

User Interface - Bug #3013

Bug # 2677 (New): fix drawing and functional differences between P2J GUI and 4GL GUI

The mouse wheel doesn't work if the mouse pointer is over scrollbars

03/03/2016 09:36 AM - Sergey Ivanovskiy

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Sergey Ivanovskiy	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#1 - 03/03/2016 09:44 AM - Sergey Ivanovskiy

1) in ./demo/demo_widgets.p (com.goldencode.testcases.demo.DemoWidgets.execute) the editor doesn't scroll its content horizontally by the mouse, but it can scroll vertically.
2) the focus doesn't move to the target widget if the mouse is over it

#2 - 03/03/2016 09:59 AM - Sergey Ivanovskiy

The code from AbstractWidget implements this function unconditionally to a possible widget id.

```
@Override
public Widget<O> findMouseSource(NativePoint p)
{
    return isVisible() && this.physicalBounds().contains(p) ? this : null;
}
```

#3 - 03/03/2016 11:15 AM - Sergey Ivanovskiy

- File 3013_1.txt added

Greg, please review committed revision 11051. It fixed the part 1). Tested with Swing and Web clients. The rest part 2 needs to be compared with the old 1811t before the current rebase.

#4 - 03/03/2016 01:00 PM - Sergey Ivanovskiy

Web client - For the old version 1811t if the mouse pointer is over the scroll bars, then the target scroll bar gets a focus. But for the current version of 1811t it is broken.

#5 - 03/03/2016 01:16 PM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

Web client - For the old version 1811t if the mouse pointer is over the scroll bars, then the target scroll bar gets a focus. But for the current version of 1811t it is broken.

Greg, I have checked web client again and this part 2) works properly. Thus all found issues are fixed.

#6 - 03/04/2016 08:58 AM - Greg Shah

Code Review Task Branch 1811t Revision 11051

I'm fine with the changes.

My only concern is that the changes were put there for a specific reason.

Igor: Please review this revision (which removes some of your changes) and let us know what specific use cases are expected to be broken now.

#7 - 03/04/2016 09:11 AM - Sergey Ivanovskiy

Code Review Task Branch 1811t Revision 11051

My only concern is that the changes were put there for a specific reason.

Greg, 2565 changed the implementation of `AbstractContainer.findMouseSource(NativePoint p)`. I think this change is incorrect. The first reason is it alters the method implementation of the abstract base class. The function now returns a mouse source widget iff this widget has an id. It is incorrect and there are no provided reasons to change its behavior in a such manner.

#8 - 03/04/2016 09:31 AM - Igor Skornyakov

Greg Shah wrote:

Code Review Task Branch 1811t Revision 11051

I'm fine with the changes.

My only concern is that the changes were put there for a specific reason.

Igor: Please review this revision (which removes some of your changes) and let us know what specific use cases are expected to be broken now.

Indeed, this change was made to provide a correct support of the custom mouse cursor for the COMBO-BOX as I've explained yesterday. However the code I was working with contained check for the non-null id for found (but not for w).

The whole stuff around findMouseSource/ids looks extremely tricky and fragile and it is difficult to judge about a particular change w/o testing. In this case the custom cursor support for COMBO-BOX is most likely broken.

#9 - 03/04/2016 09:40 AM - Sergey Ivanovskiy

Igor Skorniyakov wrote:

Greg Shah wrote:

Code Review Task Branch 1811t Revision 11051

I'm fine with the changes.

My only concern is that the changes were put there for a specific reason.

Igor: Please review this revision (which removes some of your changes) and let us know what specific use cases are expected to be broken now.

Indeed, this change was made to provide a correct support of the custom mouse cursor for the COMBO-BOX as I've explained yesterday. However the code I was working with contained check for the non-null id for found (but not for w). The whole stuff around findMouseSource/ids looks extremely tricky and fragile and it is difficult to judge about a particular change w/o testing. In this case the custom cursor support for COMBO-BOX is most likely broken.

