User Interface - Feature #2565

Feature # 2252 (Closed): implement GUI client support

implement runtime support for the SET-WAIT-STATE method

04/30/2015 02:22 PM - Greg Shah

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

Description

Related issues:

Related to User Interface - Feature #2566: implement LOAD-MOUSE-POINTER method	**Closed**	
Related to User Interface - Bug #2858: implement support for animated mouse c...	**New**	
Related to User Interface - Feature #2628: Non-fill-in column support in browse.	**WIP**	**07/30/2015**
Related to User Interface - Bug #2977: LOAD-MOUSE-POINTER implementation for ...	**New**	
Related to User Interface - Bug #2996: Cursor change at the window border	**Closed**	
Related to User Interface - Bug #2997: Add support for pre-defined cursor typ...	**New**	**02/19/2016**

History

#1 - 11/10/2015 04:25 PM - Igor Skornyakov

Created task branch 2565a from the trunk revno 10951.

#2 - 11/11/2015 10:28 AM - Igor Skornyakov

I have some questions regarding mouse cursor support in p2j.
As far I as can see at this moment p2j supports a pre-defined set of mouse cursor shapes. For Swing this is java.awt provide cursors. For the Web GUI this is cursors known at at cleint side as only a cursor id is provided to the GuiWebSocket.

The LOAD-MOUSE-POINTER supports a number of pre-defined cusrsor shapes and allows to load the hsape from the file. I'm not sure that even pre-defined shapes have counterparts in the existing p2j ones. The support of the externally defined cursors will require substantial reworking of the existing code which is based on the CursorType enum.

The questions are:
1. Is my understanding of the cusrrent mouse cursor in p2j correct or I've missed something?
2. If I undersand the suruation correctly is a rework of the mouse cursor support in the scope of this task (actially this one and #2566 aswe agreed that these tasks will be done in the same branch)?

Thank you.

#3 - 11/11/2015 12:49 PM - Greg Shah

> As far I as can see at this moment p2j supports a pre-defined set of mouse cursor shapes.

Yes. So far we have provided support for those shapes that are implicitly used by an application, in absence of any explicit LOAD-MOUSE-POINTER() method calls.

> For Swing this is java.awt provide cursors.

Yes, we currently rely upon java.awt.Cursor for the Swing client.

> For the Web GUI this is cursors known at at cleint side as only a cursor id is provided to the GuiWebSocket.

Kind of. We send the ordinal of the CursorType down to the JS (javascript) client which is running in the browser. There is code in com/goldencode/p2j/ui/client/driver/web/res/p2j.sockets.js to process the 0x94 (MSG_SET_CURSOR_STYLE) message. It calls p2j.screen.setCursorStyle(styleId, wid); as a result with that ordinal. The resulting setCursorStyle() in com/goldencode/p2j/ui/client/gui/driver/web/res/p2j.screen.js will translate the ordinal into a known CSS style value for each predefined CursorType value. Then it calls Window.setCursorStyle() (also found in p2j.screen.js) to actually set the cursor for the window's backing canvas element. It uses the CSS style value like this: this.canvas.style.cursor = style;. So we are mapping these into browser-specific well-known CSS cursors. This is very similar to the approach with using predefined Swing cursors.

> The LOAD-MOUSE-POINTER supports a number of pre-defined cusrsor shapes and allows to load the hsape from the file.

Yes it does this. But it actually does more than this. Supposedly, it also registers the cursor such that when the mouse crosses over the widget (and any contained widgets that don't have their own custom cursor registered), the cursor will change to the loaded cursor (predefined or otherwise).

Before you go further with the analysis, it is important that you write testcases to prove/disprove all of the points that are made in the Progress documentation for this method. Please make a list of those "assertions" and post it here. I see several other questions that they do not address:

- When the mouse leaves the widget, what does it switch back to? Is it always reliable (e.g. whatever it was set to when the mouse entered)? Or is there some situation in which the mouse can be changed to something else based on the context?
- How does this interact with the SET-WAIT-STATE which also modifies the cursor? Which one "wins"?
- How does it handle unknown value as a parameter?
- Is there a way to disable or remove the registered pointer? The only implied way is to set it back to the predefined default cursor, I guess.

You can and should add other questions to the list that I have not considered above.

In #1801 you spread your results over many, many entries. In addition, I had to followup on many open questions to ask for more detailed results. I expect that you now understand that we need full results. I want the results in a single history entry or at least in a very small number that can be easily understood. The results should answer all questions without me checking and asking for clarifications. It will help to make the full list of questions as I have asked above. I will review those and add more if I can think of any more. Then make sure your testcases have enough variations to cover all needed answers. Finally, when no questions remain unanswered, document it here in an easily consumed manner.

I hope you can see how important this part of the work is. Invariably, there is always something different or surprising that Progress does which deviates from the documentation. Sometimes this is bigger and sometimes smaller. But it almost never occurs that the documentation is both comprehensive and fully accurate. The implementation is usually straightforward if you do the right job on the testcases.

> I'm not sure that even pre-defined shapes have counterparts in the existing p2j ones.

I can tell you that we don't have all counterparts currently supported. But I suspect that all the predefined cursors are probably already available in Swing or CSS. You'll have to check this and extend our support to cover the full range that Progress exposes as "predefined".

> The support of the externally defined cursors will require substantial reworking of the existing code which is based on the CursorType enum.

Actually, I don't think so. We can add the additional predefined cursors and extend the current approach slightly to support them. That will be pretty easy I think.

For the "load from file" case, just add a special enum type "CUSTOM".

We already support loading images. The custom cursor case would reply upon our current image loading capabilities. Please see #2479 and #2792 to get a better understanding of how this works. Eugenie can answer questions you may have. You may have to extend our support to ensure it handles the Windows cursor bitmap types (supposedly these are .cur and .ani but who knows what is really possible?). But generally, it seems like the core functionality for loading is already there. We already support using the server-side application JAR file first and then using the client's filesystem including searching the propath for relative names. This JAR support allows us to add any image resources (now including cursors) to the converted application's jar file and to leverage those so that they don't need to be copied into the client's filesystem. But the converted code is the same. Please note that our image loading code has portions on both the server (jar stuff) and client (filesystem/propath) to implement.

So the backing code for LOAD-MOUSE-POINTER() would either reference a predefined CursorType or it would load a custom cursor and use the CursorType.CUSTOM. These would be passed to some backing functionality in the client-side widget hierarchy to register the cursor for "flyover" processing. I imagine we could have 2 overloaded methods for this:

public void registerCursor(CursorType)
public void registerCursor(ImageWrapper) (in this case CursorType.CUSTOM is implicit)

Constantin: would you please summarize what implementing this flyover processing will have to do? In particular, we must setup the mouse processing to detect this case and switch cursors to (on entry) and from (on exit). The full set of behavior will be known once the testcases are done and all questioned answered.

Most of the functionality for this should be done in generic GUI client code. However, there will need to be a driver-level registration of this "flyover" event (and the CursorType plus the optional image). The web-client in particular will be doing this flyover processing on the JS side only, once it is registered. Otherwise it would be costly. This means that the image is only ever sent once and it is just used when needed. Both the JS and Swing clients will need code added to honor the custom cursor.

> Is my understanding of the cusrrent mouse cursor in p2j correct or I've missed something?
> If I undersand the suruation correctly is a rework of the mouse cursor support in the scope of this task (actially this one and #2566 aswe agreed that these tasks will be done in the same branch)?

I hope I've made things clear.

#4 - 11/11/2015 01:10 PM - Igor Skornyakov

Greg Shah wrote:

> I hope I've made things clear.

Yes, thank you very much.

#5 - 11/11/2015 01:48 PM - Constantin Asofiei

Greg Shah wrote:

> So the backing code for LOAD-MOUSE-POINTER() would either reference a predefined CursorType or it would load a custom cursor and use the CursorType.CUSTOM. These would be passed to some backing functionality in the client-side widget hierarchy to register the cursor for "flyover" processing. I imagine we could have 2 overloaded methods for this:
>
> public void registerCursor(CursorType)
> public void registerCursor(ImageWrapper) (in this case CursorType.CUSTOM is implicit)
>
> Constantin: would you please summarize what implementing this flyover processing will have to do? In particular, we must setup the mouse processing to detect this case and switch cursors to (on entry) and from (on exit). The full set of behavior will be known once the testcases are done and all questioned answered.

I think it all reduces to making the current support for editable widgets (FILL-IN and EDITOR, where the cursor is set to text cursor - IBEAM in 4GL's table) generic, so that when the cursor enters the boundaries of a widget, it changes to that widget's cursor (if set).

On Swing side, we have the gui.driver.swing.EditableWidget class. On JS side, there is the MouseEditable class in p2j.mouse.js. These can be renamed (maybe to HoverableWidget?) and refactored to work with a specific cursor type. For custom cases, it will refer to the cursor by the image name (this needs testing if it requires only the image name, or the exact pointer-name specified to load-mouse-pointer).

Currently, the widget's editable cursor is set via GuiDriver.registerEditableWidget - we should rename this to registerCursor; beside the cursor/image, it should also receive the widget's ID. This in turn will create an HoverableWidget instance, which will save the widget's ID and cursor type, and do the same work currently done by EditableWidget (i.e. change cursor type after the mouse enters the boundaries, restore it after it exits).

For the JS side, the current usage of p2j.screen.setCursorStyle I think is becoming obsolete - for the editable widget and window border part, all work is already done on JS side.

Igor, about current "editable" support: please check what MOUSE-POINTER attribute returns for FILL-IN and EDITOR, and also for a BUTTON (or other widgets) for which LOAD-MOUSE-POINTER was not called. This should be either IBEAM (for FILL-IN and EDITOR)/ARROW (for BUTTON) or unknown/empty? value. In any case, I think we need to save the widget's cursor type in BaseConfig, as in P2J the widget is registered for "flyover" events and cursor changed only if it becomes editable; when the widget gets disabled, it is de-registered via GuiDriver.deregisterWidget. Current usage of GuiDriver.registerEditableWidget needs to be reviewed and made generic (a similar approach to how TOOLTIP was solved for all widgets).

#6 - 11/11/2015 01:57 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, about current "editable" support: please check what MOUSE-POINTER attribute returns for FILL-IN and EDITOR, and also for a BUTTON (or other widgets) for which LOAD-MOUSE-POINTER was not called. This should be either IBEAM (for FILL-IN and EDITOR)/ARROW (for BUTTON) or unknown/empty? value. In any case, I think we need to save the widget's cursor type in BaseConfig, as in P2J the widget is registered for "flyover" events and cursor changed only if it becomes editable; when the widget gets disabled, it is de-registered via GuiDriver.deregisterWidget. Current usage of GuiDriver.registerEditableWidget needs to be reviewed and made generic (a similar approach to how TOOLTIP was solved for all widgets).

Thank you very much Constantin!

#7 - 11/11/2015 01:58 PM - Greg Shah

> For custom cases, it will refer to the cursor by the image name (this needs testing if it requires only the image name, or the exact pointer-name specified to load-mouse-pointer).

I don't want the client code to know about the actual image name or filename. Let that be handled one time in the load process. Once loaded, we can just use the ImageWrapper instance. Is there some reason we can't do this?

In the load process, we would want to avoid re-loading anything already loaded. But we may already do that caching in our image support, I'm not sure.

#8 - 11/11/2015 02:00 PM - Greg Shah

> For the JS side, the current usage of p2j.screen.setCursorStyle I think is becoming obsolete - for the editable widget and window border part, all work is already done on JS side.

We still may need it for SET-WAIT-STATE().

#9 - 11/11/2015 02:04 PM - Constantin Asofiei

Greg Shah wrote:

> > For the JS side, the current usage of p2j.screen.setCursorStyle I think is becoming obsolete - for the editable widget and window border part, all work is already done on JS side.
>
> We still may need it for SET-WAIT-STATE().

Ah, you are correct here, this is an explicit cursor set.

Greg Shah wrote:

> > For custom cases, it will refer to the cursor by the image name (this needs testing if it requires only the image name, or the exact pointer-name specified to load-mouse-pointer).
>
> I don't want the client code to know about the actual image name or filename. Let that be handled one time in the load process. Once loaded, we can just use the ImageWrapper instance. Is there some reason we can't do this?

Ok, this makes sense. As I recall, the ImageWrapper is responsible for this.

#10 - 11/12/2015 05:36 AM - Igor Skornyakov

- File predefined-mouse-cursors.p added

I see something strange with LOAD-MOUSE-POINTER() method on windev01 (see attached program):
1. Initial value of the MOUSE-POINTER attribute both for FRAME and FILL-IN is an empty string.
2. LOAD-MOUSE-POINTER() method fails for "APPSTARTING", "RECTANGLE", and "COMPILER-WAIT". The error message is 4487 "Unable to open file for FILL-IN ed". The return value is no
3. For all other predefined cursor types the LOAD-MOUSE-POINTER() method returns yes, the MOUSE-POINTER attribute changes its value but I see no visual effect at all.

Any ideas?
Thank you.

#11 - 11/12/2015 06:07 AM - Constantin Asofiei

Igor Skornyakov wrote:

> I see something strange with LOAD-MOUSE-POINTER() method on windev01 (see attached program):
> 1. Initial value of the MOUSE-POINTER attribute both for FRAME and FILL-IN is an empty string.

Please check all widgets for this implicit value. Also, LOAD-MOUSE-POINTER needs to be checked with all widget types.

> 2. LOAD-MOUSE-POINTER() method fails for "APPSTARTING", "RECTANGLE", and "COMPILER-WAIT". The error message is 4487 "Unable to open file for FILL-IN ed". The return value is no

RECTANGLE/COMPILER-WAIT/APPSTARTING work with BUTTON/FILL-IN but RECTANGLE is a no-op (the mouse pointer doesn't change when hovering the widget). In your test, the problem is this: the combo-box has a format of x(8), and RECTANGLE is cut to RECTANGL - to match the format (same for the other names). So you need to change the test so that the cursor name you use in LOAD-MOUSE-POINTER is the correct value. The error message you are seeing is shown if LOAD-MOUSE-POINTER is passed an incorrect pointer name.

> 3. For all other predefined cursor types the LOAD-MOUSE-POINTER() method returns yes, the MOUSE-POINTER attribute changes its value but I see no visual effect at all.

The visual effect can be seen after you move the mouse pointer so that it hovers the fill-in (the widget on which LOAD-MOUSE-POINTER was called).

#12 - 11/12/2015 06:16 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Please check all widgets for this implicit value. Also, LOAD-MOUSE-POINTER needs to be checked with all widget types.

I understand this and will check.

> RECTANGLE/COMPILER-WAIT/APPSTARTING work with BUTTON/FILL-IN but RECTANGLE is a no-op (the mouse pointer doesn't change when hovering the widget). In your test, the problem is this: the combo-box has a format of x(8), and RECTANGLE is cut to RECTANGL - to match the format (same for the other names). So you need to change the test so that the cursor name you use in LOAD-MOUSE-POINTER is the correct value. The error message you are seeing is shown if LOAD-MOUSE-POINTER is passed an incorrect pointer name.

Thank you Constantin. After changing default format to "X(20)" all predefined types are accepted.

>> 3. For all other predefined cursor types the LOAD-MOUSE-POINTER() method returns yes, the MOUSE-POINTER attribute changes its value but I see no visual effect at all.

> The visual effect can be seen after you move the mouse pointer so that it hovers the fill-in (the widget on which LOAD-MOUSE-POINTER was called).

I understand this. However when I hover the widget I see tooltip but the mouse cursor shape doesn't change.

#13 - 11/12/2015 06:20 AM - Constantin Asofiei

Igor Skornyakov wrote:

>> RECTANGLE/COMPILER-WAIT/APPSTARTING work with BUTTON/FILL-IN but RECTANGLE is a no-op (the mouse pointer doesn't change when hovering the widget). In your test, the problem is this: the combo-box has a format of x(8), and RECTANGLE is cut to RECTANGL - to match the format (same for the other names). So you need to change the test so that the cursor name you use in LOAD-MOUSE-POINTER is the correct value. The error message you are seeing is shown if LOAD-MOUSE-POINTER is passed an incorrect pointer name.

> Thank you Constantin. After changing default format to "X(20)" all predefined types are accepted.

Please check also how LOAD-MOUSE-POINTER() treats leading or trailing spaces - is the passed text trimmed or the text is used as is?

>>> 3. For all other predefined cursor types the LOAD-MOUSE-POINTER() method returns yes, the MOUSE-POINTER attribute changes its value but I see no visual effect at all.

>> The visual effect can be seen after you move the mouse pointer so that it hovers the fill-in (the widget on which LOAD-MOUSE-POINTER was called).

> I understand this. However when I hover the widget I see tooltip but the mouse cursor shape doesn't change.

I don't understand - for me the test you posted works fine; what tooltip are you referring to? Maybe if there is a tooltip the explicit mouse pointer is ignored?

#14 - 11/12/2015 06:25 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor Skornyakov wrote:
> Please check also how LOAD-MOUSE-POINTER() treats leading or trailing spaces - is the passed text trimmed or the text is used as is?

OK, thank you.

>> I understand this. However when I hover the widget I see tooltip but the mouse cursor shape doesn't change.

> I don't understand - for me the test you posted works fine; what tooltip are you referring to? Maybe if there is a tooltip the explicit mouse pointer is ignored?

No. For me the effect is the same regardless of the tooltip. May be it is a problem with RDP client? Which one you use?
Thank you.

#15 - 11/12/2015 06:27 AM - Igor Skornyakov

Indeed. With FreeRDP I can see the effect! So the problem was with Remmina.

#16 - 11/12/2015 07:33 AM - Greg Shah

> Please make a list of those "assertions" and post it here. I see several other questions that they do not address:
> ...
> You can and should add other questions to the list that I have not considered above.

Please comply with this request **before** you write all your testcases. In other words, I expect it to be done as your first step. The point here is to ensure that there is a single complete list of questions to answer. We can discuss that list and add things as needed. Then you should have a pretty good target for what the testcases need to do.

#17 - 11/12/2015 07:41 AM - Igor Skornyakov

Greg Shah wrote:

> Please comply with this request **before** you write all your testcases. In other words, I expect it to be done as your first step. The point here is to ensure that there is a single complete list of questions to answer. We can discuss that list and add things as needed. Then you should have a pretty good target for what the testcases need to do.

Greg. I can try to do this but some questions come to my mind only when I perform some tests. I'm preparing now a list of all my findings to post them all at once. If some new questions will arise after that I will update this list and post a new version. Is it OK? Sorry, but I do not think that I can create a complete list of assertions upfront.
Thank you.

#18 - 11/12/2015 07:50 AM - Greg Shah

It is expected that as you go, you will find other questions that were not considered up front. Likewise, you see that Constantin requested some additions in the last exchange.

I am not saying that the list has to be final. Just that there must be a list. Usually you start by making a list of the assertions of behavior that are made in the Progress documentation. Break that out into a list that can be checked for correctness and comprehensiveness. Then add any other items based on your experience so far, that are often an area of boundary/error conditions AND which are NOT usually well specified in the documentation. A good example is how unknown value or other unexpected data is handled as parameters. Again, note that my list above in note 3 and Constantin's points in note 13 already suggest some of these items (unknown value, untrimmed space before and/or after the string...).

I'm asking you to do your best to make the list now.

Then work your way through the testcases to determine the answer to each question. If you find more questions, edit the note that has the list and add them there. That way there will be a single place to look for the list.

If you don't do this, then your testcases will almost always be incomplete because you had no plan in what questions you are answering.

I am asking you to work this way, because we have found that it is the most effective way to do this.

#19 - 11/12/2015 07:53 AM - Igor Skornyakov

Greg Shah wrote:

> I am asking you to work this way, because we have found that it is the most effective way to do this.

Got it. Thank you.

#20 - 11/12/2015 07:59 AM - Igor Skornyakov

1. LOAD-MOUSE-POINTER removes trailing spaces for predefined cursor shapes (the MOUSE-POINTER value doesn't contain them). Leading spaces are neither trimmed nor ignored. The RECTANGLE cursor type is stored as the MOUSE-POINTER attribute value but it doesn't affect the cursor shape.
2. For filename argument trailing spaces are ignored but are retained as the MOUSE-POINTER value). Leading spaces are neither trimmed nor ignored. The file name is stored "as-is". If it was found using PROPATH the original filename is stored as a MOUSE-POINTER value.
3. The new mouse cursor value takes effect immediately.
3. The initial value if the MOUSE-POINTER attribute is an empty string. However the default mouse cursor is different for different widgets (see below).
4. If the argument of the LOAD-MOUSE-POINTER method is UNKNOWN or empty string then ite returns "yes", the MOUSE-POINTER value becomes empty string and the mouse cursor becomes a default one.
5. When the mouse leaves the widget the cursor takes shape of the current MOUSE-POINTER of the e.g. enclosing FRAME which can be different from the one which was in effect when the mouse entered the widget.

Done When the mouse leaves the widget, what does it switch back to? Is it always reliable (e.g. whatever it was set to when the mouse entered)? Or is there some situation in which the mouse can be changed to something else based on the context?

Todo How does this interact with the SET-WAIT-STATE which also modifies the cursor? Which one "wins"?

Done How does it handle unknown value as a parameter?

Done Is there a way to disable or remove the registered pointer? The only implied way is to set it back to the predefined default cursor, I guess.

Todo Check the default value and cursor shape for all widgets

Todo Find java.awt counterparts for all predefined cursor types.

#21 - 11/12/2015 08:08 AM - Greg Shah

I have two concerns.

1. You are mixing questions and answers together in a way that makes it hard to see the list of questions.

2. Remember that we must doubt all documentation by Progress. Thus, you must prove that the following statements are true or not AND if they are complete or not. These are the **assertions** that I was talking about. You are ignoring my request to test all Progress assertions. I will show you examples from the documentation:

```
If you apply this method to a frame, field group, or window, the same mouse pointer is displayed when it is moved across all child widgets within the frame, field group, or window.
```

```
However, if you load a different mouse pointer for a child widget, the child widget mouse pointer is displayed when it is moved over that child.
```

```
ABL provides a collection of mouse pointers that you can use in graphical applications. Table 85 names and des
```

```
cribes each mouse pointer in the collection.
```

...

Hopefully, you get the idea. Create questions to prove or disprove each part of the documentation.

#22 - 11/12/2015 08:14 AM - Igor Skornyakov

Greg Shah wrote:

> Hopefully, you get the idea. Create questions to prove or disprove each part of the documentation.

OK. I have two quistions about it.
1. Do I understand correctly that there should be two notes: one with question and another one with answers and other finding?
2. How the watchers will be notified about new questions/answers/findings?
Thank you.

#23 - 11/12/2015 08:29 AM - Greg Shah

> 1. Do I understand correctly that there should be two notes: one with question and another one with answers and other finding?

