfwd.w might be updated by adding anywhere on first event (line 312), proven this is allowed by conversion:

```
on end-error of W-Win anywhere ...
```

#54 - 11/10/2022 08:20 AM - Greg Shah

I'm inclined to resolve #6901 sooner rather than later.

#55 - 11/15/2022 05:41 AM - Marian Edu

- File 6764.06.patch added

An updated patch to solve some NPE issues (thanks Tomasz for reporting it), it's made against rev #14360.

#56 - 11/16/2022 04:41 AM - Greg Shah

Hynek: Please review the latest patch.

Marian: I'd like to get these changes and the fix for #6901 / #6930 into 3821c ASAP.

#57 - 11/16/2022 07:10 AM - Hynek Cihlar

Marian Edu wrote:

> An updated patch to solve some NPE issues (thanks Tomasz for reporting it), it's made against rev #14360.

When does the client send event code of -1 to LT.apply? Should this be prevented on the client so it doesn't happen at all? Should it be logged in LT.apply?

#58 - 11/16/2022 07:12 AM - Marian Edu

Hynek Cihlar wrote:

> When does the client send event code of -1 to LT.apply? Should this be prevented on the client so it doesn't happen at all? Should it be logged in LT.apply?

This was not from the client but coming from an `apply lastkey` right after a message statement (that seems to simply reset lastkey).

#59 - 11/16/2022 07:14 AM - Hynek Cihlar

Marian Edu wrote:

> Hynek Cihlar wrote:
>
> > When does the client send event code of -1 to LT.apply? Should this be prevented on the client so it doesn't happen at all? Should it be logged in LT.apply?
>
> This was not from the client but coming from an `apply lastkey` right after a message statement (that seems to simply reset lastkey).

I see. Can you describe this case in the comment at the return statement in LT.apply?

I have no more concerns.

#60 - 12/05/2022 02:16 PM - Hynek Cihlar

Marian, Greg, I'd like to get 6764.06.patch in 3821c. Any objection?

#61 - 12/05/2022 02:33 PM - Tomasz Domin

Hi
What is the status of this issue ?
I'd like to move on with Regression Tests fix and events/triggers are next big thing to work on.
Shall I hold on ?

#62 - 12/05/2022 02:34 PM - Greg Shah

> I'd like to get 6764.06.patch in 3821c. Any objection?

No objection from me.

#63 - 12/05/2022 02:47 PM - Marian Edu

Greg Shah wrote:

> > I'd like to get 6764.06.patch in 3821c. Any objection?
>
> No objection from me.

None from me either, there is still widget specific code in thin-client/logical-terminal that might need some consideration but my tests are passing.

#64 - 12/06/2022 03:48 AM - Hynek Cihlar

Marian Edu wrote:

> Greg Shah wrote:
>
> > > I'd like to get 6764.06.patch in 3821c. Any objection?

> > No objection from me.
>
> None from me either, there is still widget specific code in thin-client/logical-terminal that might need some consideration but my tests are passing.

Marian, please add a file history entry to ThinClient.java and also fix the one in EventDefinition.java. Each entry must contain the author's initials.

#65 - 12/06/2022 07:00 AM - Marian Edu

- File 6764.07.patch added

Hynek Cihlar wrote:

> Marian, please add a file history entry to ThinClient.java and also fix the one in EventDefinition.java. Each entry must contain the author's initials.

Hynek, fixed those... the new patch attached (against rev 14399), thanks.

#66 - 12/06/2022 07:21 AM - Hynek Cihlar

6764.07.patch checked in to 3821c revision 14417.

#67 - 12/07/2022 03:33 AM - Hynek Cihlar

Hynek Cihlar wrote:

> 6764.07.patch checked in to 3821c revision 14417.

I reverted the patch in 3821c revision 14420 due to a regression.

#68 - 12/16/2022 08:50 AM - Marian Edu

- File 6764.08.patch added

Hynek Cihlar wrote:

> Hynek Cihlar wrote:

6764.07.patch checked in to 3821c revision 14417.

I reverted the patch in 3821c revision 14420 due to a regression.

Attached the patch (against rev 14444), missed a full block when key code was not valid so action should have been invoked instead :(

#69 - 12/19/2022 12:03 PM - Tomasz Domin

Please see #6667-268 for regression tests results for 6764.08.patch

#70 - 12/21/2022 10:05 AM - Hynek Cihlar

Code review 6764.08.patch. The changes look good. Can I merge it to 3821c?

#71 - 12/21/2022 10:06 AM - Greg Shah

Yes, please go ahead.

#72 - 12/21/2022 10:35 AM - Hynek Cihlar

6764.08.patch checked in to 3821c revision 14462.

Files

6764.01.patch	34.4 KB	09/28/2022	Marian Edu
6764.02.patch	34.6 KB	09/29/2022	Marian Edu
dbl1.p	2.22 KB	09/29/2022	Stanislav Lomany
dbl2.p	2.65 KB	09/30/2022	Marian Edu
6764.03.patch	49.9 KB	10/03/2022	Marian Edu
6764.04.patch	49.4 KB	10/11/2022	Marian Edu
EventList.java.20221014a.diff	523 Bytes	10/14/2022	Hynek Cihlar
EventDefinition.java.20221018.diff	592 Bytes	10/18/2022	Hynek Cihlar
6764.05.patch	14.1 KB	11/10/2022	Marian Edu
6764.06.patch	14.6 KB	11/15/2022	Marian Edu
6764.07.patch	15.4 KB	12/06/2022	Marian Edu
6764.08.patch	15.6 KB	12/16/2022	Marian Edu