Igor, could you provide test cases to check this test case?

#10 - 03/04/2016 09:44 AM - Sergey Ivanovskiy

I have check the behavior of the default cursor with combo box, but how it can be if the custom don't work while the default works properly?

#11 - 03/04/2016 09:47 AM - Igor Skorniyakov

Sergey Ivanovskiy wrote:

I have check the behavior of the default cursor with combo box, but how it can be if the custom don't work while the default works properly?

This means that it is broken as I expected. To have the default cursor working almost nothing is required. My change was for the custom cursor.

#12 - 03/04/2016 09:50 AM - Sergey Ivanovskiy

Igor Skorniyakov wrote:

Sergey Ivanovskiy wrote:

I have checked the behavior of the default cursor with combo box, but how it can be if the custom doesn't work while the default works properly?

This means that it is broken as I expected. To have the default cursor working almost nothing is required. My change was for the custom cursor.

Do you have some tests to check it?

#13 - 03/04/2016 09:51 AM - Greg Shah

Igor: please explain to Sergey, how to test the custom cursor for combo-box.

Sergey: If it is broken, please find a fix which doesn't depend on changes at the AbstractContainer level.

#14 - 03/04/2016 09:53 AM - Igor Skorniyakov

Sergey Ivanovskiy wrote:

Do you have some tests to check it?

The test for the custom cursor functionality is `uast/mouse/load-mouse-pointer.p`. The `uast/mouse/mouse-cursor-test-plan.txt` contains the test plan.

#15 - 03/04/2016 09:54 AM - Sergey Ivanovskiy

Igor Skorniyakov wrote:

Sergey Ivanovskiy wrote:

Do you have some tests to check it?

The test for the custom cursor functionality is `uast/mouse/load-mouse-pointer.p`. The `uast/mouse/mouse-cursor-test-plan.txt` contains the test plan.

Thank you, don't worry it should be fixed.

#16 - 03/04/2016 10:53 AM - Sergey Ivanovskiy

I understand the great work has been done with custom cursors, this test is really complicated and it looks that these tests are passed

0. check the the cursor is default everywhere except the EDITOR, FILL-INS and entry fields of cb1 /cb2 (should be IBEAM).
1. Select a custom mouse pointer for the window via wcur combo box (e.g. WAIT). Make sure that it is inherited by all widgets except entry fields of cb1 and cb2 and the selection list of cb1.
2. Select a custom mouse pointer for the frame via fcur combo box (e.g. CROSS). Make sure that it is inrerited by all widgets except mentioned above and the window.

For the others tests I would like to compare with p2j trunc to understand if the changes are okay or bad.

#17 - 03/04/2016 11:47 AM - Sergey Ivanovskiy

The things are very complicated because I suspect the test 1 for p2j trunc in the case of COMBO-BOX SIMPLE because if we point the mouse over the drop down component's scroll bar, then it displays wait cursor but if the mouse pointer is inside the drop down list, then the cursor is arrow (default).

But this test case with my changes doesn't pass, the cursor doesn't change to arrow if the cursor is over the drop down (Swing mode), but it works with web client.

I would like to work on COMBO-BOX SIMPLE, because for p2j it also doesn't work, can't detect its scroll bar. It gives me more confident feelings to proceed with my fix. Again, don't worry.

#18 - 03/04/2016 01:06 PM - Sergey Ivanovskiy

Another reason with AbstractContainer.findMouseSource it can impact the focus management.

#19 - 03/04/2016 04:18 PM - Sergey Ivanovskiy

In p2j trunc version uast/mouse/load-mouse-pointer.p doesn't pass the 1 test

1. Select a custom mouse pointer for the window via wcur combo box (e.g. WAIT). Make sure that it is inherited by all widgets except entry fields of cb1 and cb2 and the selection list of cb1.

This scenario is failed. Select WAIT cursor in the "wcur" combobox, open the "cb2" combobox, point the mouse over the opened drop down, then the mouse pointer isn't changed if it is over the "cb2" drop down. Continue working to find the reasons and how it can be fixed.