Yes. The first note is just questions. The second note with the answers and other findings, should repeat the questions. For example:

```
1. What do we do about blah blah blah?

This is the answer about blah blah blah.

2. How does yatta yatta work?

Yatta yatta does this and that.
```

> 2. How the watchers will be notified about new questions/answers/findings?

You can always send us an email if there is any "later edit" that you believe we should see.

#24 - 11/12/2015 08:34 AM - Igor Skornyakov

Greg Shah wrote:

> Yes. The first note is just questions. The second note with the answers and other findings, should repeat the questions. For example:

I see. Thnak you.

> You can always send us an email if there is any "later edit" that you believe we should see.

May be it is better just to add an update note? This will guarantee that all watchers will be notified and automatically provide a familiar subject.

#25 - 11/12/2015 08:50 AM - Greg Shah

> May be it is better just to add an update note? This will guarantee that all watchers will be notified and automatically provide a familiar subject.

This sounds reasonable.

#26 - 11/12/2015 10:04 AM - Igor Skornyakov

QUESTIONS:
1. Check the following statements from the Progress documentation:
1a. LOAD-MOUSE-POINTER specifies the mouse pointer to display when the pointer is moved over the widget. If you apply this method to a frame, field group, or window, the same mouse pointer is displayed when it is moved across all child widgets within the frame, field group, or window. However, if you load a different mouse pointer for a child widget, the child widget mouse pointer is displayed when it is moved over that child.

1b. If the mouse pointer is loaded successfully, the method returns TRUE.

1c. In addition to the mouse pointers that ABL supplies, you can also use a bitmap that you supply that is in the form of a Windows cursor (.cur or .ani) file. To use such a bitmap, substitute the name of the Windows cursor file for pointer-name.

1d. For browse columns, if you do not specify a mouse pointer, the AVM uses the mouse pointer the user specified for the browse.

1e. In Windows, you can specify a URL pathname. If you specify a fully-qualified URL, LOAD-MOUSE-POINTER( ) loads the pointer file directly without searching directories or URLs in PROPATH. Valid URL protocols include HTTP and HTTPS.

1f. URL pathnames cannot contain the percent symbol (%). If an error exists in a URL specified on the PROPATH, the LOAD-MOUSE-POINTER method continues searching with the next PROPATH entry.

1g. Check if .ico files are supported for the LOAD-MOUSE-POINTER method.

1h. How is the cursor hotspot determined when loading an .ico as a pointer?

2. When the mouse leaves the widget, what does it switch back to? Is it always reliable (e.g. whatever it was set to when the mouse entered)? Or is there some situation in which the mouse can be changed to something else based on the context?

3. How does this interact with the SET-WAIT-STATE which also modifies the cursor? Which one "wins"?

4. How does it handle unknown value as a parameter?

5. Is there a way to disable or remove the registered pointer? The only implied way is to set it back to the predefined default cursor, I guess.

6. Check also how LOAD-MOUSE-POINTER() treats leading or trailing spaces - is the passed text trimmed or the text is used as is?

7. Check the default value and cursor shape for all widgets

8. Find java.awt counterparts for all predefined cursor types.

9. Analyse and describe mouse cursor behaviour for the BROWSE widget.

#27 - 11/12/2015 10:16 AM - Igor Skornyakov

ANSWERS:

Q1a. LOAD-MOUSE-POINTER specifies the mouse pointer to display when the pointer is moved over the widget. If you apply this method to a frame, field group, or window, the same mouse pointer is displayed when it is moved across all child widgets within the frame, field group, or window. However, if you load a different mouse pointer for a child widget, the child widget mouse pointer is displayed when it is moved over that child.

A: This seems to be correct in general. However if the widget is not enabled its mouse cursor is inherited from the enclosing frame (doesn't change when mouse hovers over the widget).
The WINDOW:MOUSE-POINTER is inherited by all its frames (including the embedded ones) and widgets within them unless it is explicitly set via LOAD-MOUSE-POINTER.
The FRAME:MOUSE-POINTER is **not** inherited by BROWSE widgets within and its embedded frames and widgets within them.

Q1b. If the mouse pointer is loaded successfully, the method returns TRUE.

A: This is correct. The MOUSE-POINTER attribute takes the value of the argument (trailing spaces are removed in the case of pre-defined cursor shape). In the case of failure the error "Unable to open file for <Widget type> <Widget name> (4487)" is reported and the method returns no

Q1c. In addition to the mouse pointers that ABL supplies, you can also use a bitmap that you supply that is in the form of a Windows cursor (.cur or .ani) file. To use such a bitmap, substitute the name of the Windows cursor file for pointer-name.

A: Indeed, both .cur and .ani files are supported.

Q1d. For browse columns, if you do not specify a mouse pointer, the AVM uses the mouse pointer the user specified for the browse.

A: See **Q9**.

Q1e. In Windows, you can specify a URL pathname. If you specify a fully-qualified URL, LOAD-MOUSE-POINTER( ) loads the pointer file directly without searching directories or URLs in PROPATH. Valid URL protocols include HTTP and HTTPS.

A: LOAD-MOUSE-POINTER() indeed loads cursors images from the directories in PROPATH. It also loads images via http(s). 4GL validates the https server certificate and asks for a confirmation if it is not trusted each time it scans PROPATH or if https URL is specified as an argument for the LOAD-MOUSE-POINTER() method.

Q1f. URL pathnames cannot contain the percent symbol (%). If an error exists in a URL specified on the PROPATH, the LOAD-MOUSE-POINTER method continues searching with the next PROPATH entry.

A: This is correct. 4GL validates the https server certificate and asks for a confirmation if it is not trusted each time it scans PROPATH. Invalid URLs in the PROPATH are silently ignored.

Q1g. Check if .ico files are supported for the LOAD-MOUSE-POINTER method.

A: .ico files are supported as well (see uast/mouse/all-editable-widgets.p)

Q1h. How is the cursor hotspot determined when loading an .ico as a pointer?

A: As https://msdn.microsoft.com/en-us/library/0b1674x8.aspx claims it **is** possible to specify cursor hotspot for the .ico file. I've tried some free icons from http://www.iconarchive.com/tag/hand-cursor and for at least Iconshock-Real-Vista-Construction-Hand-driller.ico the hotspot looks to be not a default one (upper-left corner). The GIMP editor doesn't support hotspot-related data and I do not have VS Image Editor. I will try to find another icon editor (for Windows or Linux) which supports hotspot setting (or at least viewing it).

Q2. When the mouse leaves the widget, what does it switch back to? Is it always reliable (e.g. whatever it was set to when the mouse entered)? Or is there some situation in which the mouse can be changed to something else based on the context?

A: When the mouse leaves the widget the cursor takes shape of the current MOUSE-POINTER of the e.g. enclosing FRAME which can be different from the one which was in effect when the mouse entered the widget. The new mouse cursor shape takes effect immediately (not only at the next mouse enter).

Q3. How does this interact with the SET-WAIT-STATE which also modifies the cursor? Which one "wins"?

A: The cursor specified via SET-WAIT-STATE takes precedence.

Q4. How does it handle unknown value as a parameter?

A: If the argument of the LOAD-MOUSE-POINTER method is UNKNOWN or empty string then ite returns "yes", the MOUSE-POINTER value becomes empty string and the mouse cursor becomes a default one.

Q5. Is there a way to disable or remove the registered pointer? The only implied way is to set it back to the predefined default cursor, I guess.

A: See Q4.

Q6. Check also how LOAD-MOUSE-POINTER treats leading or trailing spaces - is the passed text trimmed or the text is used as is?

A:
- LOAD-MOUSE-POINTER removes trailing spaces for predefined cursor shapes (the MOUSE-POINTER value doesn't contain them). Leading spaces are neither trimmed nor ignored. The RECTANGLE cursor type is stored as the MOUSE-POINTER attribute value but it doesn't affect the cursor shape.
- For filename argument trailing spaces are ignored but are retained as the MOUSE-POINTER value). Leading spaces are neither trimmed nor ignored. The file name is stored "as-is". If it was found using PROPATH the original filename is stored as a MOUSE-POINTER value.

Q7. Check the default value and cursor shape for all widgets

A: The default value of the MOUSE-POINTER attribute for all widgets is empty string. The default cursor shape is IBEAM for the EDITOR and FILL-IN and ARROW for all other widgets.
For the COMBO-BOX the default cursor shape is ARROW for the DROP-DOWN type. For SIMPLE and DROP-DOWN-LIST the cursor shape is IBEAM while mouse hovers over the value part and ARROW while it hovers over the selection part.

Q8. Find java.awt counterparts for all predefined cursor types.

A:

```
APPSTARTING         No counterpart (a combination of DEFAULT_CURSOR and WAIT_CURSOR)
ARROW               DEFAULT_CURSOR
CROSS               CROSSHAIR_CURSOR
HELP                No counterpart
IBEAM               TEXT_CURSOR
NO                  No counterpart
RECTANGLE           No counterpart, not required
SIZE                MOVE_CURSOR
SIZE-E              E_RESIZE_CURSOR
SIZE-N              N_RESIZE_CURSOR
SIZE-NE             NE_RESIZE_CURSOR
SIZE-NW             NW_RESIZE_CURSOR
SIZE-S              S_RESIZE_CURSOR
SIZE-SE             SE_RESIZE_CURSOR
SIZE-SW             SW_RESIZE_CURSOR
SIZE-W              W_RESIZE_CURSOR
UPARROW             No counterpart (can be easily derived from the N_RESIZE_CURSOR or S_RESIZE_CURSOR by removing one arrowhead)
WAIT                WAIT_CURSOR
GLOVE               HAND_CURSOR
COMPILER-WAIT       No counterpart
```

Q9. Analyse and describe mouse cursor behaviour for the BROWSE widget.

A:
1. When mouse hovers over the BROWSE widget the cursor shape is always "ARROW" (regardless of the MOUSE-POINTER specified for the widget itself or enclosing FRAME or WINDOW). After the cell goes to an editing mode the cursor shape becomes "IBEAM" for FILL-IN.
2. If the MOUSE-POINTER is set for the BROWSE widget and is not set for its columns the cursor shape is visible while any mouse button is pressed on the widget title, column head or cell. After the cell goes to an editing mode the cursor shape becomes "IBEAM" for FILL-IN and "ARROW" for the COMBO-BOX and TOGGLE-BOX
3. If the MOUSE-POINTER is set both for the BROWSE widget and its column the column cursor is visible while any mouse button is pressed on the column head or cell. After the cell goes to an editing mode the cursor shape becomes "IBEAM" for FILL-IN and "ARROW" for the COMBO-BOX and TOGGLe_BOX.
If any mouse button is pressed on the widget title the the cursor shape is the one specified for the widget or it is the column cursor if the column or its cell is selected.

ADDITIONAL NOTES:
N1. Pre-defined pointers names are case-insensitive
N2. While the COMBO-BOX (DROP-DOWN/DROP-DOWN-LIST) is active its custom cursor (if defined) is used globally (in all windows) even while the mouse pointer is outside its bound.
N3. For the SIMPLE COMBO-BOX the custom cursor is used only at the border of the main part. It is always IBEAM in the main part and ARROW in the selection part.
N4. When the DROP-DOWN-LIST COMBO-BOX is not active the custom cursor is used only at the border of the main part and is always IBEAM inside the main part.
N5. If the mouse pointer is once loaded from the file any subsequent loads from the same **file** (as long as it exists) will reuse the already loaded data.
N6. If the extension is not specified for the mouse pointer file then the trailing spaces are removed and the **.bmp**, **.ico**, **.cur** and empty extensions (in this order) are appended and tried (checked with ProcessMonitor utility). Please note however that **.bmp** files seem to be considered as illegal and the LOAD-MOUSE-POINTER operation fails if such a file is found.
N7. The detailed description of the mouse cursor behavior for the COMBO-BOX:

			LOAD-MOUSE-CURSOR	
STYLE	PART	NONE	PARENT with CUSTOM	CUSTOM
DROP-DOWN-LIST	MAIN	ARROW	INHERITED	CUSTOM
DROP-DOWN-LIST	SELECTION-LIST	ARROW	INHERITED	CUSTOM
DROP-DOWN-LIST	MAIN(DROP-DOWN active)	ARROW	INHERITED	CUSTOM
SIMPLE	MAIN	TEXT	TEXT	TEXT
SIMPLE	MAIN-BORDER	TEXT	INHERITED	CUSTOM
SIMPLE	SELECTION-LIST	ARROW	ARROW	ARROW
DROP-DOWN	MAIN	TEXT	TEXT	TEXT
DROP-DOWN	MAIN-BORDER	TEXT	INHERITED	CUSTOM
DROP-DOWN	SELECTION-LIST	ARROW	INHERITED	CUSTOM
DROP-DOWN	MAIN(DROP-DOWN active)	ARROW	INHERITED	CUSTOM

#28 - 11/12/2015 10:49 AM - Igor Skornyakov

Section 27 has been updated.

#29 - 11/12/2015 11:11 AM - Greg Shah

> Valid URL protocols include HTTP and HTTPS.

When implementing, do not worry about this feature. We don't support it right now and I don't think we need to add it for the current project.

#30 - 11/12/2015 01:05 PM - Igor Skornyakov

Sections 26 and 27 have been updated.

#31 - 11/12/2015 01:36 PM - Igor Skornyakov

Section 27 has been updated.

Tests committed to uast/mouse revno 1428.

#32 - 11/13/2015 05:09 AM - Igor Skornyakov

What is the right way to test PROPATH - related features?

Even if I have sufficient rights on windev01 to change the progress.ini file and run the ini2reg utility this seems to be not safe and can affect other users.
Please advise.
Thank you.

#33 - 11/13/2015 05:35 AM - Igor Skornyakov

Section 27 has been updated.

#34 - 11/13/2015 07:50 AM - Greg Shah

Igor Skornyakov wrote:

> What is the right way to test PROPATH - related features?
>
> Even if I have sufficient rights on windev01 to change the progress.ini file and run the ini2reg utility this seems to be not safe and can affect other users.
> Please advise.
> Thank you.

The PROPATH can be changed for a single process. It is most often set as an environment variable. You can also assign it from 4GL code. There is no need to change the registry.

#35 - 11/13/2015 07:52 AM - Igor Skornyakov

Greg Shah wrote:

> The PROPATH can be changed for a single process. It is most often set as an environment variable. You can also assign it from 4GL code. There is no need to change the registry.

OK. I will try. Thank you very much!

#36 - 11/13/2015 08:02 AM - Greg Shah

In WIN32, the default way to load a cursor is with a .cur or .ani file. However, based on this:

https://msdn.microsoft.com/en-us/library/windows/desktop/ms648379%28v=vs.85%29.aspx

I think it may also be possible to load from an .ico:

```
You can also create a custom cursor at run time by using the @CreateIconIndirect@ function, which creates a cursor based on the content of an @ICONINFO@ structure. The @GetIconInfo@ function fills this structure with hot spot coordinates and information concerning the associated mask and color.
```

Please check if Progress supports .ico files (add this as a question 1g in note 36).

#37 - 11/13/2015 08:21 AM - Igor Skornyakov

Section 26 has been updated.

#38 - 11/13/2015 10:12 AM - Igor Skornyakov

Section 27 has been updated.

#39 - 11/13/2015 10:49 AM - Igor Skornyakov

Section 27 has been updated.

#40 - 11/13/2015 11:00 AM - Greg Shah

Please add a question 1h: How is the cursor hotspot determined when loading an .ico as a pointer?

The .cur and .ani naturally have a hotspot encoded in the format. I didn't think that .ico has the same facility.

#41 - 11/13/2015 11:22 AM - Igor Skornyakov

Regarding the predefined mouse cursors which curently do not have counterparts. Should I create them myself or can find free images in the Internet (I'm a VERY bad designer).

Thank you.

#42 - 11/13/2015 11:23 AM - Igor Skornyakov

Greg Shah wrote:

> Please add a question 1h: How is the cursor hotspot determined when loading an .ico as a pointer?
>
> The .cur and .ani naturally have a hotspot encoded in the format. I didn't think that .ico has the same facility.

Done

#43 - 11/13/2015 11:26 AM - Greg Shah

> Regarding the predefined mouse cursors which curently do not have counterparts. Should I create them myself or can find free images in the Internet (I'm a VERY bad designer).

Before we determine this, I'd like to know the list of cursors which have no counterpart.

#44 - 11/13/2015 11:39 AM - Igor Skornyakov

Greg Shah wrote:

> Before we determine this, I'd like to know the list of cursors which have no counterpart.

I think it is at least "UPARROW", "HELP", "NO" and "COMPILER-WAIT" (if we agree to consider java.awt HAND_CURSOR as a valid counterpart to ABL GLOVE). However I haven't even started the detailed comparision.

#45 - 11/13/2015 02:37 PM - Igor Skornyakov

I've configured HTTP server on my local machine and tryied to configure cursor image via URL: http://188.244.40.33/Busy.cur.

However at the windev01 IE reports that my current security settings do not allow this file to be downloaded. The attempt to use this URL as an argument of the LOAD-MOUSE-POINTER() fails and I guess that these two failures have the same reason.
How can I check http(s) URL for LOAD-MOUSE-POINTER() in windev01. It doesn't seem to have an http server installed.
Thank you,
Igor.

#46 - 11/13/2015 02:56 PM - Greg Shah

Don't worry about testing this right now. As I mentioned in note 29, we aren't going to support it right now. So I think it is OK to leave that part out of your tests.

#47 - 11/13/2015 02:58 PM - Igor Skornyakov

Greg Shah wrote:

> Don't worry about testing this right now. As I mentioned in note 29, we aren't going to support it right now. So I think it is OK to leave that part out of your tests.

Ok, thank you. Sorry, I understood note 29 that I do not have to **implement** it.

#48 - 11/13/2015 03:00 PM - Greg Shah

No, you did not misunderstand me. Generally, I want the full set of requirements to be known up front so I did want that to be part of your tests. But considering that it will add difficulty to the testing AND that it is not needed right now, I'm now excluding it from testing too.

#49 - 11/13/2015 03:05 PM - Igor Skornyakov

Greg Shah wrote:

> No, you did not misunderstand me. Generally, I want the full set of requirements to be known up front so I did want that to be part of your tests. But considering that it will add difficulty to the testing AND that it is not needed right now, I'm now excluding it from testing too.

I see. Thank you.

Section 27 has been updated.

#50 - 11/13/2015 04:27 PM - Igor Skornyakov

Section 27 has been updated.

#51 - 11/14/2015 02:42 AM - Igor Skornyakov

I've realised that it is possible to test https(s) - based PROPATH and LOAD-MOUSE-POINTER() on windev01. I can start pure Java https(s) server on windev01 listening on localhost. This doesn't require Windows registry modification.

It will take me about an hour to create such a server and to deploy it.

My questions are:
1. Is this allowed by the customer's security policy?
2. Is it worth to do at all?

Thank you.

#52 - 11/14/2015 04:47 AM - Igor Skornyakov

Section 27 has been updated.

All questions are answered. Can start design and implementation.

A more thorough analysis of **Q1h** and http(s) - based URLs can be performed in a background.

#53 - 11/14/2015 08:56 AM - Greg Shah

> 1. Is this allowed by the customer's security policy?

Yes.

> 2. Is it worth to do at all?

Yes, go ahead and complete it.

#54 - 11/14/2015 08:57 AM - Greg Shah

> Can start design and implementation.

Please do.

> A more thorough analysis of Q1h and http(s) - based URLs can be performed in a background.

OK.

#55 - 11/14/2015 08:58 AM - Igor Skornyakov

Greg Shah wrote:

> > 1. Is this allowed by the customer's security policy?
>
> Yes.
>
> > 2. Is it worth to do at all?
>
> Yes, go ahead and complete it.

OK, thank you.

#56 - 11/15/2015 05:43 AM - Igor Skornyakov

HTTP URLs as the LOAD-MOUSE-POINTER arguments and PROPATH entries have been tested.
Section 27 has been updated.

#57 - 11/17/2015 09:16 AM - Igor Skornyakov

I'm working now on the animated cursor support base on the sample code from http://www.informit.com/articles/article.aspx?p=1187852.
Is it acceptable?
Thank you.

#58 - 11/17/2015 11:58 AM - Greg Shah

> I'm working now on the animated cursor support base on the sample code from http://www.informit.com/articles/article.aspx?p=1187852.

> Is it acceptable?

I may have missed it, but it is not clear what the license is for that code. We cannot include code that is not clearly licensed.

Have you looked inside the twelve monkeys library that we already use to see if it supports reading cursors and animated cursors?

Finally, the animation side of this will be tricky. It must be implemented in both Swing and JS, so that means we would have to expose an API for this at the GUI driver level. The current project does not need that feature, so we are going to postpone it. Please create a new task in the UI project, document your findings and work so far. Link to this task as "related". Make sure I am a watcher.

#59 - 11/17/2015 12:13 PM - Igor Skornyakov

Greg Shah wrote:

> Have you looked inside the twelve monkeys library that we already use to see if it supports reading cursors and animated cursors?

As far as I understand this library supports .cur files but not .ani (at least its latest version from November 1, 2015). However .ani file format is documented and there is a sample reader implementation in the reference I've posted earlier. The only problem is to implement it as an ImageIO plug-in

> Finally, the animation side of this will be tricky. It must be implemented in both Swing and JS, so that means we would have to expose an API for this at the GUI driver level. The current project does not need that feature, so we are going to postpone it. Please create a new task in the UI project, document your findings and work so far. Link to this task as "related". Make sure I am a watcher.

Actually I didn't mean to re-use this code as-is, just use it as an inspiration. And I think that I have an idea how to implement animation for Swing. I know very little about modern HTML programming but I hope that there is a cursor animation support available out of the box.

#60 - 11/20/2015 01:52 PM - Igor Skornyakov

- File br.p added

BROWSE widget support with VIEW-AS TOGGLE-BOX and VIEW-AS COMBO-BOX options is incorrect - the converted code doesn't compile.
This means that mouse cursor support for the BROWSE will be incomplete as well.

#61 - 11/20/2015 02:52 PM - Greg Shah

> BROWSE widget support with VIEW-AS TOGGLE-BOX and VIEW-AS COMBO-BOX options is incorrect - the converted code doesn't compile. This means that mouse cursor support for the BROWSE will be incomplete as well.

Understood. This support will be added in #2628. Please leave clear TODOs in the code to make this easier.

#62 - 11/21/2015 05:14 AM - Igor Skornyakov

Section 27 has been updated

#63 - 11/25/2015 07:55 AM - Igor Skornyakov

The approach based on the EditableWidget should be re-designed for a number of reasons:
1. The mouse cursor shape for the ComboBox should take precedence over all other widgets' mouse cursors while this widget is active.
2. It is necessary to inherit the mouse cursor from the enclosing FRAME/WINDOW if it was set for them via LOAD-MOUSE-POINTER.
3. A special treatment is required for the BROWSE widget.