#20 - 03/04/2016 04:30 PM - Sergey Ivanovskiy

- File 3013_2.txt added

Planning to commit this diff that moves 1811t to be the same as p2j, but it doesn't fix these bugs, because there are present in P2j also.

#21 - 03/04/2016 04:53 PM - Igor Skorniyakov

Sergey Ivanovskiy wrote:

In p2j trunc version uast/mouse/load-mouse-pointer.p doesn't pass the 1 test
[...]

This scenario is failed. Select WAIT cursor in the "wcur" combobox, open the "cb2" combobox, point the mouse over the opened drop down, then the mouse pointer isn't changed if it is over the "cb2" drop down. Continue working to find the reasons and how it can be fixed.

This is actually a bad wording in the test plan. For the correct description see [#2565](#) note 27 sections N2, N3, N4 and N7. Sorry. What you describe is the correct behaviour.

#22 - 03/04/2016 04:54 PM - Greg Shah

I like the approach (in 3013_2.txt), please go ahead.

#23 - 03/04/2016 04:54 PM - Greg Shah

This is actually a bad wording in the test plan.

Igor: please update the test plan to be accurate.

#24 - 03/04/2016 05:12 PM - Igor Skorniyakov

Greg Shah wrote:

This is actually a bad wording in the test plan.

Igor: please update the test plan to be accurate.

Sorry. I've just read the description of the problem incorrectly. In fact the test plan is accurate and in the test I've run right now both cb1 and cb2 behave correctly.

#25 - 03/04/2016 05:18 PM - Greg Shah

I don't understand. Does note 19 describe a bug or not?

At a minimum, there is confusion here so perhaps some clarifications are still needed in the test plan. If Sergey executed the test plan and thinks there is a bug, but your execution of the same test plan shows no bug, then my guess is that the test plan is not specific enough in this case (such that Sergey is executing the test differently from you).

#26 - 03/04/2016 05:23 PM - Igor Skorniyakov

Greg Shah wrote:

I don't understand. Does note 19 describe a bug or not?

At a minimum, there is confusion here so perhaps some clarifications are still needed in the test plan. If Sergey executed the test plan and thinks there is a bug, but your execution of the same test plan shows no bug, then my guess is that the test plan is not specific enough in this case (such that Sergey is executing the test differently from you).

OK. Sergey, can you please describe your actions with more details (a screenshot will help a lot). After that I'll be able to understand if it is my bug or a test plan shortage.

#27 - 03/05/2016 04:14 AM - Igor Skorniyakov

- *File mouse.mkv added*

Here is the screencast of the mouse cursor test. Made with <http://www.maartenbaert.be/simplescreenrecorder/>

#28 - 03/06/2016 04:43 AM - Sergey Ivanovskiy

These steps are described in note 19

- 1) Run uast/mouse/load-mouse-pointer.p
- 2) Select "WAIT" cursor from "wcur" combobox. The cursor is changed to WAIT.
- 3) Open the "cb2" combobox, point the mouse over the opened drop down, then the mouse pointer isn't changed if it is over the "cb2" drop down.

#29 - 03/06/2016 04:50 AM - Sergey Ivanovskiy

Igor Skorniyakov wrote:

Here is the screencast of the mouse cursor test. Made with <http://www.maartenbaert.be/simplescreenrecorder/>

Igor, thank you for this material. I watched it and found that the drop down in your tests on step 1 was not opened. In that case I am planing to check it again that if apply 3013_2.txt, 1811t becomes identical to p2].

#30 - 03/06/2016 04:51 AM - Igor Skornyakov

Sergey Ivanovskiy wrote:

These steps are described in note 19

- 1) Run uast/mouse/load-mouse-pointer.p
- 2) Select "WAIT" cursor from "wcur" combobox. The cursor is changed to WAIT.
- 3) Open the "cb2" combobox, point the mouse over the opened drop down, then the mouse pointer isn't changed if it is over the "cb2" drop down.

Sorry, but according both to the test plan and [#2565](#) note 27. The mouse cursor is not inherited from the parent (window or frame) for the selection list of the SIMPLE DROP-BOX (cb1), not for DROP-DOWN-LIST (cb) or DROP-DOWB (cb2). If for cb2 you see WAIT at the drop down of cb2 it is a correct behaviour.

#31 - 03/06/2016 05:04 AM - Sergey Ivanovskiy

Igor Skornyakov wrote:

Sergey Ivanovskiy wrote:

These steps are described in note 19

- 1) Run uast/mouse/load-mouse-pointer.p
- 2) Select "WAIT" cursor from "wcur" combobox. The cursor is changed to WAIT.
- 3) Open the "cb2" combobox, point the mouse over the opened drop down, then the mouse pointer isn't changed if it is over the "cb2" drop down.

Sorry, but according both to the test plan and [#2565](#) note 27. The mouse cursor is not inherited from the parent (window or frame) for the selection list of the SIMPLE DROP-BOX (cb1), not for DROP-DOWN-LIST (cb) or DROP-DOWB (cb2). If for cb2 you see WAIT at the drop down of cb2 it is a correct behaviour.

Ok, understand. Please look at the next scenario (in note 17) Is this behavior correct?

- 1) Run uast/mouse/load-mouse-pointer.p
- 2) Select "WAIT" cursor from "wcur" combobox. The cursor is changed to WAIT.
- 3) Point the mouse over the "cb" combobox's scroll bar, then it displays the wait cursor but if the mouse pointer is inside its drop down list, then the cursor is arrow.

If it is ok, then we can apply to p2j trunc, these two fixes 3013_1.txt and 3013_2.txt. It gives the same behavior and fix the web, because web can't rely on the Swing GUI events propagation between components. And we need clean function findMouseSource that returns the closest component with or without id. What do you think?

#32 - 03/06/2016 05:08 AM - Igor Skornyakov

Sergey Ivanovskiy wrote:

Ok, understand. Please look at the next scenario (in note 17) Is this behavior correct?

- 1) Run uast/mouse/load-mouse-pointer.p
- 2) Select "WAIT" cursor from "wcur" combobox. The cursor is changed to WAIT.
- 3) Point the mouse over the "cb" combobox's scroll bar, then it displays the wait cursor but if the mouse pointer is inside its drop down list, then the cursor is arrow.

This is not correct. The cursor at the drop-down should be WAIT (both for cb and cb2). And in the trunk it works like this.

#33 - 03/06/2016 05:13 AM - Sergey Ivanovskiy

Ok. I will send you screen shot that is suspected (note 17) in P2j

#34 - 03/06/2016 05:28 AM - Igor Skornyakov

- File mouse1.mkv added

I've added a more detailed screencast of the test.

#35 - 03/06/2016 06:41 AM - Sergey Ivanovskiy

Igor Skornyakov wrote:

Sergey Ivanovskiy wrote:

Ok, understand. Please look at the next scenario (in note 17) Is this behavior correct?

- 1) Run uast/mouse/load-mouse-pointer.p
- 2) Select "WAIT" cursor from "wcur" combobox. The cursor is changed to WAIT.
- 3) Point the mouse over the "cb" combobox's scroll bar, then it displays the wait cursor but if the mouse pointer is inside its drop down list, then the cursor is arrow.

This is not correct. The cursor at the drop-down should be WAIT (both for cb and cb2). And in the trunk it works like this.

Thank you for the next screen video. In this video you didn't show my test case and simultaneously I found a misspelled "cb1" combo box in sample mode

Sorry, "cb" is a misspelled "cb1". It should be "cb1" 3) should be

3) Point the mouse over the "cb1" combobox's scroll bar, then it displays the wait cursor but if the mouse pointer is inside its drop down list, then the cursor is arrow.