I'm implementing now the following approach.
- the "editable" widgets will implement an additional interface for accessing custom mouse pointer data.
- the EditableWidget becomes a mixin for such widgets.

#64 - 11/26/2015 06:08 PM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10954. The latest revision is now 10957.

#65 - 12/01/2015 06:38 AM - Igor Skornyakov

- File slider.p added

SLIDER doesn't work.

```
java.lang.NullPointerException: Could not resolve the configuration def for <null>: class com.goldencode.p2j.u
i.SliderConfig
        at com.goldencode.p2j.ui.ConfigManager.duplicate(ConfigManager.java:712)
        at com.goldencode.p2j.ui.ConfigSyncManager.registerConfig(ConfigSyncManager.java:292)
        at com.goldencode.p2j.aspects.ui.SyncConfigChangesAspect.afterConfigSet(SyncConfigChangesAspect.java:8
3)
        at com.goldencode.p2j.ui.GenericWidget.<init>(GenericWidget.java:234)
        at com.goldencode.p2j.ui.SliderWidget.<init>(SliderWidget.java:33)
        at com.goldencode.testcases.ui.abl.SliderFr$SliderFrDef.<init>(SliderFr.java:24)
        at sun.reflect.GeneratedConstructorAccessor23.newInstance(Unknown Source)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45
)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:422)
        at java.lang.Class.newInstance(Class.java:442)
        at com.goldencode.p2j.ui.GenericFrame.getConfigInstance(GenericFrame.java:1307)
        at com.goldencode.p2j.ui.GenericFrame.initializeNewFrame(GenericFrame.java:1349)
        at com.goldencode.p2j.ui.GenericFrame.createFrame_aroundBody0(GenericFrame.java:1034)
        at com.goldencode.p2j.ui.GenericFrame.createFrame_aroundBody1$advice(GenericFrame.java:54)
        at com.goldencode.p2j.ui.GenericFrame.createFrame(GenericFrame.java:1)
        at com.goldencode.testcases.abl.Slider.<init>(Slider.java:16)
        at sun.reflect.GeneratedConstructorAccessor21.newInstance(Unknown Source)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45
)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:422)
        at java.lang.Class.newInstance(Class.java:442)
        at com.goldencode.p2j.util.Utils.invoke(Utils.java:1281)
        at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1767)
        at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1270)
        at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:427)
        at sun.reflect.GeneratedMethodAccessor18.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)
        at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
        at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:705)
        at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
        at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
        at java.lang.Thread.run(Thread.java:745)
```

#66 - 12/01/2015 06:52 AM - Igor Skornyakov

- File mouse.p added

In 4GL the attached program opens two windows (a second one after any widget gains focus). The messages are displayed in the second (DEFAULT) window.
With p2j the default window remains closed and no messages are visible.

#67 - 12/01/2015 09:05 AM - Igor Skornyakov

- File mouse.p added

ON ENTRY trigger activation is not compatible with 4GL.

When the attached program runs in 4GL the ON ENTRY ANYWHERE trigger is never activated for WINDOW or FRAME while with p2j it happens sometimes.

#68 - 12/01/2015 05:01 PM - Igor Skornyakov

At this moment it seems that mouseEntered/mouseExited are not invoked for FRAME and WINDOW. We need toi implement it (or equivalent) methods. It is also necessary to have a stack of widgets which are hovered.

#69 - 12/02/2015 04:42 PM - Igor Skornyakov

TODO (Swing):
1. Support for special cases: BROWSE, COMBO-BOX
2. mouseEntered/mouseExited for FRAME and WINDOW.

#70 - 12/03/2015 08:28 AM - Constantin Asofiei

Igor Skornyakov wrote:

> At this moment it seems that mouseEntered/mouseExited are not invoked for FRAME and WINDOW. We need toi implement it (or equivalent) methods.

As mentioned in notes 3 and 5, the solution for the set-wait-state implementation needs to be made generic. So I don't think it matters that FRAME and WINDOW are not notified by mouseEntered and mouseExited events.

> It is also necessary to have a stack of widgets which are hovered.

The widget z-order is kept in each container, in AbstractContainer.widgets list.

#71 - 12/03/2015 08:41 AM - Igor Skornyakov

Constantin Asofiei wrote:

> > At this moment it seems that mouseEntered/mouseExited are not invoked for FRAME and WINDOW. We need toi implement it (or equivalent) methods.
>
> As mentioned in notes 3 and 5, the solution for the set-wait-state implementation needs to be made generic. So I don't think it matters that FRAME and WINDOW are not notified by mouseEntered and mouseExited events.

The problem is not related to the set-wait-state method (at least directly). As far as I understand now the custom mouse cursor for WINDOW should/may be implemented in a special way, however we really need mouseEntered/mouseExited for the FRAME.

> > It is also necessary to have a stack of widgets which are hovered.
>
> The widget z-order is kept in each container, in AbstractContainer.widgets list.

I was talking about a different thing. The stack I mean is at most 3 levels deep (window/frame/widget). I've actually implemented it already.

#72 - 12/03/2015 08:42 AM - Constantin Asofiei

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
> > > At this moment it seems that mouseEntered/mouseExited are not invoked for FRAME and WINDOW. We need toi implement it (or equivalent) methods.
> >
> > As mentioned in notes 3 and 5, the solution for the set-wait-state implementation needs to be made generic. So I don't think it matters that FRAME and WINDOW are not notified by mouseEntered and mouseExited events.
>
> The problem is not related to the set-wait-state method (at least directly). As far as I understand now the custom mouse cursor for WINDOW should/may be implemented in a special way, however we really need mouseEntered/mouseExited for the FRAME.

Please give an example why these are needed for FRAME.

> > > It is also necessary to have a stack of widgets which are hovered.
> >
> > The widget z-order is kept in each container, in AbstractContainer.widgets list.
>
> I was talking about a different thing. The stack I mean is at most 3 levels deep (window/frame/widget). I've actually implemented it already.

Please elaborate. Do you need to find which widget the topmost widget, on a certain mouse location?

#73 - 12/03/2015 08:55 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Please give an example why these are needed for FRAME.

Imagine that we have two frames with different custom mouse cursors (defined via LOAD-MOUSE-POINTER). W/o mouseEntered/mouseExited for the frame we cannot neither set the correct cursor not to reset it back when mouse leaves the frame.

> > > > It is also necessary to have a stack of widgets which are hovered.

> > > The widget z-order is kept in each container, in AbstractContainer.widgets list.

> > I was talking about a different thing. The stack I mean is at most 3 levels deep (window/frame/widget). I've actually implemented it already.

> Please elaborate. Do you need to find which widget the topmost widget, on a certain mouse location?

What we need is at what level of the WINDOW/FRAME/WIDGET hierarchy the mouse cursor is now. This is needed for example when the frame cursor is changed while the mouse is hovering over the widget w/o custom cursor. See AbstractWidget.setMousePointer in 2565a task branch.

#74 - 12/03/2015 09:30 AM - Constantin Asofiei

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
> > Please give an example why these are needed for FRAME.

Imagine that we have two frames with different custom mouse cursors (defined via LOAD-MOUSE-POINTER). W/o mouseEntered/mouseExited for the frame we cannot neither set the correct cursor not to reset it back when mouse leaves the frame.

I would expect the implementation should not require any specific code in widget classes (Frame, Window, etc). This should be generic (maybe in AbstractWidget?).

It is also necessary to have a stack of widgets which are hovered.

The widget z-order is kept in each container, in AbstractContainer.widgets list.

I was talking about a different thing. The stack I mean is at most 3 levels deep (window/frame/widget). I've actually implemented it already.

Please elaborate. Do you need to find which widget the topmost widget, on a certain mouse location?

What we need is at what level of the WINDOW/FRAME/WIDGET hierarchy the mouse cursor is now. This is needed for example when the frame cursor is changed while the mouse is hovering over the widget w/o custom cursor. See AbstractWidget.setMousePointer in 2565a task branch.

I don't think is OK to duplicate something which is already implemented. See Container.findMouseSource - this will give you the topmost widget at a certain location, in a specific window.

More, I took a quick look in 2565a and I think the config mouse-pointer field should be in one of the base classes (ControlSetConfig? BaseConfig?), and not added to every widget which supports this field. It's easier to maintain.

#75 - 12/03/2015 10:03 AM - Igor Skornyakov

Constantin Asofiei wrote:

> I would expect the implementation should not require any specific code in widget classes (Frame, Window, etc). This should be generic (maybe in AbstractWidget?).

This as already implemented like this. There is a minor difference for the FRAME because we need an enclosing WINDOW (even for the nested FRAME) while for other widgets an enclosing FRAME is required.

> I don't think is OK to duplicate something which is already implemented. See Container.findMouseSource - this will give you the topmost widget at a certain location, in a specific window.

This method is:
a). Expensive
b). Returns widgets with any MouseAction while we need only those ones which have MouseHoverAction.
c). At this moment it doesn't return Frame or Window. I'm trying to understand why.

> More, I took a quick look in 2565a and I think the config mouse-pointer field should be in one of the base classes (ControlSetConfig? BaseConfig?), and not added to every widget which supports this field. It's easier to maintain.

If we want to have this field in a single place it should be BaseConfig. I can put mousePointer field there if it is more correct from some point of view.

#76 - 12/03/2015 11:42 AM - Constantin Asofiei

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
>> I would expect the implementation should not require any specific code in widget classes (Frame, Window, etc). This should be generic (maybe in AbstractWidget?).
>
> This as already implemented like this. There is a minor difference for the FRAME because we need an enclosing WINDOW (even for the nested FRAME) while for other widgets an enclosing FRAME is required.

Please describe in detail how your solution will work. Look at how currently GuiDriver.registerEditableWidget is implemented (EditableWidget is Swing and p2j.mouse.js:MouseEditable for Web) - we should reuse and expand this code, if needed move more detailed z-order dataat the JS driver. We need to do as much work at the driver level as possible. I don't want to add mouseEntry/Leave support for all widgets, and send these events down to the client side, just to change the mouse pointer.

> I don't think is OK to duplicate something which is already implemented. See Container.findMouseSource - this will give you the topmost widget at a certain location, in a specific window.

> This method is:
> a). Expensive
> b). Returns widgets with any MouseAction while we need only those ones which have MouseHoverAction.
> c). At this moment it doesn't return Frame or Window. I'm trying to understand why.

I think it doesn't return Frame because Frame.findMouseSource doesn't check its own boundaries, if the frameScroll.findMouseSource call returns null. Same might be for Window.

> More, I took a quick look in 2565a and I think the config mouse-pointer field should be in one of the base classes (ControlSetConfig? BaseConfig?), and not added to every widget which supports this field. It's easier to maintain.

> If we want to have this field in a single place it should be BaseConfig. I can put mousePointer field there if it is more correct from some point of view.

The attribute is common for enough widgets to worth it to be placed in BaseConfig. Please add to its javadoc the intended attributes using it.

#77 - 12/03/2015 11:50 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Please describe in detail how your solution will work. Look at how currently GuiDriver.registerEditableWidget is implemented (EditableWidget is Swing and p2j.mouse.js:MouseEditable for Web) - we should reuse and expand this code, if needed move more detailed z-order dataat the JS driver. We need to do as much work at the driver level as possible. I don't want to add mouseEntry/Leave support for all widgets, and send these events down to the client side, just to change the mouse pointer.

Constantin. I'm deep in the testing/coding right now. It is possible that the ideas I have (and mostly implemented right now) will be revised. If you don't mind I will provide a detailed description when it will work (at least for Swing). After that I'm ready to improve my approach.
The EditableWidget had a number of limitations (the most notable is the "hard-coded" cursor type) It is replaced by the MouseHoverAction in 2565a.

#78 - 12/03/2015 05:49 PM - Igor Skornyakov

FrameGuiImpl.mouse(Entered/Exited) is implemented. It is clear how to do the same for the WindowGuiImpl.

Committed to branch 2565a revno 10962.

The SELECTION-LIST doesn't work. When mouse hovers over it we see the following in the client log:

```
Received mouse event for zombie source -37: java.awt.event.MouseEvent[MOUSE_ENTERED,(359,152),absolute(1282,23
0),button=0,clickCount=0] on com.goldencode.p2j.ui.client.gui.driver.swing.SwingEmulatedWindow$1[,0,0,608x919,
alignmentX=0.0,alignmentY=0.0,border=,flags=0,maximumSize=java.awt.Dimension[width=608,height=919],minimumSize
=java.awt.Dimension[width=608,height=919],preferredSize=java.awt.Dimension[width=608,height=919]]
```

This can be a result of my changes. Investigating.

#79 - 12/04/2015 03:45 AM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10955. The latest revision is now 10963.

#80 - 12/04/2015 04:58 PM - Igor Skornyakov

1. Implemented WindowGuiImpl.mouse(Entered/Exited). However there is a problem. When mouse enters Window from the enclosed frame, top, and bottom the mouse pointer changes OK, but it doesn't changes when mouse enters from the lift or from the right. I understand that it is because of the MouseResizeable but at this moment don't know how to fix it. Investigating.
2. The issue with SelectionListGuiImpl mentioned in the note 78 is the result of the scrollpane was not registered. I've added the registration to the initialize method and now the widget is selectable, LOAD-MOUSE-POINTER works, but one cannot make a selection with mouse. It doesn't look like a result of my changes. Should I fix the rest of the issue with the SELECTION-LIST in the scope of this task? I couldn't find this issue in the opened

Redmine tasks.
Thank you.
Committed to the task branch 2565a revno 10964.

#81 - 12/04/2015 05:04 PM - Greg Shah

> but one cannot make a selection with mouse.

I think this is a known issue. See #2879. If you are talking about the same thing, then you don't need to work this now.

#82 - 12/04/2015 05:06 PM - Igor Skornyakov

Greg Shah wrote:

> I think this is a known issue. See #2879. If you are talking about the same thing, then you don't need to work this now.

Yes, it looks like that. Thank you.

#83 - 12/04/2015 05:30 PM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10956. The latest revision is now 10965.

#84 - 12/07/2015 04:41 AM - Igor Skornyakov

Resolved the issue with WINDOW custom cursor mentioned in the note 80.

Committed to branch 2565a revno 10966.

#85 - 12/07/2015 05:48 AM - Igor Skornyakov

Section 27 has been updated.

#86 - 12/07/2015 06:42 AM - Igor Skornyakov

Because of the #2832 the custom mouse pointer for the SIMPLE COMBO-BOX cannot be implemented now.

#87 - 12/07/2015 08:47 AM - Greg Shah

Actually, Eugenie is not too far from having #2832 resolved. He already has it resolved enough to use simple combo-boxes for your purposes.

Please implement the support. You can test it by creating a branch from 2832a and then applying your changes to it. You can then use/test simple mode. Apply any fixes back to either 2565a or 2832a depending on what fixes are needed. Anything you apply back to 2832a will need to be discussed/coordinated with Eugenie.

#88 - 12/07/2015 08:54 AM - Igor Skornyakov

Greg Shah wrote:

> Actually, Eugenie is not too far from having #2832 resolved. He already has it resolved enough to use simple combo-boxes for your purposes.
>
> Please implement the support. You can test it by creating a branch from 2832a and then applying your changes to it. You can then use/test simple mode. Apply any fixes back to either 2565a or 2832a depending on what fixes are needed. Anything you apply back to 2832a will need to be discussed/coordinated with Eugenie.

OK. Thank you, The problem is that there are too many changes to be applied. I will think how do do it.

#89 - 12/07/2015 08:55 AM - Greg Shah

Make sure you are working in a fresh branch, so you don't mix up the 2565a branch. Then, just use bzr merge and take care of any conflicts as you go.

#90 - 12/07/2015 09:00 AM - Igor Skornyakov

Greg Shah wrote:

> Make sure you are working in a fresh branch, so you don't mix up the 2565a branch. Then, just use bzr merge and take care of any conflicts as you go.

Got it. Thank you.

#91 - 12/10/2015 03:51 AM - Igor Skornyakov

Section 27 has been updated.

#92 - 12/12/2015 06:13 PM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10958. The latest revision is now 10973.

#93 - 12/13/2015 04:17 AM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10959. The latest revision is now 10974.

#94 - 12/13/2015 03:14 PM - Igor Skornyakov

After the rebase the situation with COMBO-BOX border and custom mouse cursor activation is fixed. However the ON ENTRY trigger is not activated for the SIMPLE and DROP-DOWN-LIST COMBO-BOX if it gains focus after TAB or mouse click **inside** the entry field.

#95 - 12/14/2015 08:42 AM - Greg Shah

OK, create a sub-task of #2677 for this, unless you need it fixed for this task.

#96 - 12/14/2015 09:00 AM - Igor Skornyakov

Greg Shah wrote:

> OK, create a sub-task of #2677 for this, unless you need it fixed for this task.

OK, thank you. The reason why I need it for this task is that SET-WAIT-STATE method closes the active drop-down list and in my test it is initiated by the trigger. If I will find a workaround I will create a sub-task. Otherwise I have to fix it.

#97 - 12/14/2015 10:56 AM - Igor Skornyakov

Added JavaDoc and history
Committed to the task branch 2565a revno 10977.

#98 - 12/14/2015 04:13 PM - Igor Skornyakov

TODO:
1. Load custom cursor from file. This can be done in one or two days.
2. Add custom cursor support for BROWSE in the edit mode and SLIDER. This can be done when corresponding functionality will be implemented.
3. Close active drop-down on SET-WAIT-STATE activation. The clean implementation can be possible when #2935 will be resolved
4. Implement LOAD-MOUSE-POINTER and SET-WAIT-STATE for the Web GUI. I'm not familiar with this part of the functionality so cannot provide an estimation at this moment.

#99 - 12/15/2015 04:18 AM - Igor Skornyakov

Section 27 has been updated.

#100 - 12/15/2015 08:26 AM - Greg Shah

> 1. Load custom cursor from file. This can be done in one or two days.
> 2. Add custom cursor support for BROWSE in the edit mode and SLIDER. This can be done when corresponding functionality will be implemented.

Don't work these right now.

> 3. Close active drop-down on SET-WAIT-STATE activation. The clean implementation can be possible when #2935 will be resolved

Go ahead and fix #2935. Then finish this feature.

> 4. Implement LOAD-MOUSE-POINTER and SET-WAIT-STATE for the Web GUI. I'm not familiar with this part of the functionality so cannot provide an estimation at this moment.

Ask Sergey for help. Explain the requirements (e.g. GuiDriver changes that must be implemented) and he can help with the implementation.

#101 - 12/15/2015 08:30 AM - Igor Skornyakov

Greg Shah wrote:

> > 1. Load custom cursor from file. This can be done in one or two days.
> > 2. Add custom cursor support for BROWSE in the edit mode and SLIDER. This can be done when corresponding functionality will be implemented.
>
> Don't work these right now.
>
> > 3. Close active drop-down on SET-WAIT-STATE activation. The clean implementation can be possible when #2935 will be resolved
>
> Go ahead and fix #2935. Then finish this feature.
>
> > 4. Implement LOAD-MOUSE-POINTER and SET-WAIT-STATE for the Web GUI. I'm not familiar with this part of the functionality so cannot provide an estimation at this moment.
>
> Ask Sergey for help. Explain the requirements (e.g. GuiDriver changes that must be implemented) and he can help with the implementation.

OK, thank you. Can you please assign #2935 to me? Should I create a separate task branch for this task?
Thank you.

#102 - 12/15/2015 08:32 AM - Greg Shah

You should be able to assign #2935 to yourself. If that doesn't work, let me know.

> Should I create a separate task branch for this task?

No. The changes are needed for this task so just put them into 2565a.

#103 - 12/15/2015 08:40 AM - Igor Skornyakov

Greg Shah wrote:

> You should be able to assign #2935 to yourself. If that doesn't work, let me know.
>
> > Should I create a separate task branch for this task?
>
> No. The changes are needed for this task so just put them into 2565a.

Thank you Greg.

#104 - 01/04/2016 09:58 AM - Igor Skornyakov

Implemented active COMBO-BOX close on SET-WAIT-STATE.

Committed to the task branch 2565a revno 10981.

#105 - 01/07/2016 09:43 AM - Greg Shah

Please post details on your current work for loading pointers from file. Also: please check in your current work so I can see the progress you are making.

The code in rev 10978 is not matching this guidance from note 3:

> We already support loading images. The custom cursor case would reply upon our current image loading capabilities. Please see #2479 and #2792 to get a better understanding of how this works. Eugenie can answer questions you may have. You may have to extend our support to ensure it handles the Windows cursor bitmap types (supposedly these are .cur and .ani but who knows what is really possible?). But generally, it seems like the core functionality for loading is already there. We already support using the server-side application JAR file first and then using the client's filesystem including searching the propath for relative names. This JAR support allows us to add any image resources (now including cursors) to the converted application's jar file and to leverage those so that they don't need to be copied into the client's filesystem. But the converted code is the same. Please note that our image loading code has portions on both the server (jar stuff) and client (filesystem/propath) to implement.

You previously estimated that it would take 1-2 days to implement this. Is there some problem that is causing it to take longer?

#106 - 01/07/2016 09:47 AM - Igor Skornyakov

Greg Shah wrote:

> Please post details on your current work for loading pointers from file. Also: please check in your current work so I can see the progress you are making.
>
> The code in rev 10978 is not matching this guidance from note 3:
>
> > We already support loading images. The custom cursor case would reply upon our current image loading capabilities. Please see #2479 and #2792 to get a better understanding of how this works. Eugenie can answer questions you may have. You may have to extend our support to ensure it handles the Windows cursor bitmap types (supposedly these are .cur and .ani but who knows what is really possible?). But generally, it seems like the core functionality for loading is already there. We already support using the server-side application JAR file first and then using the client's filesystem including searching the propath for relative names. This JAR support allows us to add any image resources (now including cursors) to the converted application's jar file and to leverage those so that they don't need to be copied into the client's filesystem. But the converted code is the same. Please note that our image loading code has portions on both the server (jar stuff) and client (filesystem/propath) to implement.
>
> You previously estimated that it would take 1-2 days to implement this. Is there some problem that is causing it to take longer?

Actually I hope to finish the load from the file (at least for the .cur files) in an hour or two. After that I will answer your questions. Is it OK?
Thank you.

#107 - 01/07/2016 09:47 AM - Greg Shah

OK.

#108 - 01/07/2016 11:04 AM - Igor Skornyakov

Implemented custom cursor load from .cur files.
Committed to the task branch 2565a revno 10982.

Reviewing the file resolution logic.

#109 - 01/07/2016 11:15 AM - Igor Skornyakov

> Greg Shah wrote:

> The code in rev 10978 is not matching this guidance from note 3:
>
>> We already support loading images. The custom cursor case would reply upon our current image loading capabilities. Please see #2479 and #2792 to get a better understanding of how this works. Eugenie can answer questions you may have. You may have to extend our support to ensure it handles the Windows cursor bitmap types (supposedly these are .cur and .ani but who knows what is really possible?). But generally, it seems like the core functionality for loading is already there. We already support using the server-side application JAR file first and then using the client's filesystem including searching the propath for relative names. This JAR support allows us to add any image resources (now including cursors) to the converted application's jar file and to leverage those so that they don't need to be copied into the client's filesystem. But the converted code is the same. Please note that our image loading code has portions on both the server (jar stuff) and client (filesystem/propath) to implement.

Sorry Greg, as far as I understand I resolve cursor file name in exactly the same way it is done for images/icons (via fsd.searchPath(name)). There was a typo which is fixed in revno 10983. Did you mean this typo?
Thank you.

> You previously estimated that it would take 1-2 days to implement this. Is there some problem that is causing it to take longer?

Sorry, but after I was distracted with #2935 I had to recall the logic I had in mind.

#110 - 01/07/2016 11:31 AM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10962. The latest revision is now 10986.

Greg,
Can I start working on LOAD-MOUSE-POINTER implementation for BROWSE?
Thank you.

#111 - 01/07/2016 02:07 PM - Greg Shah

> Sorry Greg, as far as I understand I resolve cursor file name in exactly the same way it is done for images/icons (via fsd.searchPath(name)).

No, I don't think you understood what I have requested. Please carefully read #2792.

Then look at the following:

- LogicalTerminal.setImage() (used from ButtonWidget and ImageWidget on the server-side)
- LogicalTerminal.getImageStreamFromApplication()
- ThinClient.setImage()
- ThinClient.loadImage()
- ImageHelper.isPrepackaged()
- ImageHelper.getPrepackagedImageData()

You can see through these methods, that we support loading images using the following precedence order:

1. well-known image names that are hard coded into the Progress 4GL runtime (bypasses server-side processing and loads them as resources from p2j.jar on the client when ClientExports.loadImage() is called below)
2. images from the server-side application jar (not p2j.jar), using getImageStreamFromApplication() we search the application jars in the classpath and honor the original system case-sensitivity, the "pkgroot" of the converted code and the propath, if found in the application jar, we read the bytes as a resource and send them to the client using ClientExports.setImage()
3. call ClientExports.loadImage() which does the following:
   1. if processing a predefined image (see above), then will load this from the p2j.jar as a resource
   2. search the client filesystem, honoring the propath AND original system case-sensitivity
   3. search the client filesystem, honoring the propath AND original system case-sensitivity AND trying common file extensions (e.g. .bmp) which is how the 4GL does it

Your implementation has problems:

- It doesn't implement this same precedence order/features of loading.
- It doesn't leverage existing image loading code.
- It directly uses AWT (e.g. BufferedImage, Toolkit, Point...) from ThinClient which is not good. All AWT and image-specific processing should be hidden in helper classes.
- The common code for the client directly uses java.awt.Cursor which is not good. Note how we have an ImageWrapper instance used at the GuiDriver level (and in the common code for all the widgets) to hide the AWT-specific implementation details. Look carefully at how that is used in ui/client/gui/driver/ and use a similar approach.
- It is not clear if your implementation supports the full range of cursors that are supported in the 4GL. Please document this.
- The web client support is missing. As noted above, please document the requirements and work with Sergey to find the proper implementation.

  Can I start working on LOAD-MOUSE-POINTER implementation for BROWSE?

You can work on that after you to finish the custom cursor loading support.

#112 - 01/07/2016 02:12 PM - Igor Skornyakov

Greg Shah wrote:

> > Sorry Greg, as far as I understand I resolve cursor file name in exactly the same way it is done for images/icons (via fsd.searchPath(name)).
>
> No, I don't think you understood what I have requested. Please carefully read #2792.
>
> Then look at the following:
>
> - LogicalTerminal.setImage() (used from ButtonWidget and ImageWidget on the server-side)
> - LogicalTerminal.getImageStreamFromApplication()
> - ThinClient.setImage()
> - ThinClient.loadImage()
> - ImageHelper.isPrepackaged()
> - ImageHelper.getPrepackagedImageData()
>
> You can see through these methods, that we support loading images using the following precedence order:
>
> 1. well-known image names that are hard coded into the Progress 4GL runtime (bypasses server-side processing and loads them as resources from p2j.jar on the client when ClientExports.loadImage() is called below)
> 2. images from the server-side application jar (not p2j.jar), using getImageStreamFromApplication() we search the application jars in the classpath and honor the original system case-sensitivity, the "pkgroot" of the converted code and the propath, if found in the application jar, we read the bytes as a resource and send them to the client using ClientExports.setImage()
> 3. call ClientExports.loadImage() which does the following:
>   1. if processing a predefined image (see above), then will load this from the p2j.jar as a resource
>   2. search the client filesystem, honoring the propath AND original system case-sensitivity
>   3. search the client filesystem, honoring the propath AND original system case-sensitivity AND trying common file extensions (e.g. .bmp) which is how the 4GL does it
>
> Your implementation has problems:
>
> - It doesn't implement this same precedence order/features of loading.
> - It doesn't leverage existing image loading code.
> - It directly uses AWT (e.g. BufferedImage, Toolkit, Point...) from ThinClient which is not good. All AWT and image-specific processing should be hidden in helper classes.
> - The common code for the client directly uses java.awt.Cursor which is not good. Note how we have an ImageWrapper instance used at the GuiDriver level (and in the common code for all the widgets) to hide the AWT-specific implementation details. Look carefully at how that is used in ui/client/gui/driver/ and use a similar approach.
> - It is not clear if your implementation supports the full range of cursors that are supported in the 4GL. Please document this.
> - The web client support is missing. As noted above, please document the requirements and work with Sergey to find the proper implementation.

I see. Thank you. I will re-work my approach.

#113 - 01/07/2016 03:19 PM - Igor Skornyakov

Section 27 has been updated. (see section **N6**)

#114 - 01/08/2016 03:36 AM - Igor Skornyakov

Greg,
I've noticed that there are "default" extensions for the cursor pointer files (see note 27 **N6**). For me this was completely unexpected especially regarding the **.ico** extension which wasn't mentioned in the Progress documentation at all. I think that it may have sense to monitor the files' operations at the mouse cursor load with something like a FileMon utility (see https://technet.microsoft.com/en-us/sysinternals/filemon.aspx) to check which extensions are checked and in what order. Do you think that it makes sense and does it comply with the security policy on windev01 and the Progress licence terms?
Thank you.

#115 - 01/08/2016 07:57 AM - Greg Shah

> I think that it may have sense to monitor the files' operations at the mouse cursor load with something like a FileMon utility (see https://technet.microsoft.com/en-us/sysinternals/filemon.aspx) to check which extensions are checked and in what order.

Yes, I completely agree. Good idea.

> does it comply with the security policy on windev01 and the Progress licence terms?

Yes, it is in compliance with the security policy. It is also completely safe regarding the Progress license. By using something like Process Monitor, we are just observing the API calls that are invoked on the OS. This is completely legal. We are not disassembling the binaries or looking at the Progress runtime's source or object code in any way. Please go ahead.

#116 - 01/08/2016 08:01 AM - Igor Skornyakov

Greg Shah wrote:

> Yes, it is in compliance with the security policy. It is also completely safe regarding the Progress license. By using something like Process Monitor, we are just observing the API calls that are invoked on the OS. This is completely legal. We are not disassembling the binaries or looking at the Progress runtime's source or object code in any way. Please go ahead.

OK, thank you.

#117 - 01/08/2016 08:30 AM - Igor Skornyakov

Section 27 has been updated. (see section **N6**)

#118 - 01/08/2016 10:19 AM - Igor Skornyakov

Section 27 has been updated. (see section **N6**)

#119 - 01/13/2016 06:01 AM - Igor Skornyakov

Re-worked the image search.

Committed to the task branch 2565a revno 10989.

#120 - 01/14/2016 06:45 AM - Igor Skornyakov

Finished re-work of the mouse cursor load from a file.
1. The file search re-uses the (updated) logic for the image.
2. Swing specific code is moved to the SwingGuiDriver
3. The 'prepackaged' semantics is encoded in the MousePointer class.

Committed to the task branch 2565a revno 10991.

Working on BROWSE:LOAD-MOUSE-POINTER

There is still needed to have images for pre-defined mouse pointers which do not have counterparts in Swing (see note 27 **Q8**).

#121 - 01/14/2016 12:06 PM - Greg Shah

Code Review Task Branch 2565a Revision 10991

This is much closer to the correct implementation, but there are still some things to modify.

1. It seems to me that if the 4GL code specifies any of the Progress pre-defined cursor names, then all other processing for that cursor load should occur on the client side. Today, the code makes a server-side distinction between the pre-defined cursors that have known AWT cursors that can be loaded and the pre-defined cursors for which we don't yet have an image. I don't want the server code to have any idea about this distinction. It makes the server code more complicated and it also adds to the number of ClientExports needed to support things. Perhaps I am missing something, but it seems the solution can be simpler/cleaner by hiding the knowledge of how to load a pre-defined cursor in a single class on the client side. For example, I don't want ThinClient or other client side code to know this fact either. Every pre-defined cursor is either being mapped to an AWT cursor or will be loaded from some image resource in the p2j.jar (when we have such images available). That loading is not very hard. Perhaps it should be hidden in CursorType.

In other words, I don't think we should treat the non-AWT pre-defined cursor types as "CUSTOM". It confuses things.

2. I appreciate that the AWT code is no longer in ThinClient. That is a good step forward. However we do not want the specific loading code to be in the Swing driver. Consider that in both the web client and the Swing client, image loading and cursor loading occurs:

- in the same way
- using a byte[] from the server side, a resource from the J2SE (AWT), a resource from p2j.jar or a file in the client file system

This means we want to create a helper class to do this work and this helper class can be common code that is in the generic GUI code, not in any specific GUI driver.

We already do it this way for images. In other words, we accept that there will be a small amount of image and cursor code that uses AWT but is still common GUI code.

3. There are a large number of coding standard issues. It would take a long time for me to write them all here. Please review the diffs in 2565a from the trunk and clean up the code. The following are some of the issues to look out for:

- missing spaces in control structure (e.g. for(;;) instead of for (;;)
- missing javadoc for classes
- missing javadoc for methods

- missing javadoc for data members
- incomplete javadoc (e.g. parameter names missing or @param sections missing for some parameters)
- missing blank lines in between data members or methods
- extra blank lines at the end of a class or method
- incorrect ordering of methods (e.g. protected/private methods up in the public method section)

4. I think the existing javadoc for the ClientExports/ThinClient/LogicalTerminal methods needs to be much more clear about the purpose of each method. Some of the descriptions seem wrong. The parameters are also not documented well.

5. GenericWidget.loadMousePointer(String) really would be better if it had comments that explained the different paths through that code. It seems to me that there are 4 possible paths:

- reset to default cursor
- predefined cursor to be loaded from the client (AWT or p2j.jar)
- filename which is found in the application jar via propath search
- filename which is to be searched for in the client file system via propath search

But it is not easy to see those paths from the code.

6. In GenericWidget.loadMousePointer(String), it seems like the new code you added will exit before setting the pointer name:

```
         if (ptr.fileName != null)
         {
            return LogicalTerminal.loadMousePointer(getId(), ptr);
         }
         h.setMousePointerName(ptrName);
```

Is that intentional?

#122 - 01/14/2016 12:09 PM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 10991
>
> This is much closer to the correct implementation, but there are still some things to modify.

Thank you Greg,
I will re-work my code.

#123 - 01/14/2016 02:01 PM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10963. The latest revision is now 10992.

#124 - 01/14/2016 04:05 PM - Igor Skornyakov

Partially reworked the code based on the code review

Committed to the task branch 2565a 10994.

Sections 1, 4 and 6 where addressed.

Greg,

I would like to discuss section 2 of your comments. Do you insist on a separate helper class? It seems natural for me that GuiDriver creates cursor which is suitable for it. For example SwingGuiDriver creates AWT cursor which is not suitable for the WebGuiDriver. The common code can be placed to the AbstractGuiDriver. Creating a helper class with two flavours seems to be just an overhead.
What do you think about that?
Thank you.

#125 - 01/14/2016 04:11 PM - Greg Shah

> It seems natural for me that GuiDriver creates cursor which is suitable for it. For example SwingGuiDriver creates AWT cursor which is not suitable for the WebGuiDriver.

We have a common implementation already for images. Are you sure that the same in-memory cursor implementation cannot be used in the browser? I am hoping it can be used that same way.

It is an important thing to resolve now (web client support).

#126 - 01/14/2016 04:18 PM - Igor Skornyakov

Greg Shah wrote:

> We have a common implementation already for images. Are you sure that the same in-memory cursor implementation cannot be used in the browser? I am hoping it can be used that same way.
>
> It is an important thing to resolve now (web client support).

I see. So I will start the design of the web part and when it will be clear for me what is needed exactly I will continue will the refactoring. Is it OK?
Thank you.

#127 - 01/14/2016 04:37 PM - Igor Skornyakov

Greg Shah wrote:

> We have a common implementation already for images. Are you sure that the same in-memory cursor implementation cannot be used in the browser? I am hoping it can be used that same way.

BTW: As far as I can see the ImageWrapper is created by GuiDriver which looks similar to my approach. The ImageHelper is just an image data repository.

#128 - 01/15/2016 05:57 AM - Igor Skornyakov

Fixed javadoc/formatting issues.
Committed to the 2562a branch revno 10995.

I've found a lot of formatting issues like "for(" or "if(". Those which are part of my changes are fixed (hopefully all). The rest are left intact.
There is also a number of ordering issues of public/private methods (e.g. in ThinClient). I haven't fixed those ones which are not mine to avoid problems with merging.

#129 - 01/15/2016 07:21 AM - Greg Shah

> So I will start the design of the web part and when it will be clear for me what is needed exactly I will continue will the refactoring. Is it OK?

Yes. Focus first on the way that the javascript code needs to register a custom cursor. Work with Sergey on this, he already does have some cursor management code there, but it will need to be extended. This should clarify the format of the data needed to send a custom cursor to the javascript side. If possible, use a common format with AWT.

> As far as I can see the ImageWrapper is created by GuiDriver which looks similar to my approach.

It is. But it is used for both drivers, not just Swing.

> I haven't fixed those ones which are not mine to avoid problems with merging.

Yes, this is a good choice.

#130 - 01/20/2016 02:41 PM - Igor Skornyakov

Currently in the WebGUI the widget-specific mouse cursor support is mostly hard-coded: based on the single flag the cursor is changed to the 'text' style on enter and to the 'default' one on exit.

To implement a full LOAD-MOUSE-PONTER support the logic based on a p2j.screen.setCursorStyle(cursorStyleId, wid) should be re-designed to support dynamically changeable cursors (both for the widgets and enclosing frames).

The support for the custom cursors (from files) seems to be not that involved: according to http://stackoverflow.com/questions/10866471/javascript-how-to-change-mouse-cursor-to-an-image it is sufficient to use assignment to canvas.style.cursor of a file's URL. It seems that this URL should be resolved in a way similar to resolving URLs of JS files in the index.html. As far as I can see the images are processed in a different way.

At this moment it is unclear if the LOAD-MOUSE-PONTER support for FRAME and WINDOW and drop-down flavours of the COMBO-BOX will add additional complexity

#131 - 01/20/2016 02:56 PM - Sergey Ivanovskiy

Igor Skornyakov wrote:

> Currently in the WebGUI the widget-specific mouse cursor support is mostly hard-coded: based on the single flag the cursor is changed to the 'text' style on enter and to the 'default' one on exit.
>
> To implement a full LOAD-MOUSE-PONTER support the logic based on a p2j.screen.setCursorStyle(cursorStyleId, wid) should be re-designed to support dynamically changeable cursors (both for the widgets and enclosing frames).
>
> The support for the custom cursors (from files) seems to be not that involved: according to http://stackoverflow.com/questions/10866471/javascript-how-to-change-mouse-cursor-to-an-image it is sufficient to use assignment to canvas.style.cursor of a file's URL. It seems that this URL should be resolved in a way similar to resolving URLs of JS files in the index.html. As far as I can see the images are processed in a different way.

Yes, agree. For the web the custom cursor styles can be implemented like the web embedded fonts with help of css style and forming url(...) with binary data. I believe that the customs cursor styles are more like the web embedded fonts styles in implementation.
p2j.screen.setCursorStyle(cursorStyleId, wid) can be extended to support binary data.

#132 - 01/20/2016 03:01 PM - Greg Shah

> The support for the custom cursors (from files) seems to be not that involved: according to http://stackoverflow.com/questions/10866471/javascript-how-to-change-mouse-cursor-to-an-image it is sufficient to use assignment to

> canvas.style.cursor of a file's URL. It seems that this URL should be resolved in a way similar to resolving URLs of JS files in the index.html. As far as I can see the images are processed in a different way.

In javascript apps, images are commonly accessed via URL, but we implemented our own mechanism for delivering the image to the client through our websocket. This was done for multiple reasons. We want to do the same thing for cursors. In other words, we will not be accessing cursors via a URL. Instead, on the Java client side, we will do any processing of the cursor into the correct format. Then we will send the bytes down to the JS side using the websocket. Finally, we must load that result in some way.

Please investigate options for loading cursors from memory. See #1811-652 (and 653) which describe the data: URI approach. Hopefully this is possible with cursors too. Also see #1811-1576.

#133 - 01/20/2016 03:06 PM - Igor Skornyakov

Greg Shah wrote:

> In javascript apps, images are commonly accessed via URL, but we implemented our own mechanism for delivering the image to the client through our websocket. This was done for multiple reasons. We want to do the same thing for cursors. In other words, we will not be accessing cursors via a URL. Instead, on the Java client side, we will do any processing of the cursor into the correct format. Then we will send the bytes down to the JS side using the websocket. Finally, we must load that result in some way.
>
> Please investigate options for loading cursors from memory. See #1811-652 (and 653) which describe the data: URI approach. Hopefully this is possible with cursors too. Also see #1811-1576.

OK, thank you.

#134 - 01/20/2016 03:46 PM - Igor Skornyakov

- File cursor2.html added

The mouse cursor definition with base65 encoded data works (at least with Firefox and Chrome). See attached files.

#135 - 01/20/2016 03:48 PM - Igor Skornyakov

- File cursor2.css added

See CSS file for the previous note

#136 - 01/21/2016 10:31 AM - Greg Shah

This is good work. Is the format just the base64 encoded version of the in-memory Java cursor?

With this knowledge, how quickly can you finish this work?

#137 - 01/21/2016 10:36 AM - Igor Skornyakov

Greg Shah wrote:

> This is good work. Is the format just the base64 encoded version of the in-memory Java cursor?

This is a base64 encoded version of the .cur file.

> With this knowledge, how quickly can you finish this work?

Apart from the FRAME, WINDOW and drop-down flavours of the COMBO-BOX I plan to finish it tomorrow or at the weekend. Please remember however that I haven't yet started work on BROWSE (even for Swing).

#138 - 01/26/2016 03:41 AM - Igor Skornyakov

The WebGUI logic based on mouseActions should be substantially re-worked.
1. The MouseEditable JS object state should be expanded by enabled, parent and cursor fields to support dynamically assigned custom cursor.
2. As far as I can see currently the registerMouseWidgets() is invoked in getKeyStroke() method only. This looks both expensive and (more importantly) doesn't disable/enable mouse cursor change on hover over a widget which is disabled/enabled.

#139 - 01/26/2016 08:30 AM - Constantin Asofiei

Igor Skornyakov wrote:

> The WebGUI logic based on mouseActions should be substantially re-worked.
> 1. The MouseEditable JS object state should be expanded by enabled, parent and cursor fields to support dynamically assigned custom cursor.

Why do you need enabled information at the JS level? The driver already excludes non-enabled widgets from this list. If the widget becomes enabled, with will be registered with JS side. If it gets disabled, will be de-registered.
Also, how is parent involved for cursor processing? The way I see it, the cursor should be registered with a widget ID, and when hovering that widget, get its custom cursor; if it doesn't exist, use default.

> 2. As far as I can see currently the registerMouseWidgets() is invoked in getKeyStroke() method only. This looks both expensive and (more importantly) doesn't disable/enable mouse cursor change on hover over a widget which is disabled/enabled.

registerMouseWidgets() is built to:

1. on first call, send all widgets which match visible && enabled
2. on subsequent calls, send only new or dead widgets (matching the same criteria). If there are no changes in the widgets, then this will be a no-op.

This is not supposed to send to the JS side all the widgets, every time is called. Or are you referring that the Java-side execution of registerMouseWidgets() is expensive?

#140 - 01/26/2016 08:37 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor Skornyakov wrote:
>
> > The WebGUI logic based on mouseActions should be substantially re-worked.
> > 1. The MouseEditable JS object state should be expanded by enabled, parent and cursor fields to support dynamically assigned custom cursor.
>
> Why do you need enabled information at the JS level? The driver already excludes non-enabled widgets from this list. If the widget becomes enabled, with will be registered with JS side. If it gets disabled, will be de-registered.

This is correct, I do not need it.

> Also, how is parent involved for cursor processing? The way I see it, the cursor should be registered with a widget ID, and when hovering that widget, get its custom cursor; if it doesn't exist, use default.

The parent is needed to set the correct cursor on mouse exit. This can be different from the cursor before enter if e.g. the cursor for the enclosing FRAME was changed while hovering over the widget.

> > 2. As far as I can see currently the registerMouseWidgets() is invoked in getKeyStroke() method only. This looks both expensive and (more importantly) doesn't disable/enable mouse cursor change on hover over a widget which is disabled/enabled.
>
> registerMouseWidgets() is built to:
>
> 1. on first call, send all widgets which match visible && enabled
> 2. on subsequent calls, send only new or dead widgets (matching the same criteria). If there are no changes in the widgets, then this will be a no-op.
>
> This is not supposed to send to the JS side all the widgets, every time is called. Or are you referring that the Java-side execution of registerMouseWidgets() is expensive?

Yes, I mean Java-side processing. But regardless of the cost I've noticed that the information about disabled widgets is not properly propagated to the browser.

#141 - 01/26/2016 08:51 AM - Constantin Asofiei

Igor Skornyakov wrote:

> > Also, how is parent involved for cursor processing? The way I see it, the cursor should be registered with a widget ID, and when hovering that widget, get its custom cursor; if it doesn't exist, use default.
>
> The parent is needed to set the correct cursor on mouse exit. This can be different from the cursor before enter if e.g. the cursor for the enclosing FRAME was changed while hovering over the widget.