I tried to produce the same video with vlc, but its resolution is not enough to display graphics cursor. I will try the tool you advised.

One remark, the current p2j trunc is not good. The Swing client is opened with incorrect sizes and may be has more unrelated issues. I think my test case is ok, because on your video if the mouse pointer is over the combobox's button, then it is changed to WAIT cursor. It is exactly what is suspected with the simple mode combobox's scroll bar.

#36 - 03/06/2016 07:04 AM - Igor Skorniyakov

Sergey Ivanovskiy wrote:

Igor Skorniyakov wrote:

Sergey Ivanovskiy wrote:

Ok, understand. Please look at the next scenario (in note 17) Is this behavior correct?

1) Run uast/mouse/load-mouse-pointer.p

2) Select "WAIT" cursor from "wcur" combobox. The cursor is changed to WAIT.

3) Point the mouse over the "cb" combobox's scroll bar, then it displays the wait cursor but if the mouse pointer is inside its drop down list, then the cursor is arrow.

This is not correct. The cursor at the drop-down should be WAIT (both for cb and cb2). And in the trunk it works like this.

Thank you for the next screen video. In this video you didn't show my test case and simultaneously I found a misspelled "cb1" combo box in sample mode

Sorry, "cb" is a misspelled "cb1". It should be "cb1" 3) should be

[...]

I tried to produce the same video with vlc, but its resolution is not enough to display graphics cursor. I will try the tool you advised.

One remark, the current p2j trunc is not good. The Swing client is opened with incorrect sizes and may be has more unrelated issues. I think my test case is ok, because on your video if the mouse pointer is over the combobox's button, then it is changed to WAIT cursor. It is exactly what is suspected with the simple mode combobox's scroll bar.

Sorry Sergey. I'm totally confused now. Please provide a screencast.

Thank you.

#37 - 03/06/2016 07:34 AM - Sergey Ivanovskiy

- File test1.mkv added

This tool is helpful. I tried to display the mouse pointer over a scroll bar and a combobox button.

#38 - 03/06/2016 07:54 AM - Igor Skornyakov

Sergey Ivanovskiy wrote:

This tool is helpful. I tried to display the mouse pointer over a scroll bar and a combobox button.

Well, now I see. When the mouse is over a button its shape is correct but when it is over the scrollbar it is not (should be arrow). It is my bug.

#39 - 03/06/2016 07:58 AM - Igor Skornyakov

BTW: you can make a screencast for a single window if in the profile you will choose "Record a single rectangle" and then "Select window" and click on the desired window. By default CTRL-R stops the recording.

#40 - 03/06/2016 08:01 AM - Sergey Ivanovskiy

BTW: you can make a screencast for a single window if in the profile you will choose "Record a single rectangle" and then "Select window" and click on the desired window. By default CTRL-R stops the recording.

Thanks, I will use it. Igor, are you planning to fix it? Can I look at this too, because if we leave the P2j version of AbstractContainer.findMouseSource, then it becomes impossible to detect the mouse source in the web client and editor's scroll bar issue can't be fixed. Do you agree to proceed with my changes or let me work on it?

#41 - 03/06/2016 08:44 AM - Igor Skornyakov

Sergey Ivanovskiy wrote:

Thanks, I will use it. Igor, are you planning to fix it? Can I look at this too, because if we leave the P2j version of AbstractContainer.findMouseSource, then it becomes impossible to detect the mouse source in the web client and editor's scroll bar issue can't be fixed. Do you agree to proceed with my changes or let me work on it?

As far As I can see there is no this bug in the web client. I'm ready to fix this bug but we have a procedure for the tasks' assignment and committing the fixes and I do not want to violate it. I'm almost sure that in the trunk the bug can be fixed w/o changing findMouseSource method. My guess is that for the combo-box the scroll bar is registered with not the same window as the selection list but I can be wrong.

#42 - 03/06/2016 10:35 AM - Sergey Ivanovskiy

- File 3013_3.txt added

Greg Shah wrote:

I like the approach (in 3013_2.txt), please go ahead.