Please post a 4GL example. I doubt we need parent-child relationship on JS side... we already have z-order, and the frame background should be below its containing widgets, in z-order. So, a FRAME should act as a normal widget.

> > > 2. As far as I can see currently the registerMouseWidgets() is invoked in getKeyStroke() method only. This looks both expensive and (more importantly) doesn't disable/enable mouse cursor change on hover over a widget which is disabled/enabled.
> >
> > registerMouseWidgets() is built to:
> >
> > 1. on first call, send all widgets which match visible && enabled
> > 2. on subsequent calls, send only new or dead widgets (matching the same criteria). If there are no changes in the widgets, then this will be a no-op.
> >
> > This is not supposed to send to the JS side all the widgets, every time is called. Or are you referring that the Java-side execution of registerMouseWidgets() is expensive?
>
> Yes, I mean Java-side processing.

The alternative would be to capture all widgets with changed state since the java client has last accessed the JS side. But I don't think this is easily possible - we would need a parallel mechanism, similar to what we do now for sending to the server-side changes in widget state.

> But regardless of the cost I've noticed that the information about disabled widgets is not properly propagated to the browser.

OK, please fix it then. What scenario fails?

#142 - 01/28/2016 01:25 PM - Igor Skornyakov

LOAD-VOUSE-POINTER for the WebGUI is essentially done (including file-based cursors).

Committed to the task branch 2565a revno 10999.

However the processMouseMove in the p2j.mouse.js needs some fixes: it erroneously invokes ouseExit on the frame enter from the enclosing window and vice-versa. The processWindowBorder method relies on the cursor name which it not safe. It is also necessary to add custom cursor support for the Window widget.

#143 - 01/30/2016 04:08 PM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10967. The latest revision is now 11006.

#144 - 01/31/2016 05:01 PM - Igor Skornyakov

Refactored GuiWebDriver.registerMouseWidgets()

Committed to task branch 2565a revno 11007.

#145 - 02/02/2016 11:03 AM - Igor Skornyakov

- File mouse.p added

When running converted mouse.p (attached) the FRAME and RADIO-SET are registered with (non-visible) WINDOW with id = 1 while the rest are correctly registered with WINDOW with id = 44 (title "Window").
This doesn't (?) cause any issues with Swing GUI but is bad for a Web one as widgets' registration is window-based.
Investigating.

#146 - 02/02/2016 12:13 PM - Igor Skornyakov

The issue mentioned in the note 145 is fixed.

#147 - 02/02/2016 05:57 PM - Igor Skornyakov

With the mouse.p (see note 145) the DISABLE fi IN FRAME fr. results in the FILL-IN re-creation with new id. This results in the incorrect mouse listener de-registration.

#148 - 02/03/2016 06:40 AM - Igor Skornyakov

The issue mentioned in the previous note was a result of the syntax error: DISABLE fi IN FRAME fr. instead of DISABLE fi WITH FRAME fr. and a missing IN WINDOW clause in the ENABLE statement.

#149 - 02/03/2016 09:08 AM - Greg Shah

Please make a list of all remaining work on this task and post it here.

#150 - 02/03/2016 09:22 AM - Igor Skornyakov

Most of the functionality is working in WebGUI. The remaining tasks are:
1. Lock mouse cursor while drop-down is active (will be finished in an hour or two).
2. SET-WAIT-STATE - will be done today.
3. Fix cursor change on hovering over the WINDOW border - will be fixed today or tomorrow.
4. Custom mouse pointer support for the BROWSE (both SwingGUI and WebGUI).

#151 - 02/03/2016 09:56 AM - Greg Shah

> 4. Custom mouse pointer support for the BROWSE (both SwingGUI and WebGUI).

I want to get 2565a tested and merged to trunk **before** the browse pointer work is done. Please create a separate task for that and link it here as a related task.

#152 - 02/03/2016 05:07 PM - Igor Skornyakov

The runtime support of LOAD-MOUSE-POINTER and SET-WAIT-STATE for WebGUI is mostly finished.

Committed to the task branch 2565a revno 11010.

The remaining issues are:

1. Fix the mouse cursor change on the hovering over the window border,
2. Fix the support for SIMPLE and DROP-DOWN-LIST COMBO-BOX.

#153 - 02/05/2016 05:21 AM - Igor Skornyakov

Section 27 has been updated.

#154 - 02/10/2016 02:28 PM - Igor Skornyakov

Fixed LOAD-MOUSE-POINTER support for the COMBO-BOX (Swing GUI).

Committed to the task branch 2565a revno 11012.

#155 - 02/10/2016 02:36 PM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10970. The latest revision is now 11015.

#156 - 02/11/2016 11:25 AM - Eugenie Lyzenko

Hi Eugenie,
I'm working now on the runtime support for the custom mouse cursor (LOAD-MOUSE-POINTER method). At this moment the only remaining issue (except a minor one regarding hovering over a window border in WebGUI) is COMBO-BOX support.

I have to track MOUSE_ENTERED and MOUSE_EXITED events for the drop-down part of this widget. The source of this event is ScrollableSelectionListGuiImpl which doesn't have config (hence, no ID). For such widgets the mouse action cannot be registered. I tried to assign a config (and ID) to ScrollableSelectionListGuiImpl in a way it is done for the EntryFieldGuiImpl in ComboBoxGuiImpl. However after this change the selection via the mouse click stops working correctly (nothing happens until the focus is moved from the COMBO-BOX).

I tried to understand why it happens and fix the issue but have no success so far. Another way I have in mind is to to change logic in the ThinClient.processAction() with ad hoc hook but it looks ugly and fragile.

Can you please provide some advise?

#157 - 02/11/2016 11:26 AM - Eugenie Lyzenko

Hi Igor,

The mouse events for drop-down's ScrollableSelectionListGuiImpl component is routed from ScrollPaneGuiImpl. These two classes has parent<->child relationship. Meaning the viewport().getScrollWidget() for ScrollPaneGuiImpl is ScrollableSelectionListGuiImpl. When the mouse event is coming to ScrollPaneGuiImpl it is routed to respective ScrollableSelectionListGuiImpl if it is a child for given ScrollPaneGuiImpl. See the ScrollPaneGuiImpl.java for details. This allow us to work with mouse events without having config for ScrollableSelectionListGuiImpl.

This is working now for mouseMoved() and mouseClicked(). You need to add the same routing for mouseEntered() and mouseExited(). And of course need to add:

```
            MouseEvent.MOUSE_ENTERED,
            MouseEvent.MOUSE_EXITED
```

to the list of events returning by mouseActions() method of the ScrollPaneGuiImpl class or the events will not coming to ScrollPane.

#158 - 02/11/2016 11:26 AM - Eugenie Lyzenko

I will take a look. Please note however that for all widgets except drop-down part of the COMBO-BOX mouse hover event are activated in MouseHandler.processAction() and use ID-based registration (in my original mail I've erroneosly mentioned ThinClient.processAction(), sorry) . The MOUSE_CLICK event is activated in different way (from MouseHandler.applyMouseEvent()). At this moment this look confusing for me.

#159 - 02/11/2016 11:28 AM - Eugenie Lyzenko

Well. it seem that the problem is in the AbstractContainer. findMouseSource method. Its implementation seems a little bit inconsistent:
The statement

```
         return (found != null) ? found :
                (w.getId() == null ? null : w);
```

checks the 'w' for non-null ID but not the 'found'. This is because the source of the MOSE_MOVE event is ScrollableSelectionListGuiImpl which has no ID. There is an obvious fix, but I'm not sure how safe is it.
What do you think about this?

#160 - 02/11/2016 11:28 AM - Eugenie Lyzenko

1. The discussion become too deep inside the code logic to be private. Please move it to the Redmine, appropriate task you are working on.
2. I do not see the code you mentioned in AbstractContainer.findMouseSource method(). What exact line you are referring to?
3. I do not understand what the obvious fix idea is. The call for found.getId() when found == null will cause NPE in the statement below. So this looks correct, if w is valid.

#161 - 02/16/2016 08:42 AM - Greg Shah

-------- Forwarded Message --------
Subject: Re: Mouse cursor
Date: Wed, 10 Feb 2016 13:26:26 -0500
From: Igor ...
To: Constantin ..., Hynek ..., Eugenie ...

Thank you Constantin.
I suspected something like this but the breakpoint at
java.awt.Component.setCursor() was not activated. I will double check.
Thanks again,
Igor.

On 02/10/2016 01:24 PM, Constantin Asofiei wrote:

> Igor,
>
> Please check the MouseMoveavble and MouseResizeable classes. Now the drop-down is a overlay window (as I understand its a Swing window for the Swing client) and the drop-down may incorrectly have registered these, for resize or move support, at the driver level; we don't need movable or resizeable support for the overlay windows.
>
> Thanks,
> Constantin
>
> On 2/10/2016 8:21 PM, ias wrote:
>
>> Gentlemen,
>> Do we change mouse cursor anywhere in Swing GUI bypassing the java.awt.Component.setCursor() method? I'm struggling with LOAD-MOUSE-SUPPORT for the COMBO-BOX. Although the code seems to change the cursor correctly (according to printouts) each time the mouse enters the drop-down part the cursor is changed to a default one.
>> Please provide any clue.
>> Thank you,
>> Igor.

#162 - 02/16/2016 08:43 AM - Greg Shah

-------- Forwarded Message --------
Subject: Re: Mouse cursor
Date: Wed, 10 Feb 2016 13:38:27 -0500
From: Igor ...
To: Constantin ..., Hynek ..., Eugenie ...

Nope,
Drop-down is not registered as MouseMovable/MouseResizabe. Actually I've added printouts to these classes when they change cursor as the logic was changed a little to support custom cursors.
So the mystery remains.
Igor.

#163 - 02/16/2016 08:44 AM - Greg Shah

-------- Forwarded Message --------
Subject: Re: Mouse cursor
Date: Wed, 10 Feb 2016 13:55:42 -0500
From: Igor ...
To: Constantin ..., Hynek ..., Eugenie ...

Thank you Hynek,
I've just realized this and now changing the logic accordingly.
Igor.

On 02/10/2016 01:53 PM, Hynek Cihlar wrote:

> Igor,
>
> the combo-box and the drop-down are different components attached to different awt windows. Do you set the cursor for both?
>
> Hynek
>
> On 10.2.2016 19:21, ias wrote:
>
>> Gentlemen,
>> Do we change mouse cursor anywhere in Swing GUI bypassing the java.awt.Component.setCursor() method? I'm struggling with LOAD-MOUSE-SUPPORT for the COMBO-BOX. Although the code seems to change the cursor correctly (according to printouts) each time the mouse enters the drop-down part the cursor is changed to a default one.
>> Please provide any clue.
>> Thank you,
>> Igor.

#164 - 02/17/2016 08:06 AM - Greg Shah

Igor: Please provide some details on the state of this task. I know you are working on some problems with combo-box mouse pointers in the web client. Is there anything else left to do?

#165 - 02/17/2016 08:41 AM - Igor Skornyakov

Greg Shah wrote:

> Igor: Please provide some details on the state of this task. I know you are working on some problems with combo-box mouse pointers in the web client. Is there anything else left to do?

I hope to finish with COMBO-BOX today. After that the only remaining issue will be cursor change on the window border. It should be difficult

#166 - 02/17/2016 10:08 AM - Greg Shah

> After that the only remaining issue will be cursor change on the window border. It should be difficult

Please create a separate task for this. We may have someone else work it. Do not work on the window border issue now.

Is it reasonable to expect that you will have a clean, review-ready, functionally finished task branch 2565a by the end of the day?

I expect that 2565a will close #2565, #2566, #2935 and #2995. The only remaining work will be #2858, #2977 and the new task you will create for the window border custom pointer issue (LE: #2996). Am I correct?

#167 - 02/17/2016 10:13 AM - Igor Skornyakov

Greg Shah wrote:

> Please create a separate task for this. We may have someone else work it. Do not work on the window border issue now.

OK.

> Is it reasonable to expect that you will have a clean, review-ready, functionally finished task branch 2565a by the end of the day?

I think so.

> I expect that 2565a will close #2565, #2566, #2935 and #2995. The only remaining work will be #2558, #2977 and the new task you will create for the window border custom pointer issue. Am I correct?

Yes, this is correct

#168 - 02/17/2016 02:30 PM - Igor Skornyakov

Custom mouse pointer support for the COMBO-BOX (WebGUI) finished.

Committed to the task branch 2565a revno 11023.

Working on Javadoc/history/cleanup

#169 - 02/17/2016 05:14 PM - Igor Skornyakov

Cleanup finished.
Committed to the task branch 2565a revno 11026

Ready for the code review

#170 - 02/18/2016 05:37 AM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10972. The latest revision is now 11028.

#171 - 02/18/2016 06:41 AM - Igor Skornyakov

Re-tested after rebase. Added custom mouse pointer support for the SLIDER widget.

Committed to the task branch 2565a revno 11029

#172 - 02/18/2016 07:41 AM - Greg Shah

Code Review Task Branch 2565a Revision 11026

This is part 1 of my code review. I want you to fix these things very quickly, since this work has been going on for over 3 months and it is long overdue. I will continue with the rest of the code review which I will post separately.

1. Most issues in note 121 (a code review) above have not been resolved. Items 1, 2 and 5 all seem to have been largely ignored. Also, I think the request for more comments in the code needs to be also addressed in the ThinClient changes.

2. LogicalTerminal.getWaitState() is still unimplemented.

3. Doesn't LogicalTerminal.setWaitState() need to do its string matching on a case-insensitive basis? It would be surprising if the 4GL treated this differently than all other strings.

4. Please explain the logic behind duplicating the mousePointer member in so many of the *Config classes (and thus having to implement Hoverable in the corresponding *Widget classes. Is there a better class in which to place that code?

5. Are triggers disabled (or disabled in the same way) for all types of wait-state? For example, do GENERAL or COMPILER or custom wait-states act differently?

6. Using multiple nested ternary expressions like the following in ServerImageWorker makes the code harder to read:

```
public static String getResourceName(JarClassLoader jcl, String name, boolean caseSens)
{
  return !"".equals(Utils.getExt(name)) ?
           jcl.getResourceName(name, caseSens) :
           Utils.tryWhileNull(caseSens ? ImageExt.EXT_VARIANTS : ImageExt.EXT_BASES,
                 ext -> jcl.getResourceName(name.concat(ext), caseSens));
}
```

I believe formatting it like this would be significantly better:

```
   public static String getResourceName(JarClassLoader jcl, String name, boolean caseSens)
   {
      if (!"".equals(Utils.getExt(name))
      {
         return jcl.getResourceName(name, caseSens);
      }

     return Utils.tryWhileNull(caseSens ? ImageExt.EXT_VARIANTS : ImageExt.EXT_BASES,
                               ext -> jcl.getResourceName(name.concat(ext), caseSens));
   }
```

The reason is that it takes much more time to read and understand the logic of your version than for my version.

7. What was the scenario that required the Search.java modification?

8. I am still finding a significant number of coding standards problems. It seems to me that you are not doing a review of the code (bzr diff -r10971 --using meld) before submitting it for review by me. Why do I have to find and request all these be fixed? It is very time consuming for me to do this. I don't understand why you are writing sloppy code in the first place. Write the code properly the first time so that you (and I) don't have to clean this up later. Some items:

build.xml has a change that is just a debugging option that has been commented. Please revert that change.

Please remove the extra blank line you have added at the end of LogicalTerminal (between the moveToBottom() method and the final closing curly brace of the class). This is not allowed, per our coding standards.

Search.java is missing a history entry.

The javadoc for the BrowseColumnConfig default constructor has been misformatted.

The history entries for BrowseColumnWidget and ControlSetConfig have their text mis-aligned.

#173 - 02/18/2016 08:25 AM - Igor Skornyakov

Thank you Greg,
I'm working on your comments right now.

#174 - 02/18/2016 09:01 AM - Igor Skornyakov

Greg Shah wrote:

> 1. Most issues in note 121 (a code review) above have not been resolved. Items 1, 2 and 5 all seem to have been largely ignored. Also, I think the request for more comments in the code needs to be also addressed in the ThinClient changes.

Greg,
1. Regarding the place where cursor should be created. As I've mentioned in note 127 my approach is essentially the same as is used for images where the driver creates an internal representation it will work with. You've agreed (in note 129) but wrote that for images is done in a common part for both Swing and Web GUI.
Please note however that unlike the situation with images cursors from files are very different for Swing and Web GUI so it seems logical to place different implementations of the createMousePointer(InputStream is, String ptrName) to the corresponding drivers. The createMousePointer(CursorType type) implementation is the same for both and it is in the AbstractDriver now (it was erroneously duplicated in the GuiWebDriver and I've removed it now).

2. I've also reduced the number of methods in the ClientExports to just two - for pre-defined cursor and for the one loaded from file.

3. Regarding the comment 5 in the note 121. I've changed the GenericWidget.loadMousePointer while working on the first code review and you seemed to be satisfied with these changes. I haven't changed it since that.

As you remember we agreed that I will adjust program logic after the functionality required for the Web GUI will be clear. Right now the only difference in the internal cursor representation is in a single method which implementation is different for Swing and Web GUI simply because they are really very different (AWT cursor vs raw bytes). I cannot agree that items 1, 2 and 5 all from note 121 have been largely ignored.

#175 - 02/18/2016 10:00 AM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11026
> 4. Please explain the logic behind duplicating the mousePointer member in so many of the *Config classes (and thus having to implement Hoverable in the corresponding *Widget classes. Is there a better class in which to place that code?

This was done because not all widgets support LOAD-MOUSE-POINTER method. For example EDITOR and FILL-IN do support while TEXT and Literal widgets do not. When possible I tried to put the functionality as high in the hierarchy as possible (see e.g. ControlEntity).

> 5. Are triggers disabled (or disabled in the same way) for all types of wait-state? For example, do GENERAL or COMPILER or custom wait-states act differently?

Sorry - I do not understand the question. All the UI is locked while the application is in the wait-state. I haven't noticed any difference (except the cursor shape).

> 6. Using multiple nested ternary expressions like the following in ServerImageWorker makes the code harder to read:

Fixed

> 7. What was the scenario that required the Search.java modification?

I've experienced NPEs at the beginning of my attempts to start working with the Web GUI. And added a check for null.

> build.xml has a change that is just a debugging option that has been commented. Please revert that change.

Done.

> Please remove the extra blank line you have added at the end of LogicalTerminal (between the moveToBottom() method and the final closing curly brace of the class). This is not allowed, per our coding standards.

Done.

> Search.java is missing a history entry.

Done.

> The javadoc for the BrowseColumnConfig default constructor has been misformatted.

Done.

> The history entries for BrowseColumnWidget and ControlSetConfig have their text mis-aligned.

Done.

Committed to the task branch 2565a revno 11030

#176 - 02/18/2016 10:03 AM - Igor Skornyakov

Committed to the task branch 2565a revno 11031

#177 - 02/18/2016 10:08 AM - Greg Shah

You're right that some changes have been made. But I am concerned about the core approach. There are comments that I have previously made, which have not yet been addressed.

1. The cursor loading and processing should be as hidden as is possible on the client side. Yet we have server-side code like MousePointer that has code in it that directly references client driver methods! This is not appropriate. The server code should (ideally) have NO references to any of the client packages. But it most definitely should not be referencing the drivers, which are the most low-level, implementation-specific portions of the client.

The server code only needs to deal with 3 things:

- The text name (which can be a filename or a predetermined cursor name) of any non-default cursor being specified.
- Whether the text name is a predetermined name.
- Any byte[] of a custom cursor that was found in the application jar on the server.

Everything else should be handled on the client. Today, you have too much of the implementation on the server.

2. I have stated before (notes 3, 111, 121) that a custom cursor filename should be loaded from the application JAR (on the server) if it exists there. This is done in our current image loading (see LT.getImageStreamFromApplication()) but you do not have this implemented.

3. I still think there are not enough comments in the server-side load/set pointer code.

#178 - 02/18/2016 10:13 AM - Greg Shah

> > 4. Please explain the logic behind duplicating the mousePointer member in so many of the *Config classes (and thus having to implement Hoverable in the corresponding *Widget classes. Is there a better class in which to place that code?
>
> This was done because not all widgets support LOAD-MOUSE-POINTER method. For example EDITOR and FILL-IN do support while TEXT and Literal widgets do not. When possible I tried to put the functionality as high in the hierarchy as possible (see e.g. ControlEntity).

If the list of widgets that have the support is larger than the list of widgets that do not support it, then it would be better to push the functionality further up the hierarchy and just override the default for those widgets that don't support it. That override can raise an appropriate error. Would that reduce the amount of code here?

> > 5. Are triggers disabled (or disabled in the same way) for all types of wait-state? For example, do GENERAL or COMPILER or custom wait-states act differently?
>
> Sorry - I do not understand the question. All the UI is locked while the application is in the wait-state. I haven't noticed any difference (except the cursor shape).

Noticing is one thing, and testing is another. Have you tried to cause a trigger to fire (in the 4GL) with different wait-state cursors? If not, please confirm through testing that they all have the same behavior.

#179 - 02/18/2016 10:23 AM - Igor Skornyakov

Greg Shah wrote:

> > > 4. Please explain the logic behind duplicating the mousePointer member in so many of the *Config classes (and thus having to implement Hoverable in the corresponding *Widget classes. Is there a better class in which to place that code?
> >
> > This was done because not all widgets support LOAD-MOUSE-POINTER method. For example EDITOR and FILL-IN do support while TEXT and Literal widgets do not. When possible I tried to put the functionality as high in the hierarchy as possible (see e.g. ControlEntity).
>
> If the list of widgets that have the support is larger than the list of widgets that do not support it, then it would be better to push the functionality further up the hierarchy and just override the default for those widgets that don't support it. That override can raise an appropriate error. Would that reduce the amount of code here?

I'm sure that this approach will not reduce the amount of code simply because the number of widgets which do not support custom mouse pointer is much bigger than the number of ones which support. It is possible however to put a corresponding field to the BaseConfig which will slightly reduce the code in its descendants.

> > > 5. Are triggers disabled (or disabled in the same way) for all types of wait-state? For example, do GENERAL or COMPILER or custom wait-states act differently?
> >
> > Sorry - I do not understand the question. All the UI is locked while the application is in the wait-state. I haven't noticed any difference (except the cursor shape).
>
> Noticing is one thing, and testing is another. Have you tried to cause a trigger to fire (in the 4GL) with different wait-state cursors? If not, please confirm through testing that they all have the same behavior.

Sorry, previously I've fired triggers only via UI which is completely blocked in a wait-state. What is the typical situation when the trigger is activated in another way?

#180 - 02/18/2016 10:35 AM - Igor Skornyakov

I've added the block of triggers as SET-WAIT-STATE causes deactivation of the active drop-down. I had a trigger on this event and it was fired in p2j but not in 4GL. This behaviour was the same for all wait states.

#181 - 02/18/2016 10:48 AM - Igor Skornyakov

Greg Shah wrote:

> 1. The cursor loading and processing should be as hidden as is possible on the client side. Yet we have server-side code like MousePointer that has code in it that directly references client driver methods! This is not appropriate. The server code should (ideally) have NO references to any of the client packages. But it most definitely should not be referencing the drivers, which are the most low-level, implementation-specific portions of the client.
>
> The server code only needs to deal with 3 things:
>
> - The text name (which can be a filename or a predetermined cursor name) of any non-default cursor being specified.
> - Whether the text name is a predetermined name.
> - Any byte[] of a custom cursor that was found in the application jar on the server.
>   Everything else should be handled on the client. Today, you have too much of the implementation on the server.
>
> 2. I have stated before (notes 3, 111, 121) that a custom cursor filename should be loaded from the application JAR (on the server) if it exists there. This is done in our current image loading (see LT.getImageStreamFromApplication()) but you do not have this implemented.

But in fact GenericWidget.loadLousePointer does exactly what you describe (and no more). It doesn't load cursor data. Initially I did it but as far as I remember we agreed that this should be done only for pre-defined cursors which do not have AWT/JS counterparts. The question about such cursors was postponed. When we'll have them then instead of UnimplementedFeature.missing there will be code for loading cursor data at the server side.

> 3. I still think there are not enough comments in the server-side load/set pointer code.

I've added more comments.

Committed to the tsak branch 2565a revno 11032.

#182 - 02/18/2016 11:49 AM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11026
> 8. I am still finding a significant number of coding standards problems. It seems to me that you are not doing a review of the code (bzr diff -r10971 --using meld) before submitting it for review by me. Why do I have to find and request all these be fixed? It is very time consuming for me to do this. I don't understand why you are writing sloppy code in the first place. Write the code properly the first time so that you (and I) don't have to clean this up later. Some items:

I've walked again through all my changes and indeed found a number of formatting issues (extra/missing spaces and new lines). It looks like I just do not see them. Sorry. May be it makes sense to formalize our rules and use something like checkstyle?

Committed to the task branch 2565a revno 11033.

#183 - 02/18/2016 12:49 PM - Greg Shah

Code Review Task Branch 2565a Revision 11031

This is part 2 of the code review.

1. You have put GUI-specific code into the ComboBox common code. This code is shared by ChUI and GUI, so GUI-specifics are not appropriate there. If necessary, create empty methods that ComboBoxGuiImpl will override with the proper functionality. If you have to import com.goldencode.p2j.ui.client.gui.*; in that file, there is a problem.

2. The import com.sun.swing.internal.plaf.synth.resources.*; should not be referenced in ComboBox.

3. ComboBox.getDropDownList() is missing javadoc.

4. DropDown.java is missing a history entry.

5. Search.java has the wrong initials and the wrong date in the history entry.

6. Is import java.io.*; necessary in GenericWidget?

7. Please put a blank line in between all members. This problem can be seen in ImageExts, MousePointer, BinaryMousePtrWrapper.

8. Please add a task to Redmine (not a sub-task for #2565, just a related task) for adding cursor support for those cursors that are not yet supported.

9. Should we be dropping all input events that occur when wait-state is on or should they queue up? Today, you disable triggers in TC and you "eat" keys in SwingKeyboardReader. But there is some behavior that will execute regardless of disabling triggers. That seems incorrect. I think we should probably be discarding events very early on and avoiding any processing at all. Again, this should be confirmed via testing.

10. Why is import com.goldencode.p2j.ui.client.gui.driver.*; being added to LabeledDataContainer. That is common code, not GUI.

11. UiUtils.lookup4GLWidgetId() could have been done with less code. Why not create a variant of lookupWidgetId(Widget<>, boolean) where the 2nd parameter forces the elimination of virtual widgets (ones with -1 as an ID)? This core worker would allow common code with lookupWidgetId(Widget<>) which now would just call the worker with false as the 2nd arg. The same thing would occur for lookupWidgetIdAsInt().

12. MousePtrWrapper has all implements clauses on the same line as the class statements.

13. PreDefinedWebMousePtrWrapper data member is missing javadoc.

14. PreDefinedWebMousePtrWrapper.setCursor() should have a blank line between it and the constructor that precedes it.

15. MousePtrWrapper the end of the file has this issue:

```
      }

}
```

16. WidgetFactory and WidgetFactoryAdapter should not have direct references to GuiOutputManager, they are common code. If you have to import com.goldencode.p2j.ui.client.gui.driver.*; in those files, there is a problem.

17. I see that you have included 3rd party code in p2j.socket.js. You must NEVER do this without approval in advance. All 3rd technology is minimized in our project and Eric or myself must review each possible inclusion for license compatibility and other criteria. We have contracts with customers that could be broken by including the wrong 3rd party technology. That is why we have made this part of our coding standards, why we have explicit rules for intellectual property as described in our developer handbook and why we have very specific clauses in all of our contractor agreements.

18. In the case of this base64 3rd party code you added, I don't think it is needed. We already use window.btoa() in p2j.sockets.js for the same purpose.

19. DropDownGuiImpl is missing a history entry.

20. Why did you make the EditorGuiImpl.setEnabled() and FillInGuiImpl.setEnabled() changes? This seems dangerous.

21. FrameGuiImpl should not import java.awt.event.*;.

22. In GuiWidgetFactory, you have deleted a blank line that was intentionally placed there to make it easier to read the code:

Your version:

```
import com.goldencode.p2j.util.*;
// explicit import to resolve conflict with com.goldencode.p2j.util.Text
import com.goldencode.p2j.ui.client.Text;
```

original version:

```
import com.goldencode.p2j.util.*;

// explicit import to resolve conflict with com.goldencode.p2j.util.Text
import com.goldencode.p2j.ui.client.Text;
```

If you remove the line, it is harder to see that there is a comment there. This is an example of whitespace making something more readable (at least for most developers).

23. ModalWindow is missing a history entry.

24. OverlayWindow is missing a history entry.

25. There is an extra blank line between createMouseHoverAction() and mouseActions() in ScrollPaneGuiImpl.

26. ScrollPaneGuiImpl should not be directly referencing potential contained widget types. Please remove the import com.goldencode.p2j.ui.ComboBoxConfig.*;.

27. ScrollPaneGuiImpl data member wnd is missing javadoc.

28. ScrollableSelectionListGuiImpl should not import com.goldencode.p2j.ui.ComboBoxConfig.Mode;. If this needs to be referenced from multiple classes, then it should not be in code that is specific to combo-box.

29. ScrollableSelectionListGuiImpl has two blank lines instead of one between getEffectiveMousePointer() and getLoadedMousePointer().

30. SliderGuiImpl is missing a history entry.

31. ToolTip is missing a history entry.

32. Are you sure that setCursor() code (2 places) in WindowGuiImpl.init() is safe enough that try/finally is not needed?

33. The data member and constructor of CursorType are missing javadoc.

34. There is an extra blank line at the end of CursorType.

35. In CursorType, please do not import all of awt. Just import Cursor. And put a comment there that this is intentionally being imported in common code. Something like this:

```
// this usage of AWT Cursor is intentional, it is being used separately from the
// rest of AWT/Swing; no other AWT/Swing usage is occurring in this class
```

36. In GuiDriver.createMousePointer(), I think it is better to use a byte[] instead of InputStream. This will be compatible with server-loaded cursors from the application jar. It also allows us to remove the dependency on java.io.*.

37. Please reinsert the blank line you removed above MouseWidgetAction.getEventId().

38. We implemented the image support for the GUI client in a way that avoided putting AWT image and ImageIO support into SwingGuiDriver and GuiWebDriver, but you have added these dependencies. That is not a preferred approach. Is there a cleaner way?

39. GuiWebDriver should not import com.goldencode.p2j.ui.client.gui.driver.swing.*;.

40. GuiWebDriver has 2 history entries for this same task branch.

41. GuiWebDriver.isInLastMouseWidgets() exceeds the line length limit.

42. Your expansion of the use of Streams in GuiWebDriver.registerMouseWidgets() has made it very hard for me to determine the validity of your logic changes. How thoroughly have you tested that code?

43. I'm trying to understand why we are directly referencing Cursor in GuiDriver and EmulatedWindowState, considering this is also used in GuiWebDriver and GuiWebEmulatedWindow (which has an empty setCursor() method). I think the approach is not doing enough to hide the implementation at the driver level and we are left with AWT references in lots of places where we don't want them. The point of having a pointer wrapper class is to use that class to hide the implementation details for the driver.

44. Why is GuiWebEmulatedWindow.setCursor() an empty implementation and what are the implications for this?

45. GuiWebSocket.BinaryMessage has missing blank lines between data members.

46. GuiWebSocket.BinaryMessage needs a black line added after the closing curly brace.

47. MouseHoverAction has implements on the same line as class.

48. MouseHoverAction is missing blank lines between data members.

49. MouseHoverAction.setCursor() should probably have the gd.releaseWindow() in a finally block.

50. p2j.screen.js has two history entries for the same task branch.

51. p2j.screen.js needs javadoc on the mouseCursors member.

52. Please add back the blank line you removed after AbstractWidget.getMenuMouse().

53. The reference to ScrollPaneGuiImpl in AbstractWidget.getActualBounds() should not be there. ScrollPaneGuiImpl is a GUI reference that should not be in common code. ScrollPaneGuiImpl is also a subclass of AbstractWidget. It is a bad practice to directly reference child classes from the parent class. This same problem also exists in AbstractWidget.enclosingFrame() which has references to both DropDown and ComboBox.

54. FileSystemDaemon.WIN_FILE_SEPARATOR is missing javadoc.

55. Utils.InstrumentedMap has an extra following blank line.

#184 - 02/18/2016 12:50 PM - Greg Shah

Sergey: Please review the web client changes in 2565a.

Constantin: Please review the mouse processing changes in 2565a.

Eugenie: Please review the FileSytemDaemon changes in 2565a.

#185 - 02/18/2016 02:10 PM - Greg Shah

> I've added the block of triggers as SET-WAIT-STATE causes deactivation of the active drop-down. I had a trigger on this event and it was fired in p2j but not in 4GL. This behaviour was the same for all wait states.

Good. Yes, that was what I was asking you to check.

#186 - 02/18/2016 02:21 PM - Greg Shah

> But in fact GenericWidget.loadLousePointer does exactly what you describe (and no more).

You are processing on the server using the MousePointer class, which has deep knowledge of the client and the client driver infrastructure. I am requesting that the server-side only know the 3 things above.

> as far as I remember we agreed that this should be done only for pre-defined cursors which do not have AWT/JS counterparts. The question about such cursors was postponed. When we'll have them then instead of UnimplementedFeature.missing there will be code for loading cursor data at the server side.

No, I think there is a misunderstanding here. When I talk about the "application jar", I am not referring to pre-defined cursors in the p2j.jar (which should only be read on the client side). The "application jar" is the converted application code that is being executed in the P2J server. If you are building something in testcases/uast/, this would be the testcases.jar that is built after converting the sample code.

Consider that it is common to ship GUI applications with graphics resources. For example, an application might have a toolbar (using buttons that have images loaded instead of text). Or the application may have custom pointers that it uses. Although in the 4GL, the application must always search the client filesystem (via propath) to load these, in P2J we are offering an improvement: to store these in the converted application's jar file and load them from there.

This must be done on the server because that jar doesn't exist on the client. And it is a deployment improvement over the 4GL because one doesn't have to copy all these graphics resources to every client system for the app to work properly. We already support this for images using LT.getImageStreamFromApplication(). You need to add support for this same idea for cursors. The resulting cursor would be sent to the client as a byte[] instead of a MousePointer or MousePtrWrapper.

#187 - 02/18/2016 02:43 PM - Sergey Ivanovskiy

Igor, there are lot of changes, do you know that in the current 2565a the following function Window.prototype.isParentOf defined in module p2j.screen.js is not used

```
   /**
    * Check if one hoverable widget is a parent of the another
    *
    * @param  {Number} pid
    *          potential parent widget  id
    * @param  {Number} wid
    *          widget id
    * @returns true if the first widget is the parent of the second one
    */
   Window.prototype.isParentOf = function(pid, wid)
   {
      if (!pid)
      {
         return false;
      }
```

```


var widget = this.getHoverableWidget(wid);
      if (!widget)
{
         return false;
      }
      if (pid == wid.pid)
{
         return true;
      }
      return this.isParentOf(pid, widget.pid);
   }
```

#188 - 02/18/2016 02:51 PM - Igor Skornyakov

Sergey Ivanovskiy wrote:

> Igor, there are lot of changes, do you know that in the current 2565a the following function Window.prototype.isParentOf defined in module p2j.screen.js is not used

Thank you Sergey. I will remove it.

#189 - 02/18/2016 03:06 PM - Sergey Ivanovskiy

If we consider p2j.socket.js, then we can find two methods to transform to base64 encoding. The first method is implemented by embedded fonts and the second one uses this function   function uint8ToBase64 (uint8). Is it correct to have two methods to encode to base64? Could you explain it?

#190 - 02/18/2016 03:12 PM - Igor Skornyakov

Sergey Ivanovskiy wrote:

> If we consider p2j.socket.js, then we can find two methods to transform to base64 encoding. The first method is implemented by embedded fonts and the second one uses this function   function uint8ToBase64 (uint8). Is it correct to have two methods to encode to base64? Could you explain it?

I have a little experience with JS but I've seen at least two places on Stackoverflow where people strongly to not recommend using btoa. If you believe that there is nothing wrong with this function we can get rid of the custom implementation.

#191 - 02/18/2016 03:13 PM - Sergey Ivanovskiy

It is okay, I only want to select the best, it should be your solution.

#192 - 02/18/2016 03:40 PM - Sergey Ivanovskiy

Let us consider two methods to set a custom cursor:
1) The first we use now

```
Window.prototype.setCursorStyle = function(style, ignoreLockedPtr)
{
   if (!ignoreLockedPtr)
   {
      var es = p2j.screen.lockedMousePtr();
      if (es != null)
      {
         style = es;
      }
   }
   this.canvas.style.cursor = style;
};
```

where style can be this large string url(....).
2) The second method is similar to embedded fonts
a) create new style element and set cursor property
b) give new name for this class if it has not been loaded before
c) use this name as an id this.canvas.class += (" " + id);
I don't know what is the best but the second one uses css and new cursor's classes are cached. May be the second is not a correct solution.

#193 - 02/18/2016 03:43 PM - Igor Skornyakov

Fixed the following issues from the code review part 2: 1-7, 10, 12-16, 21-35, 37, 39-41, 45,46, 48-52, 54, 55.
Regarding 47: There is no 'implements' keyword in the MouseHoverAction.

Committed to the task branch 2565a revno 11034.

#194 - 02/18/2016 03:46 PM - Sergey Ivanovskiy

Igor, please could you consider the second method due to the optimization. I am worried about the large string that we set each time if we need to change cursor. The note 192 is updated.

#195 - 02/18/2016 03:51 PM - Igor Skornyakov

Sergey Ivanovskiy wrote:

> Igor, please could you consider the second method due to the optimization. I am worried about the large string that we set each time if we need to change cursor. The note 192 is updated.

Thank you Sergey. As I wrote before I have little experience with JS. I've used the way which is most widely recommended in the Internet. As it is very simple I do not think that it makes sense to look for another one just for unification. My understanding is that font is a more complicated thing. After all I do not think that custom cursors are used very often in a real applications to try to squeeze the last drop of performance. Is the string assignment in JS is not just a pointer assignment?

#196 - 02/18/2016 04:01 PM - Sergey Ivanovskiy

I don't know, but the reason can be found if we answer this question what css engine can do with this.canvas.style.cursor = "url(...)"; and this.canvas.class="cursor1". Is it true that the first or the second one is more expensive or they take approximately the same time to be parsed. I think that css classes are prepared somehow by css engine.

#197 - 02/18/2016 04:03 PM - Greg Shah

As I noted in items 17 and 18 of my code review, we are going to use window.btoa() for our encoding. Please remove the 3rd party code.

#198 - 02/18/2016 04:04 PM - Igor Skornyakov

Sergey Ivanovskiy wrote:

> I don't know, but the reason can be found if we answer this question what css engine can do with this.canvas.style.cursor = "url(...)"; and this.canvas.class="cursor1". Is it true that the first or the second one is more expensive or they take approximately the same time to be parsed. I

> think that css classes are prepared somehow by css engine.

Oh, now I understand your point. I think you're right. I will try to use the approach you suggested. Thank you.

#199 - 02/18/2016 04:04 PM - Igor Skornyakov

Greg Shah wrote:

> As I noted in items 17 and 18 of my code review, we are going to use window.btoa() for our encoding. Please remove the 3rd party code.

OK.

#200 - 02/18/2016 04:08 PM - Greg Shah

> Regarding 47: There is no 'implements' keyword in the MouseHoverAction.

I should have written extends.

#201 - 02/18/2016 05:09 PM - Igor Skornyakov

Fixed the following issues from the code review part 2: 17-19, 47, 53.

Committed to the task branch 2565a revno 11035.

#202 - 02/18/2016 05:54 PM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11031
>
> This is part 2 of the code review.
> 42. Your expansion of the use of Streams in GuiWebDriver.registerMouseWidgets() has made it very hard for me to determine the validity of your logic changes. How thoroughly have you tested that code?

Actually this refactoring is supposed to be (and I hope it is) 100% logically equivalent to the initial one. I did it just to understand what this code is doing. Of course it is a matter of my personal perception. I have tested it with my test program (mouse.p) which is quite complicated. Of course I can revert to the original version w/o any problems - just by replacing two methods with thire old versions. It is also possible to keep both versions for a while with invoking both of them and automatically comparing the results - I've used this approach before to check the correctness of refactoring a

complicated code.
However I feel that something is wrong with the very idea of this method - it seems to be very difficult to maintain in the future. Of course I can be wrong.

#203 - 02/18/2016 07:24 PM - Eugenie Lyzenko

> Eugenie: Please review the FileSytemDaemon changes in 2565a.

Well, the code was significantly rewritten from the syntax point of view. However I do not see any logical difference between this version and the previous one. And now the question - what is the purpose of this change? Do you have any optimization or add more features that was missed before or consider the cases that was omitted in previous version? I'm asking because what I can tell for sure is the code became much less readable. May be adding more comment for what is going on will be useful.

Again, I have no objection for the changes, I do not see the reason for the change if it actually changes nothing. Anyway - you will need to run testcases(at least all known at this time for the area the FileSytemDaemon is involved in) to find out whether this version of FileSytemDaemon works fine or may be has any regressions.

#204 - 02/19/2016 03:45 AM - Igor Skornyakov

Eugenie Lyzenko wrote:

> > Eugenie: Please review the FileSytemDaemon changes in 2565a.
>
> Well, the code was significantly rewritten from the syntax point of view. However I do not see any logical difference between this version and the previous one. And now the question - what is the purpose of this change? Do you have any optimization or add more features that was missed before or consider the cases that was omitted in previous version? I'm asking because what I can tell for sure is the code became much less readable. May be adding more comment for what is going on will be useful.
>
> Again, I have no objection for the changes, I do not see the reason for the change if it actually changes nothing. Anyway - you will need to run testcases(at least all known at this time for the area the FileSytemDaemon is involved in) to find out whether this version of FileSytemDaemon works fine or may be has any regressions.

Thank you Eugenie. The changes where needed for two reasons: 1.To fix some incompatibilities with Progress (see #2972) and 2. To make possible the re-use the code for processing cursor files.
I'm sorry that you find the new version of the code less readable but you understand that this is mostly a matter of a personal taste. For me it was not very easy to understand some places in the previous version. I've tried to make only minimal changes to the existing code.

#205 - 02/19/2016 05:50 AM - Constantin Asofiei

Igor, in rev 11035 ThinClient.trigger returns if wait-state is on. But this code in 4GL proves that triggers are executed. Press "a" to enable wait-state then "b". Although mouse operations don't work while wait-state is on, 4GL does respond to keyboard input (you can enter data in the fill-in, or press "space" on the button). The only keyboard events I think 4GL "consumes" is widget focus change (i.e. TAB doesn't work to focus on the button, if wait-state is enabled).

```
DEF VAR i AS INT.
DEF BUTTON btn LABEL "btn".

FORM i btn WITH FRAME f1.

ON 'a':U ANYWHERE
DO:
   MESSAGE SESSION:SET-WAIT-STATE("general").
END.

ON 'b':U ANYWHERE
DO:
   MESSAGE "here".
END.
UPDATE i btn WITH FRAME f1.
```

Does this work in P2J?

#206 - 02/19/2016 05:55 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, in rev 11035 ThinClient.trigger returns if wait-state is on. But this code in 4GL proves that triggers are executed. Press "a" to enable wait-state then "b". Although mouse operations don't work while wait-state is on, 4GL does respond to keyboard input (you can enter data in the fill-in, or press "space" on the button). The only keyboard events I think 4GL "consumes" is widget focus change (i.e. TAB doesn't work to focus on the button, if wait-state is enabled).
>
> [...]
>
> Does this work in P2J?

Thank you Constantin. I will double check. I think I've tested this functionality and found that it works like described in the Progress documentation.

#207 - 02/19/2016 06:08 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, in rev 11035 ThinClient.trigger returns if wait-state is on. But this code in 4GL proves that triggers are executed. Press "a" to enable wait-state then "b". Although mouse operations don't work while wait-state is on, 4GL does respond to keyboard input (you can enter data in the fill-in, or press "space" on the button). The only keyboard events I think 4GL "consumes" is widget focus change (i.e. TAB doesn't work to focus on the button, if wait-state is enabled).
>
> [...]
>
> Does this work in P2J?

You're right Constantin, thank you very much. In the wait-state only mouse operations and focus change are blocked. The keyboard operations on the focused widgets and keyboard-initiated triggers are enabled. I will change the p2j logic accordingly.

#208 - 02/19/2016 06:47 AM - Igor Skornyakov

More fixes according to the code review part 2:

> 11. UiUtils.lookup4GLWidgetId() could have been done with less code. Why not create a variant of lookupWidgetId(Widget<>, boolean) where the 2nd parameter forces the elimination of virtual widgets (ones with -1 as an ID)? This core worker would allow common code with lookupWidgetId(Widget<>) which now would just call the worker with false as the 2nd arg. The same thing would occur for lookupWidgetIdAsInt().

Done.

> 36. In GuiDriver.createMousePointer(), I think it is better to use a byte[] instead of InputStream. This will be compatible with server-loaded cursors from the application jar. It also allows us to remove the dependency on java.io.*.