Greg, please review the committed revision 11054. It made 1811t and p2j identical in Swing client and fixed incorrect web client in P2j.

#43 - 03/06/2016 10:54 AM - Sergey Ivanovskiy

Igor Skorniyakov wrote:

Sergey Ivanovskiy wrote:

Thanks, I will use it. Igor, are you planning to fix it? Can I look at this too, because if we leave the P2j version of AbstractContainer.findMouseSource, then it becomes impossible to detect the mouse source in the web client and editor's scroll bar issue can't be fixed. Do you agree to proceed with my changes or let me work on it?

As far As I can see there is no this bug in the web client. I'm ready to fix this bug but we have a procedure for the tasks' assignment and committing the fixes and I do not want to violate it. I'm almost sure that in the trunk the bug can be fixed w/o changing findMouseSource method. My guess is that for the combo-box the scroll bar is registered with not the same window as the selection list but I can be wrong.

Understand your position and trying to explain and to prove that this change of AbstractContainer.findMouseSource is incorrect. It impacts the web client because this function must return the minimal component that catches the mouse event.

#44 - 03/06/2016 10:57 AM - Igor Skorniyakov

Sergey Ivanovskiy wrote:

Understand your position and trying to explain and to prove that this change of AbstractContainer.findMouseSource is incorrect. It impacts the web client because this function must return the minimal component that catches the mouse event.

I do not argue with this point. The semantics you describe can looks OK. The problem is that I it seems to be not exactly correct - simply because

some widgets override this method. I'll be happy to see that this simple semantics is working.

#45 - 03/06/2016 11:21 AM - Sergey Ivanovskiy

Igor Skorniyakov wrote:

The semantics you describe can look OK. The problem is that it seems to be not exactly correct - simply because some widgets override this method. I'll be happy to see that this simple semantics is working.

There are FrameGuiImpl, ModalWindow, OverlayWindow and WindowGuiImpl classes that override it. FrameGuiImpl takes into account its title and delegates it to its scroll pane. ModalWindow, OverlayWindow and WindowGuiImpl have the similar (identical) implementations. I only move the "to check id" logic out of findMouseSource, all other important custom cursors implementation is the same as in P2j.

#46 - 03/06/2016 12:44 PM - Igor Skorniyakov

Sergey Ivanovskiy wrote:

There are FrameGuiImpl, ModalWindow, OverlayWindow and WindowGuiImpl classes that override it. FrameGuiImpl takes into account its title and delegates it to its scroll pane. ModalWindow, OverlayWindow and WindowGuiImpl have the similar (identical) implementations.

Actually in the trunk this list is different (and includes EditorGuiImpl)

#47 - 03/06/2016 12:49 PM - Sergey Ivanovskiy

- File 3013_4.txt added

Igor Skorniyakov wrote:

Sergey Ivanovskiy wrote:

There are FrameGuiImpl, ModalWindow, OverlayWindow and WindowGuiImpl classes that override it. FrameGuiImpl takes into account its title and delegates it to its scroll pane. ModalWindow, OverlayWindow and WindowGuiImpl have the similar (identical) implementations.

Actually in the trunk this list is different (and includes EditorGuiImpl)

Yes, for the editor component Web client has its editor's scroll pane, its scroll bars and the editor mouse regions with z-order according to their parent child relations. Please look at this fix for the scroll bar cursor, committed revision 11055.

#48 - 03/07/2016 11:51 AM - Greg Shah

Code Review Task Branch 1811t Revisions 11054/11055

I like the approach.

1. I prefer not to encode widget-specific logic in `SwingMouseHandler.processAction()` (references to `ScrollPaneGuiImpl`) or in `WebMouseHandler.raiseMouseEvent()` (references to `EditorGuiImpl`). I am OK moving forward with the current code but I would like you to put a TODO in both places like this:

```
// TODO: move this widget-specific logic into methods that are implemented in the widget hierarchy
//       so that we do not have direct references to widget classes in this common code
```

2. `WidgetRegistry`, `FrameGuiImpl`, `SwingMouseHandler` each need a history entry.

3. `SwingMouseHandler` needs its copyright date updated.

#49 - 03/07/2016 11:52 AM - Greg Shah

Is there anything left to do on this task?

#50 - 03/07/2016 11:57 AM - Sergey Ivanovskiy

It seems that p2j and 1811t perform `uast/mouse/load-mouse-pointer.p` similarly.

#51 - 03/07/2016 11:59 AM - Greg Shah

Good. Make the cleanup I requested in the code review.

Also, in regard to the issue from note 38 (it is still a problem, right?) please open a new bug as subtask for [#2677](#). Make Igor and myself watchers.

#52 - 03/07/2016 12:05 PM - Sergey Ivanovskiy

Also, in regard to the issue from note 38 (it is still a problem, right?) please open a new bug as subtask for [#2677](#). Make Igor and myself watchers.

No, it is fixed in 1811t.

Code Review Task Branch 1811t Revisions 11054/11055

1. I prefer not to encode widget-specific logic in `SwingMouseHandler.processAction()` (references to `ScrollPaneGuiImpl`) or in `WebMouseHandler.raiseMouseEvent()` (references to `EditorGuiImpl`). I am OK moving forward with the current code but I would like you to put a TODO in both places like this:

Do you mean to add processAction(MouseEvent e) to Widget and to implement it in subclasses?

#53 - 03/07/2016 12:13 PM - Greg Shah

Do you mean to add processAction(MouseEvent e) to Widget and to implement it in subclasses?

I'm not sure. At this point, just put the comment there and we will consider the right approach later.

#54 - 03/07/2016 01:04 PM - Sergey Ivanovskiy

Committed revision 11056 fixed history entries and added todo comments.

#55 - 03/07/2016 01:07 PM - Greg Shah

- Target version set to Milestone 12
- % Done changed from 0 to 100
- Status changed from New to Closed
- Start date deleted (03/03/2016)

It looks good. I'm closing this task.

#56 - 03/10/2016 02:31 PM - Sergey Ivanovskiy

- File test_p2j.mkv added

Igor, please look at the video of the last P2j test against ./mouse/load-mouse-pointer.p. It seems that two unknown issues have been found:

If we set the WAIT cursor to the application window and next set the CROSS cursor to the enclosed frame, then

- 1) If the mouse pointer is over labels, then it changes to the WAIT cursor.
- 2) If the mouse pointer is crossed the application window from left to right, then it doesn't change to CROSS, but if it moves from right to left, then the mouse pointer is changed to the CROSS cursor.
- 3) Unrelated: if we press CTRL + D to set the GLOVE cursor to the widget, then the focused widget is sometimes different from the widget under the mouse pointer. Thus the cursor is applied to the current focused widget.

Are these two issues correct or not?

#57 - 03/10/2016 02:57 PM - Sergey Ivanovskiy

- File current_5.txt added

Committed revision 11059 fixed the 1) issue with labels with TODO comments.

#58 - 03/10/2016 03:01 PM - Igor Skorniyakov

Sergey Ivanovskiy wrote:

Igor, please look at the video of the last P2j test against ./mouse/load-mouse-pointer.p. It seems that two unknown issues have been found:

If we set the WAIT cursor to the application window and next set the CROSS cursor to the enclosed frame, then

- 1) If the mouse pointer is over labels, then it changes to the WAIT cursor.
- 2) If the mouse pointer is crossed the application window from left to right, then it doesn't change to CROSS, but if it moves from right to left,

then the mouse pointer is changed to the CROSS cursor.

3) Unrelated: if we press CTRL + D to set the GLOVE cursor to the widget, then the focused widget is sometimes different from the widget under the mouse pointer. Thus the cursor is applied to the current focused widget.

Are these two issues correct or not?