Done.

> 38. We implemented the image support for the GUI client in a way that avoided putting AWT image and ImageIO support into SwingGuiDriver and GuiWebDriver, but you have added these dependencies. That is not a preferred approach. Is there a cleaner way?

Actually neither ava.awt.image nor javax.imageio are used in the GuiWebDriver. I've removed the corresponding imports. I've also remove the reference to ImageIO from the SwingGuiDriver. It is not looks feasible to get rid of reference to the java.awt.image as AWT-specific cursor creation depends on it.

Committed to the task branch 2565a revno 11037.

#209 - 02/19/2016 07:14 AM - Constantin Asofiei

About 2565a rev 11035:

1. ClientExports.setMousePointer - the javadoc is missing the parameter name for ptr.
2. CommonWidget.getMousePointer - you have an extra line after the method
3. EventList.merge - is missing javadoc for the return type.
4. FillinConfig.mousePointer - missing javadoc
5. in MousePointer.wrapper, why are you lowercasing the file name in ImageHelper.class.getResourceAsStream(fileName.toLowerCase());? Jar resources are case-sensitive...
6. ClientExports.setWaitState and getWaitState - please use Java Boolean and String types for the return - assume null is 4GL unknown, and let the caller (i.e. LogicalTerminal) construct the 4GL-compatible instance. Doing so, it avoids overhead; for example, logical P2J type serializes two boolean values instead of a single value.
7. SwingKeyboardReader.read() doesn't have to block if wait-state is on (see previous note 206/207).
8. ScrollPaneGuiImpl.mouseActions - why have you removed the MouseEvent.MOUSE_MOVED?
9. why make the MouseOps enum public? It has no reason to be exposed outside the gui.driver.web package
10. GuiWebSocket
    - the location of the BinaryMessage inner class is incorrect
    - registerHoverableWidget - javadoc doesn't contain all parameters, the parameters need to be column-aligned, one on each line
    - registerHoverableWidget - when you are computing message length like 1 + 4 + 4 + 4 + 4*2 + 2 + 1, please add comments to explain what every constant in the expression means (i.e. widget id, window id, etc). It's easier to follow if at some point the code needs to be changed.
    - registerHoverableWidget - you have NativeRectangle.width() and height() APIs - use this instead of i.e. bound.right() - bound.left() + 1 to compute the width/height.

Otherwise, the logic looks OK.

#210 - 02/19/2016 09:51 AM - Eugenie Lyzenko

> I'm sorry that you find the new version of the code less readable but you understand that this is mostly a matter of a personal taste.

It is not a problem et all, no reasons to apologize. Mostly I meant the using of the lambda java 8 feature. Yes, it is "elegant". But it is required from the reader to decode the -> usage to extended form or to find out what is the data type used. To do this(if you are not the code writer - you find the method used ...). And in general it is not a problem. But all of this takes a time before the reader starts to think about the main program logic.

I think adding the informative comments will significantly help to review such code, just think about it. No-one has to know what the people thinking about, it is better to tell.

#211 - 02/19/2016 09:55 AM - Igor Skornyakov

Eugenie Lyzenko wrote:

> I think adding the informative comments will significantly help to review such code, just think about it. No-one has to know what the people thinking about, it is better to tell.

I see. I will try to add comments in the future.

#212 - 02/19/2016 09:59 AM - Igor Skornyakov

More fixes according to the code review part 2:

> 20. Why did you make the EditorGuiImpl.setEnabled() and FillInGuiImpl.setEnabled() changes? This seems dangerous.

Both widgets had a logic for changing mouse cursor which is more generic now and is located in the AbstractWidget.

> 43. I'm trying to understand why we are directly referencing Cursor in GuiDriver and EmulatedWindowState, considering this is also used in GuiWebDriver and GuiWebEmulatedWindow (which has an empty setCursor() method). I think the approach is not doing enough to hide the implementation at the driver level and we are left with AWT references in lots of places where we don't want them. The point of having a pointer wrapper class is to use that class to hide the implementation details for the driver.

Fixed

> 44. Why is GuiWebEmulatedWindow.setCursor() an empty implementation and what are the implications for this?

Fixed

Committed to the task branch 2565a revno 11038.

Now I start working on Constantin's comments (and Greg's comment #9 from the part 2)

#213 - 02/19/2016 11:12 AM - Igor Skornyakov

Constantin Asofiei wrote:

> About 2565a rev 11035:
>
> 1. ClientExports.setMousePointer - the javadoc is missing the parameter name for ptr.
> 2. CommonWidget.getMousePointer - you have an extra line after the method
> 3. EventList.merge - is missing javadoc for the return type.
> 4. FillinConfig.mousePointer - missing javadoc
> 5. in MousePointer.wrapper, why are you lowercasing the file name in ImageHelper.class.getResourceAsStream(fileName.toLowerCase());? Jar resources are case-sensitive...
> 6. ClientExports.setWaitState and getWaitState - please use Java Boolean and String types for the return - assume null is 4GL unknown, and let the caller (i.e. LogicalTerminal) construct the 4GL-compatible instance. Doing so, it avoids overhead; for example, logical P2J type serializes two boolean values instead of a single value.
> 7. SwingKeyboardReader.read() doesn't have to block if wait-state is on (see previous note 206/207).
> 8. ScrollPaneGuiImpl.mouseActions - why have you removed the MouseEvent.MOUSE_MOVED?
> 9. why make the MouseOps enum public? It has no reason to be exposed outside the gui.driver.web package
> 10. GuiWebSocket
>   - the location of the BinaryMessage inner class is incorrect
>   - registerHoverableWidget - javadoc doesn't contain all parameters, the parameters need to be column-aligned, one on each line
>   - registerHoverableWidget - when you are computing message length like 1 + 4 + 4 + 4 + 4*2 + 2 + 1, please add comments to explain what every constant in the expression means (i.e. widget id, window id, etc). It's easier to follow if at some point the code needs to be changed.
>   - registerHoverableWidget - you have NativeRectangle.width() and height() APIs - use this instead of i.e. bound.right() - bound.left() + 1 to compute the width/height.

1-6, 9, 10 - fixed
Committed to the task branch 2565a revno 11039.

About note 8: With the MouseEvent.MOUSE_MOVED action p2j.mouse.processMouseMove does not convert "mousemove" into "mouseenter" when mouse hovers over ScrollPanel. Instead of changing the p2j.mouse.processMouseMove logic (which is already tricky) I just removed this option. However after you note I've noticed that it may not be a right solution. I will work on it.

#214 - 02/19/2016 12:37 PM - Igor Skornyakov

MouseEvent.MOUSE_MOVED is restored in ScrollPaneGuiImpl.mouseActions.

Committed to the task branch 2565a revno 11040.

As far as I understand there are 3 remaining issues

1. Rework cursor assignment via css (Sergey - note 192).
2. Update SET-WAIT-STATE support (Greg - part 2, issue 9, Constantin - issue 7)
3. Try to load cursor at the server side (Greg - note 186).

There is still open question regarding GuiWebDriver.registerMouseWidgets() (note 202). Depending on the decision there is nothing to do or roll back to a previous version.

#215 - 02/22/2016 08:27 AM - Igor Skornyakov

Re-worked custom cursor load from a file. Added server-side load.

Committed to the task branch 2565a revno 11041

#216 - 02/22/2016 09:04 AM - Greg Shah

> Anyway - you will need to run testcases(at least all known at this time for the area the FileSytemDaemon is involved in) to find out whether this version of FileSytemDaemon works fine or may be has any regressions.

Eugenie: Please make a list of the important testcases to run for testing of this area.

#217 - 02/22/2016 09:10 AM - Greg Shah

Igor Skornyakov wrote:

> Greg Shah wrote:
>
>> Code Review Task Branch 2565a Revision 11031
>>
>> This is part 2 of the code review.
>> 42. Your expansion of the use of Streams in GuiWebDriver.registerMouseWidgets() has made it very hard for me to determine the validity of your logic changes. How thoroughly have you tested that code?
>
> Actually this refactoring is supposed to be (and I hope it is) 100% logically equivalent to the initial one. I did it just to understand what this code is doing. Of course it is a matter of my personal perception. I have tested it with my test program (mouse.p) which is quite complicated. Of course I can revert to the original version w/o any problems - just by replacing two methods with thire old versions. It is also possible to keep both versions for a while with invoking both of them and automatically comparing the results - I've used this approach before to check the correctness of refactoring a complicated code.
> However I feel that something is wrong with the very idea of this method - it seems to be very difficult to maintain in the future. Of course I can be wrong.

I agree that including both approaches permanently in the code is never something that we would want.

Pretty soon, you will be ready for regression testing. Since you have changed code that affects ChUI as well as GUI, you will have to run the normal MAJIC regression testing. But it is also important that you run all of the standalone GUI testcases. You will need to run them in both the trunk and in your branch and compare the results. I expect that while doing this, you will look carefully for any unexpected/unwanted differences in mouse processing. If you are careful with your testing AND you find no regressions, then we can accept the current changes in GuiWebDriver.registerMouseWidgets().

#218 - 02/22/2016 09:44 AM - Igor Skornyakov

Greg Shah wrote:

> Pretty soon, you will be ready for regression testing. Since you have changed code that affects ChUI as well as GUI, you will have to run the normal MAJIC regression testing. But it is also important that you run all of the standalone GUI testcases. You will need to run them in both the trunk and in your branch and compare the results. I expect that while doing this, you will look carefully for any unexpected/unwanted differences in mouse processing. If you are careful with your testing AND you find no regressions, then we can accept the current changes in GuiWebDriver.registerMouseWidgets().

Greg, this code is in the GuiWebDriver so it affects Web GUI only. I do not think that it is feasible to run all GUI tests manually and check the result. I will revert my changes.

#219 - 02/22/2016 09:52 AM - Eugenie Lyzenko

Greg Shah wrote:

> Eugenie: Please make a list of the important testcases to run for testing of this area.

1. Tests gui_btn_test5*.p. You will need all images to be in PROPATH directory - modify searchpath-fixed entry in directory.xml. The img*.png and promt-u.bmp files can be found at ../uast/button/images in testcases bzr.

2. Tests image2.p-image7.p for icon and image loading cases.

Note: may be it will require to change directory.xml file to load some tests or place images into predefined location of the file system.

#220 - 02/22/2016 09:54 AM - Greg Shah

> Greg, this code is in the GuiWebDriver so it affects Web GUI only. I do not think that it is feasible to run all GUI tests manually and check the result. I will revert my changes.

Your changes are extensive and even without these specific changes, you will have to run all of the standalone GUI testcases. We have no other way (at the present) to ensure that your code doesn't regress things.

#221 - 02/22/2016 10:01 AM - Igor Skornyakov

Greg Shah wrote:

> Your changes are extensive and even without these specific changes, you will have to run all of the standalone GUI testcases. We have no other way (at the present) to ensure that your code doesn't regress things.

Basically my changes mostly affect mouse cursor only. Anyway - how can I figure which tests should be checked and what is the best practice for doing this?
Thank you,
Igor.

#222 - 02/22/2016 10:16 AM - Greg Shah

This is the standard test list:

./ask-gui.p
./hello.p
./demo/calc.p
./demo/calc-static.p
./demo/calc-static-chars.p
./demo/calc-static-pixels.p
./toggle_box/gui/tbx_present.p
./combo_box/combo_box9_1.p
./rectangle/rect_test2.p
./rectangle/rect_test6.p
./rectangle/rect_test7_1.p
./image/image0.p
./image/image10_1.p
./button/gui_btn_test3.p
./button/gui_btn_test4.p
./button/gui_btn_test5.p
./demo/movie-ratings-dynamic.p
./demo/movie-ratings-static.p
./demo/simple_windows.p
./demo/demo_widgets.p
./window_sizing/create_empty_window.p
./window_sizing/default_empty_window.p
./window_sizing/test_runner.p
./simple_alert_box.p
./simpler_alert_box.p
./message-update6.p
./minimal_view_dynamic_widget.p
./menu/simple_sm.p
./menu/popup_ext.p
./window_parenting/waitfor_2wnd.p
./frame-z-order/zw1.p
./browse/gui/browse-gui-stat1.p

#223 - 02/22/2016 10:17 AM - Igor Skornyakov

Greg Shah wrote:

> This is the standard test list:

Thank you Greg. This looks not that much as I anticipated.

#224 - 02/22/2016 10:39 AM - Igor Skornyakov

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > Eugenie: Please make a list of the important testcases to run for testing of this area.
>
> 1. Tests gui_btn_test5*.p. You will need all images to be in PROPATH directory - modify searchpath-fixed entry in directory.xml. The img*.png and promt-u.bmp files can be found at ../uast/button/images in testcases bzr.
>
> 2. Tests image2.p-image7.p for icon and image loading cases.
>
> Note: may be it will require to change directory.xml file to load some tests or place images into predefined location of the file system.

Thank you Eugenie. Please note however that FileSystemDaemon is a collection of non-UI utility classes with well-defined semantics. May be it makes sense to have an single automated test rather than validate the correctness looking at the GUI appearance?

#225 - 02/22/2016 12:06 PM - Eugenie Lyzenko

> Thank you Eugenie. Please note however that FileSystemDaemon is a collection of non-UI utility classes with well-defined semantics. May be it makes sense to have an single automated test rather than validate the correctness looking at the GUI appearance?

May be. The tests was initially designed to verify button/image widgets working. However the automation will conflict the requirement to make directory.xml related PROPATH changes to cover all test/config conditions. I think it is enough to make all tests(1-2 hours of extra time) once and if everything is OK - we can freeze the image loading code and further testing will not required.

#226 - 02/22/2016 01:55 PM - Greg Shah

Yes, we would want to have automated tests for this. My only concern is the time it would take to create these. I don't think we have the time right now.

Eugenie: don't we have non-gui testcases for FSD in testcases/uast/ already?

#227 - 02/22/2016 02:07 PM - Eugenie Lyzenko

Greg Shah wrote:

> Eugenie: don't we have non-gui testcases for FSD in testcases/uast/ already?

Honestly I do not know for sure, may be: propath-assignment.p test.

#228 - 02/22/2016 05:17 PM - Igor Skornyakov

SET-WAIT-STATE re-worked

Committed to the task branch 2565a revno 11042.

The only remaining issue is optimized cursor assignment (via css - Sergey, see note 192). It should affect JS code only.

#229 - 02/23/2016 04:44 AM - Igor Skornyakov

According to the Firebug profiler setting a custom cursor (from the file) take 7+ times less time than e.g. foundMouseSource method and it is only 20% more expensive than setting pre-defined cursor. I suggest to postpone optimization until we'll see that it is really needed,

In a meantime I'm running manual GUI tests.

#230 - 02/23/2016 11:45 AM - Greg Shah

> According to the Firebug profiler setting a custom cursor (from the file) take 7+ times less time than e.g. foundMouseSource method and it is only 20% more expensive than setting pre-defined cursor. I suggest to postpone optimization until we'll see that it is really needed,

I'm OK with deferring this. We need to get this code through testing and into the trunk. I have added some details to #2672 to put this on the list of potential optimizations.

#231 - 02/23/2016 01:16 PM - Igor Skornyakov

With SwingGUI I've noticed only one difference while running tests mentioned in the note 222:
/demo/demo_widgets.p - difference in the EDITOR (doesn'r react on the wheel scroll and selection via mouse doesn't work)

Investigating.

#232 - 02/23/2016 01:17 PM - Constantin Asofiei

Igor Skornyakov wrote:

> With SwingGUI I've noticed only one difference while running tests mentioned in the note 222:
> /demo/demo_widgets.p - difference in the EDITOR (doesn'r react on the wheel scroll and selection via mouse doesn't work)
>
> Investigating.

Does this work in trunk?

#233 - 02/23/2016 01:18 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor Skornyakov wrote:
>
>> With SwingGUI I've noticed only one difference while running tests mentioned in the note 222:
>> /demo/demo_widgets.p - difference in the EDITOR (doesn'r react on the wheel scroll and selection via mouse doesn't work)
>>
>> Investigating.
>
> Does this work in trunk?

Yes, it works in trunk but not in 2565a

#234 - 02/23/2016 04:58 PM - Igor Skornyakov

I'm a little confused. The method mouseWheelMoved exist both in MouseWheelListener (invoked from the MouseHandler.applyMouseEvent) and MouseAdapter (invoked from the MouseHandler.processWidgetActions and SwingMouseHandler.processAction). The same is about all other mouse actions. What is the difference?
Thank you.

#235 - 02/23/2016 07:20 PM - Greg Shah

Code Review Task Branch 2565a Revision 11042

1. As mentioned previously, the UI system is designed as a series of abstraction layers. At the highest level is the converted 4GL code, which by nature will correspond closely to the original 4GL. Below that layer is the server-side UI. This is responsible for all of the runtime logic that can be handled without a trip to the client. It delegates any client-specific processing to the client UI. The client UI can be thought of in 3 abstract layers (each depending on the one below): universal common code, UI-modality common code and the driver implementations. These drivers actually utilize some platform-level UI technology for implementation.

We organize our layers into packages and sub-packages. The nesting of these packages is a clue about dependencies. Each layer should implement an API defined in the layer above and should define an API to be implemented by the layer below.

Layer	Description	Packages
1	Converted 4GL Code	com.some_customer.app.whatever
2	Server-Side UI	com.goldencode.p2j.ui
3	Universal Common Code	com.goldencode.p2j.ui.client (and some old stuff like ThinClient is also in com.goldencode.p2j.ui.chui and hasn't been moved out yet)
4	UI Modality Common Code	ChUI is in com.goldencode.p2j.ui.client.chui and GUI is in com.goldencode.p2j.ui.client.gui
5	Drivers	ChUI has com.goldencode.p2j.ui.client.chui.driver.{batch, console,swing,web} and GUI has com.goldencode.p2j.ui.client.gui.driver.{swing, web}
6	UI Technology	NCURSES, Swing, Javascript each have their own implementations outside of P2J

When you add code that references a child package from a parent package, that is a sure sign that your code is not acceptable. It will be a very rare case when that code will be allowed to be checked in. Usually, there is a better way of implementing things. This is explicitly defined in our coding standards and I have also mentioned it many times in this task.

Likewise, sibling packages that implement equivalent layers should not reference each other. For example, a Swing GUI driver should not have references to the web GUI driver and vice versa. If there is code that should be used in both places, it should be implemented in the common-code layer above those siblings.

The revision still has multiple problems in this regard:

- MousePointer should not import anything from the client.
- MousePtrWrapper should not have direct references to the SwingGuiDriver. Shouldn't the GuiDriver.setCursor() just be passed a MousePtrWrapper instance and then the driver implementation can be specific to that driver?
- ThinClient should not have any swing-specific imports.
- Too much of your code added to ThinClient is hard coded to specific GUI-widgets or to the GUI driver. This is completely inappropriate since the ThinClient is meant to be common code. I'm sure we already have many dependencies that I would not want. But your task branch makes this much worse. That needs to be fixed. What can we do to hide this away? At a minimum, the need for such code suggests that some logic should be in the widget hierarchy. In addition, perhaps most of the hover processing can exist in a separate class?

2. I don't understand the idea behind ClientExports/ThinClient.getWaitState(). It is only ever set from setWaitState() which is only ever called from the server. When either of the two possible ClientExports/ThinClient.setWaitState() entry points are called, the server already knows exactly what ThinClient.waitStateName will be when setWaitState() completes. It seems to me that we should implement getWaitState() completely on the server and eliminate it from ClientExports/ThinClient. Am I missing something here?

3. The basic premise of MousePointer is not correct. I have several concerns with it:

- Although I initially suggested that we use an enum value of CUSTOM as a placeholder for the missing AWT cursors, in note 121 I asked you to take a different path: "In other words, I don't think we should treat the non-AWT pre-defined cursor types as "CUSTOM". It confuses things.". The purpose of CUSTOM should only be when the cursor is NOT one of the well-defined cursors known by the 4GL. The current use of this in MousePointer just makes that class complicated and harder to understand.
- What is the point of the fileName member? It is not used and I don't see how or why it would be used in the future. If the text name of a cursor is a well-known cursor, then the client side will know how to load it. If the cursor is not well-known, then it must be a filename and it will be found

on the server side in the application jar or on the client side from the propath + filesystem. A fileName member that is always null is just confusing. Please remove it.
- I don't see the point of having the CursorType member or references. It is largely duplicative and I see no value on the server side.

Please just turn MousePointer into a simple helper class used to detect:

- If a cursor string name is one of the well-known cursors (which would then be loaded on the client-side using the String name).
- If a cursor string name is ignored or not supported. These can be simple boolean flags and don't need to be derived from other values.

4. I think there is too much of the server-side logic in GenericWidget.loadMousePointer(). That code should handle the widget-specific error processing at the beginning and end, but the middle code should be handled in the LogicalTerminal. The advantage of that is there will be one place to understand the core server-side cursor loading logic. Today it is split into 3 places which makes it harder to understand.

I'm talking about this code:

```
      // Check is the cursor name is a pre-defined one
      String ptrName = StringHelper.safeTrimTrailing(pointerName);
      MousePointer ptr = MousePointer.fromString(ptrName);
      if (ptr != null)
      {
         if (ptr.isNotSupported())
         {
            // the pre-defined cursor which doesn't have
            // AWT/JS counterprats
            // TODO: load cursor data at the server side when we'll have
            // corresponding images
            UnimplementedFeature.missing(String.format
                  ("%s mouse pointer is not yet supported", pointerName)
            );
            return new logical();
         }

         // Valid pre-defined cursor
         h.setMousePointerName(ptrName);
         if (!ptr.isIgnored())
         {
            // set pre-defined cursor
            return new logical(LogicalTerminal.setMousePointer(getId(), ptr));
         }
         return new logical(true);
      }

      // load custom cursor
      if (!LogicalTerminal.loadMousePointer(getId(), ptrName, ptrName))
```

If you move that to a single API of LogicalTerminal.loadMousePointer(int, String), the logic will be centralized.

5. This logic in LogicalTerminal.setMousePointer() is dead:

```
      if ((ptr != null) && ptr.isFileBased())
      {
         return loadMousePointer(id, ptr.progressName, ptr.fileName);
      }
```

As far as I can tell, ptr.fileName will always be null and isFileBased() will always return false so that code is never called.

Without this dead code (and with the changes noted in item 4 above), there is no need for LogicalTerminal.setMousePointer().

6. I don't see the point to the 3rd parameter in LogicalTerminal.loadMousePointer(int, String, String). It is confusing and serves no obvious purpose. Remove this 3rd parameter and make this the common server-side logic location.