Sergey.

I was able to see the first two issues. These are my bugs. Thanks for pointing out.

Sorry but I do not understand the last point (regarding CTRL-D).

#59 - 03/10/2016 03:07 PM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

3) Unrelated: if we press CTRL + D to set the GLOVE cursor to the widget, then the focused widget is sometimes different from the widget under the mouse pointer. Thus the cursor is applied to the current focused widget.

It relates to the focus management, it seems that sometimes I can't focus the given widget. For an example, click on **cb2** and then press CTRL+D, but the focus input are gained by **cb** combo box.

#60 - 03/10/2016 03:11 PM - Igor Skorniyakov

Sergey Ivanovskiy wrote:

It relates to the focus management, it seems that sometimes I can't focus the given widget. For an example, click on **cb2** and then press CTRL+D, but the focus input are gained by **cb** combo box.

This is a known bug with COMBO-BOX ([#2934](#)). See last section of the test plan.

#61 - 03/10/2016 06:09 PM - Sergey Ivanovskiy

Ok. Planning to check

```
Widget<?> mouseSource = findMouseSource(e);
```



```

// if the widget has changed, send EXITED/ENTERED events
if (lastHoveredWidget != null && lastHoveredWidget != mouseSource)
{
    MouseEvent exite = new MouseEvent((Component) e.getSource(),
                                     MouseEvent.MOUSE_EXITED,
                                     e.getWhen(),
                                     e.getModifiers(),
                                     e.getX(),
                                     e.getY(),
                                     e.getClickCount(),
                                     false,
                                     e.getButton());

    // lastHoveredWidget.mouseExited(exite);
    processAction(lastHoveredWidget, exite);
    // call this unconditionally, as the widget needs to be notified that it was exit...
    tc.postMouseEvent(exite, ews.getWindowId(), UiUtils.getWidgetIdAsInt(lastHoveredWidget));

    MouseEvent entere = new MouseEvent((Component) e.getSource(),
                                       MouseEvent.MOUSE_ENTERED,
                                       e.getWhen(),
                                       e.getModifiers(),
                                       e.getX(),
                                       e.getY(),
                                       e.getClickCount(),
                                       false,
                                       e.getButton());

    // mouseSource.mouseEntered(entere);
    processAction(mouseSource, entere);
    // call this unconditionally, as the widget needs to be notified that it was entered...
    tc.postMouseEvent(entere, ews.getWindowId(), UiUtils.getWidgetIdAsInt(mouseSource));

    lastHoveredWidget = mouseSource;
}
else
{
    if (!processAction(e))
    {
        SwingMouseHandler.super.mouseMoved(e);
    }
}
}

```

If we have actual bounds, then we should check if the mouse pointer is out or within lastHoveredWidget actual bounds and if the mouse pointer enters mouseSource bounds, because mouseSource is a parent of the original mouse source.

#62 - 03/10/2016 06:35 PM - Sergey Ivanovskiy

- File current_7.txt added

Committed revision 11061 fixed 2) issue from note 56.

#63 - 11/16/2016 12:12 PM - Greg Shah

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

Files			
3013_1.txt	1.91 KB	03/03/2016	Sergey Ivanovskiy
3013_2.txt	4.01 KB	03/04/2016	Sergey Ivanovskiy
mouse.mkv	650 KB	03/05/2016	Igor Skorniyakov
mouse1.mkv	827 KB	03/06/2016	Igor Skorniyakov
test1.mkv	1.53 MB	03/06/2016	Sergey Ivanovskiy
3013_3.txt	4.87 KB	03/06/2016	Sergey Ivanovskiy
3013_4.txt	1.19 KB	03/06/2016	Sergey Ivanovskiy
test_p2j.mkv	1.28 MB	03/10/2016	Sergey Ivanovskiy
current_5.txt	6.24 KB	03/10/2016	Sergey Ivanovskiy
current_7.txt	1.54 KB	03/10/2016	Sergey Ivanovskiy