7. The LogicalTerminal.setWaitState() worker code is duplicative with the core logic of LogicalTerminal.loadMousePointer(). For example, it duplicates the unsupported and ignored processing. Also note that as currently written, it doesn't support the application jar loading using getImageStreamFromApplication(). But it should support that. Note that this means that a byte[] version of the ClientExport/ThinClient API will be needed. It seems to me that the same basic cursor loading logic applies to both. I'd like to see a more common implementation if that is reasonable. The ClientExport/ThinClient API for setWaitState() would mirror that of loadMousePointer().

8. ThinClient.loadMousePointer(String, String) has this code:

```
if (!file.exists() && !file.isFile() && !file.canRead())
```

This logic seems wrong.

9. Please rename hasCustomMousePointer() to supportsCustomMousePointers(). I think this reads closer to the intent of the method.

10. The javadoc you added for the FillInConfig.mousePointer member has custome instead of custom.

11. ImageExt, MousePointer have missing blank lines between its data members.

12. The ordering of ImageExt data members doesn't match our standard.

13. In ImageExt on line 80, the ternary operator is formatted like NONE: but should be NONE :.

14. In AbstractGuiDriver, GuiWebDriver and GuiWebEmulatedWindow please do not move to wildcards in the imports that have special comments. As noted previously, those cases are some of the few where we intentionally use explicit imports. In the case of GuiWebDriver, the move of java.util.logging to wildcards conflicts with the explicit comment that there are conflicts with jetty logging.

15. In AbstractGuiDriver and GuiWebDriver please do not remove blank lines that were put there to make the code easier to read. Put them back.

16. In many cases, you are scrambling the order of existing imports. As far as I can tell, this is unnecessary. In some cases it is actually pretty bad because it moves imports away from comments that were put there deliberately. See GuiWebDriver and the jetty logging comment. Is this your IDE messing up the code? If so, perhaps you should stop using that IDE because your IDE is costing Golden Code time (your time AND mine).

17. In AbstractGuiDriver, your addition of makeImage() shows 2 coding standards problems. The throws clause should be on a separate line and there should be a blank line in between the method definition and the javadoc of the next method.

```
   public BufferedImage makeImage(byte[] data) throws IOException
   {
      return ImageIO.read(ImageIO.createImageInputStream(data));
   }
   /**
```

18. The MouseHoverAction.widgetId data member is missing javadoc.

19. In p2j.screen.js the getActiveWindowMouseHandler() function should have a blank line after it.

20. The indentation of Utils.getExt() and Utils.mimeType() is incorrect. The convention is 3 spaces per indention level and no hard tabs.

#236 - 02/23/2016 09:38 PM - Eugenie Lyzenko

> I'm a little confused. The method mouseWheelMoved exist both in MouseWheelListener (invoked from the MouseHandler.applyMouseEvent) and MouseAdapter (invoked from the MouseHandler.processWidgetActions and SwingMouseHandler.processAction). The same is about all other mouse actions. What is the difference?

We have two approaches of handling mouse events:
1. Source oriented(locating the source and executing event for given widget source)
2. Action oriented(getting the action and executing action event for any widget that is registered for this action)

The order of mouse events handling is:
- source oriented if the mouse widget source can be found MouseHandler.applyMouseEvent()
- action oriented if the widget can be identified for given action MouseHandler.processWidgetActions()

So you have the opportunity to choose one of them or both or even none.

Because by default the P2J widget has no reaction to mouse event(we intercept the mouse control to be able to implement exactly what we need in P2J and even insert new artificial mouse event into queue).

#237 - 02/24/2016 03:28 AM - Igor Skornyakov

Eugenie Lyzenko wrote:

> > I'm a little confused. The method mouseWheelMoved exist both in MouseWheelListener (invoked from the MouseHandler.applyMouseEvent) and MouseAdapter (invoked from the MouseHandler.processWidgetActions and SwingMouseHandler.processAction). The same is about all other mouse actions. What is the difference?
>
> We have two approaches of handling mouse events:
> 1. Source oriented(locating the source and executing event for given widget source)
> 2. Action oriented(getting the action and executing action event for any widget that is registered for this action)
>
> The order of mouse events handling is:
> - source oriented if the mouse widget source can be found MouseHandler.applyMouseEvent()
> - action oriented if the widget can be identified for given action MouseHandler.processWidgetActions()
>
> So you have the opportunity to choose one of them or both or even none.
>
> Because by default the P2J widget has no reaction to mouse event(we intercept the mouse control to be able to implement exactly what we need in P2J and even insert new artificial mouse event into queue).

Thank you Eugenie. Unfortunately the approaches you described seem to be not completely interchangeable. For example if I invoke EditorGuiImpl.mouseWeelMoved() from the action it doesn't have any effect. May be I do something wrong,

#238 - 02/24/2016 03:39 AM - Eugenie Lyzenko

Igor Skornyakov wrote:

> Thank you Eugenie. Unfortunately the approaches you described seem to be not completely interchangeable. For example if I invoke EditorGuiImpl.mouseWeelMoved() from the action it doesn't have any effect. May be I do something wrong,

For this approach to work you need the widget to register itself as mouse wheel move event client/receiver. It does not work by default.

#239 - 02/24/2016 03:49 AM - Constantin Asofiei

Igor Skornyakov wrote:

> With SwingGUI I've noticed only one difference while running tests mentioned in the note 222:
> /demo/demo_widgets.p - difference in the EDITOR (doesn'r react on the wheel scroll and selection via mouse doesn't work)
>
> Investigating.

The problem is that the mouse source is resolved incorrectly when a mouse action takes place within the editor. Is resolved to a ScrollPaneGuiImpl widget instead of the EditorGuiImpl instance. But if I change the EditorGuiImpl.findMouseSource to this:

```
   public Widget<GuiOutputManager> findMouseSource(NativePoint p)
   {
      Widget<GuiOutputManager> src = super.findMouseSource(p);

      return (src == editor || src == contentPane || this == src.parent(EditorGuiImpl.class)) ? this : src;
   }
```

The mouse actions will be processed (click, selection, wheel) but the cursor will no longer be set to TEXT mode, when hovering the EDITOR widget.

#240 - 02/24/2016 03:58 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor Skornyakov wrote:
>
> > With SwingGUI I've noticed only one difference while running tests mentioned in the note 222:
> > /demo/demo_widgets.p - difference in the EDITOR (doesn'r react on the wheel scroll and selection via mouse doesn't work)
> >
> > Investigating.
>
> The problem is that the mouse source is resolved incorrectly when a mouse action takes place within the editor. Is resolved to a ScrollPaneGuiImpl widget instead of the EditorGuiImpl instance. But if I change the EditorGuiImpl.findMouseSource to this:
> [...]
>
> The mouse actions will be processed (click, selection, wheel) but the cursor will no longer be set to TEXT mode, when hovering the EDITOR widget.

Thank you Constantin. I've realized this and trying to fix the regression with retaining correct support for the cursor. At this moment I see the problem with conflict of two ways to handle mouse events. Mouse hovering is processed via actions while the processing of all/most other events is source oriented. However this is the fact in the trunk as well and the question is what have been broken in 2565a.

#241 - 02/24/2016 04:03 AM - Constantin Asofiei

The changes in this diff solve the EDITOR problem in 2565a:

```
=== modified file 'src/com/goldencode/p2j/ui/client/gui/EditorGuiImpl.java'
--- src/com/goldencode/p2j/ui/client/gui/EditorGuiImpl.java     2016-02-18 20:12:53 +0000
+++ src/com/goldencode/p2j/ui/client/gui/EditorGuiImpl.java     2016-02-24 08:57:46 +0000
@@ -1346,7 +1346,6 @@
       return (src == editor || src == contentPane) ? this : src;
    }

-
    /**
     * Get the widget's actual bounds
     *
@@ -2020,8 +2019,7 @@
    @Override
    protected MouseHoverAction createMouseHoverAction()
    {
-      return editorScroll == null ? null : new MouseHoverAction(this,
-            editorScroll.getId().asInt());
+      return editorScroll == null ? null : new MouseHoverAction(this, getId().asInt());
    }
    /**

=== modified file 'src/com/goldencode/p2j/ui/client/widget/AbstractContainer.java'
--- src/com/goldencode/p2j/ui/client/widget/AbstractContainer.java      2016-02-10 19:24:23 +0000
+++ src/com/goldencode/p2j/ui/client/widget/AbstractContainer.java      2016-02-24 08:57:17 +0000
@@ -360,9 +360,8 @@
          {
             continue;
          }
-
-         return (found != null && found.getId() != null) ? found :
-               (w.getId() == null ? null : w);
+
+         return (found == null) ? w : found;
       }

       return null;
```

So, question: why was the AbstractContainer.findMouseSource change needed? I'm referring to this:

```
-         return (found != null && found.getId() != null) ? found :
-               (w.getId() == null ? null : w);
```

#242 - 02/24/2016 04:08 AM - Igor Skornyakov

Constantin Asofiei wrote:

> The changes in this diff solve the EDITOR problem in 2565a:
> [...]

Thank you Constantin.

> So, question: why was the AbstractContainer.findMouseSource change needed? I'm referring to this:
> [...]

W/o this change the mouse source was found incorrectly for the combo-box. Please see note 159.

#243 - 02/24/2016 05:36 AM - Constantin Asofiei

The MouseHoverAction in case of a drop-down is registered with a ScrollPaneGuiImpl, as we need a widget with an ID, to be able to register it. For drop-down, neither the DropDownGuiImpl or ScrollableSelectionListGuiImpl have an ID. At this time, I think your change in AbstractContainer.findMouseSource tried to solve this issue: let only widgets which have an ID be returned, as otherwise hover actions for these widget at the driver level can not be resolved (and can not even be registered). So, why not do this: instead of changing the logic in AbstractContainer.findMouseSource, move the MouseHoverAction in case of drop-down at ScrollableSelectionListGuiImpl, as this is what AbstractContainer.findMouseSource would find, if return (found == null) ? w : found; is used (and ensure the instance has a proper ID).

Beside this, for widgets which don't have a custom cursor, we don't need to register a MouseHoverAction - this will make the Web driver respond more slowly. Only widgets which have explicitly called SET-MOUSE-POINTER to a cursor other than default OR widgets which show the IBEAM (i.e. FILL-IN/EDITOR), need to have a MouseHoverAction.

#244 - 02/24/2016 05:47 AM - Igor Skornyakov

Constantin Asofiei wrote:

> The MouseHoverAction in case of a drop-down is registered with a ScrollPaneGuiImpl, as we need a widget with an ID, to be able to register it. For drop-down, neither the DropDownGuiImpl or ScrollableSelectionListGuiImpl have an ID. At this time, I think your change in AbstractContainer.findMouseSource tried to solve this issue: let only widgets which have an ID be returned, as otherwise hover actions for these widget at the driver level can not be resolved (and can not even be registered). So, why not do this: instead of changing the logic in AbstractContainer.findMouseSource, move the MouseHoverAction in case of drop-down at ScrollableSelectionListGuiImpl, as this is what AbstractContainer.findMouseSource would find, if return (found == null) ? w : found; is used (and ensure the instance has a proper ID).

This what I've tried initially, but it didn't work. I will try again.

> Beside this, for widgets which don't have a custom cursor, we don't need to register a MouseHoverAction - this will make the Web driver respond more slowly. Only widgets which have explicitly called SET-MOUSE-POINTER to a cursor other than default OR widgets which show the IBEAM (i.e. FILL-IN/EDITOR), need to have a MouseHoverAction.

Actually this is what I do now. For EDITOR, WINDOW and SELECTION-LIST the constructor with second argument is required because the detected mouse source was not the widget itself but its sub-widget.

#245 - 02/24/2016 07:09 AM - Constantin Asofiei

Igor Skornyakov wrote:

> > Beside this, for widgets which don't have a custom cursor, we don't need to register a MouseHoverAction - this will make the Web driver respond more slowly. Only widgets which have explicitly called SET-MOUSE-POINTER to a cursor other than default OR widgets which show the IBEAM (i.e. FILL-IN/EDITOR), need to have a MouseHoverAction.
>
> Actually this is what I do now. For EDITOR, WINDOW and SELECTION-LIST the constructor with second argument is required because the detected mouse source was not the widget itself but its sub-widget.

I understand what you say here. But in the SwingMouseHandler$WorkArea.actions map there are frames/buttons/radio-set/etc (in the demo_widgets.p) which don't have a customized cursor (IBEAM or set via SET-MOUSE-POINTER).

#246 - 02/24/2016 07:11 AM - Igor Skornyakov

Greg, I've reworked MousePonter/MousePtrWrapper (committed to the task branch 25652 revno 11045).
I've also removed the reference to the ComboBoxGuiImpl from the ThinClient. The later still has ComboBox-specific code. At this moment I do not see a good way to get rid of it.

The idea of the filename field in the MousePointer is to have some place where the name of the file with cursor data will be located when we'll have them for the pre-defined cursors w/o AWT counterparts. This is reflected in the comment at the beginning of the class. I decided that it is better to have the information about such cursors in the place where they are defined. I've considered the option of keeping it in an external properties file but it seems to add unneeded complexity.

I'm working now on the regression with the EDITOR (note 232) and will address the remaining issues after that

#247 - 02/24/2016 07:13 AM - Igor Skornyakov

Constantin Asofiei wrote:

> I understand what you say here. But in the SwingMouseHandler$WorkArea.actions map there are frames/buttons/radio-set/etc (in the demo_widgets.p) which don't have a customized cursor (IBEAM or set via SET-MOUSE-POINTER).

Sorry Constantin, I do not understand what you mean. All the widgets you've mentioned **do** support custom mouse cursor.

#248 - 02/24/2016 07:28 AM - Igor Skornyakov

My guess is that the root of the problem is that the findMouseSource method doesn't take into account the type of the event. As result it stops too early or too late in the z-order. A similar issue is with looking for the trigger source as it is difficult to figure if the widget can have triggers at all.

#249 - 02/24/2016 07:30 AM - Greg Shah

> The idea of the filename field in the MousePointer is to have some place where the name of the file with cursor data will be located when we'll have them for the pre-defined cursors w/o AWT counterparts. This is reflected in the comment at the beginning of the class. I decided that it is better to have the information about such cursors in the place where they are defined. I've considered the option of keeping it in an external properties file but it seems to add unneeded complexity.

This will never be used. As I have requested previously, only the client should have any knowledge of how to load a pre-defined cursor. How those cursors are mapped (AWT or file in our p2j.jar) is not related to the MousePointer class. Your comment mentions loading these from the server, which we don't want to do. The only thing **ever** loaded from the server is a converted application resource (a filename that matches an application resource that has been moved to the jar file). All pre-defined cursors will be loaded from the client and the client will know if the cursor is taken from AWT or is loaded from the **client's** p2j.jar.

#250 - 02/24/2016 07:55 AM - Igor Skornyakov

Greg Shah wrote:

> > The idea of the filename field in the MousePointer is to have some place where the name of the file with cursor data will be located when we'll have them for the pre-defined cursors w/o AWT counterparts. This is reflected in the comment at the beginning of the class. I decided that it is better to have the information about such cursors in the place where they are defined. I've considered the option of keeping it in an external properties file but it seems to add unneeded complexity.
>
> This will never be used. As I have requested previously, only the client should have any knowledge of how to load a pre-defined cursor. How those cursors are mapped (AWT or file in our p2j.jar) is not related to the MousePointer class. Your comment mentions loading these from the server, which we don't want to do. The only thing **ever** loaded from the server is a converted application resource (a filename that matches an application resource that has been moved to the jar file). All pre-defined cursors will be loaded from the client and the client will know if the cursor is taken from AWT or is loaded from the **client's** p2j.jar.

Thank you Greg. I will change the code accordingly. Sorry for my stupidity - I was busy with resolving many functionality issues and paid not enough attention to the code hygiene.

#251 - 02/24/2016 09:36 AM - Igor Skornyakov

Fixed the issue with MOUSE_WHEEL/MOUSE_DRAGGED in the EDITOR (Swing). Many thanks to Constantin for a clue.

Committed to the task branch 2565a revno 11046.

#252 - 02/24/2016 12:47 PM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11042

All issues except 14-16 have been addressed.

Committed to the task branch 2565a revno 11047.

#253 - 02/24/2016 02:18 PM - Greg Shah

Code Review Task Branch 2565a Revision 11047

This is substantially better.

1. ThinClient/ClientExports.loadMousePointer() still has a ptrName javadoc parameter.

2. ThinClient/ClientExports.setWaitState() still has a stateName javadoc parameter.

3. The initial value for LogicalTerminal.waitState is null. Will that cause a correct result if the converted code calls getWaitState() before ever using setWaitState()?

4. The date in your history entry for ImageHelper is 2015-224 but should be 20160224.

5. There should be a blank line following SwingGuiDriver.setCursor().

6. I don't see why the GuiDriver.createMousePointer(CursorType) exists. There is just a single implementation (used by all drivers, which calls MousePtrWrapper.createPreDefinedInstance(CursorType). Considering it is only used in one place, can't ThinClient.createPtrWrapper() just call that directly and then you eliminate the extra API in GuiDriver?

#254 - 02/24/2016 02:34 PM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11047

> This is substantially better.
>
> 1. ThinClient/ClientExports.loadMousePointer() still has a ptrName javadoc parameter.
>
> 2. ThinClient/ClientExports.setWaitState() still has a stateName javadoc parameter.
>
> 3. The initial value for LogicalTerminal.waitState is null. Will that cause a correct result if the converted code calls getWaitState() before ever using setWaitState()?
>
> 4. The date in your history entry for ImageHelper is 2015-224 but should be 20160224.
>
> 5. There should be a blank line following SwingGuiDriver.setCursor().
>
> 6. I don't see why the GuiDriver.createMousePointer(CursorType) exists. There is just a single implementation (used by all drivers, which calls MousePtrWrapper.createPreDefinedInstance(CursorType). Considering it is only used in one place, can't ThinClient.createPtrWrapper() just call that directly and then you eliminate the extra API in GuiDriver?

Fixed.
Committed to the task branch 2565a revno 11050.

#255 - 02/24/2016 02:43 PM - Greg Shah

Code Review Task Branch 2565a Revision 11050

ClientExports.loadMousePointer() still has a ptrName javadoc parameter.

What about items 14-16 from note 235?

#256 - 02/24/2016 02:46 PM - Igor Skornyakov

I'm comparing 2565a with the trunk. This is the only way to see what I messed up.

#257 - 02/24/2016 03:32 PM - Igor Skornyakov

Fixed items 14-16 from note 235?

Committed to the task branch 2565a revno 11051,

#258 - 02/24/2016 03:52 PM - Greg Shah

OK, it looks fine, except for the ClientExports.loadMousePointer() which still has a ptrName javadoc parameter.

Please make this last fix and then put this code into regression testing on devsrv01.

In addition, please re-run the GUI regression testing to ensure that the latest changes have not caused problems.

#259 - 02/24/2016 04:08 PM - Igor Skornyakov

Greg Shah wrote:

> OK, it looks fine, except for the ClientExports.loadMousePointer() which still has a ptrName javadoc parameter.
>
> Please make this last fix and then put this code into regression testing on devsrv01.

Sorry. Fixed and committed to 2565a revno 11052
Regresion started

> In addition, please re-run the GUI regression testing to ensure that the latest changes have not caused problems.

I will do the manual testing tomorrow as I'm not sure that today I'm able to concentrate enough.

#260 - 02/25/2016 06:19 AM - Constantin Asofiei

Please check the LOAD-MOUSE-POINTER for combo-box/button - is not working anymore, in rev 11052.

#261 - 02/25/2016 06:20 AM - Igor Skornyakov

The regression found: the additional window opened in the ./demo/calc-static-chars.p is not movable/resizable in 2565a (WebGUI)

#262 - 02/25/2016 06:20 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Please check the LOAD-MOUSE-POINTER for combo-box/button - is not working anymore, in rev 11052.

Thank you Constantin.

#263 - 02/25/2016 06:40 AM - Igor Skornyakov

- File mouse.p added

Constantin Asofiei wrote:

> Please check the LOAD-MOUSE-POINTER for combo-box/button - is not working anymore, in rev 11052.

Sorry Constantin. In my test (attached) it works fine both in Swing and Web.
Please note that LOAD-MOUSE-POINTER for the COMBO-BOX has a complicated (and counter-intuitive) semantics - see note 27 **N7**.
Can you please provide your test?
Thank you,
Igor.

#264 - 02/25/2016 06:54 AM - Igor Skornyakov

Igor Skornyakov wrote:

> The regression found: the additional window opened in the ./demo/calc-static-chars.p is not movable/resizable in 2565a (WebGUI)

This seems to be a manifestation of #2996. Continue analysis.

#265 - 02/25/2016 04:36 PM - Igor Skornyakov

The results of the regression testing:

1. All ctrl-C tests passed.

2. The following main tests failed in 3 runs:

```
38     gso_146     GSO 146 - Can not add training in Training Maintenance.     FAILED     NONE     BACKOUT
 Driver 1     192.763
```

```
failure in step 23: 'timeout before the specific screen buffer became available (Mismatched data at line 0, co
lumn 0. Expected '┌' (0x250C at relative Y 0, relative X 0) and found ' ' (0x0000 at relative Y 0, relative X
0), template: screens/gso/gso_146/training_maintenance_add_step5.txt.)'
```

```
101     gso242_session2     GSO 242 (Session 2) - Adding Item Number and Text during SO Copy will lock both pr
ocesses.     FAILED     CONCURRENT     BACKOUT     Driver 1     223.711
```

```
failure in step 3: 'timeout before the specific screen buffer became available (Mismatched data at line 0, col
umn 10. Expected ' ' (0x0020 at relative Y 0, relative X 10) and found 'M' (0x004D at relative Y 0, relative X
 10), template: screens/common/item_master_1_of_2.txt.)'
```

```
256     gso_491     GSO 491 - Training module focus problems.     FAILED     NONE     BACKOUT     Driver 3
 189.812
```

```
failure in step 15: 'timeout before the specific screen buffer became available (Mismatched data at line 0, co
lumn 0. Expected '┌' (0x250C at relative Y 0, relative X 0) and found ' ' (0x0000 at relative Y 0, relative X
0), template: screens/gso/gso_491/training_module_focus_issue_step1.txt.)'
```

```
260     logout_from_main_menu     Logout from the MAJIC main menu.     FAILED     POST_CONDITION     BACKOUT
   Driver 1     180.052
```

```
failure in step 1: 'timeout before the specific screen buffer became available (Mismatched data at line 0, col
umn 1. Expected 'M' (0x004D at relative Y 0, relative X 1) and found ' ' (0x0000 at relative Y 0, relative X 1
), template: screens/navigation/syman_menu.txt.)'
```

```
263     logout_from_main_menu     Logout from the MAJIC main menu.     FAILED     POST_CONDITION     BACKOUT
   Driver 3     180.055

failure in step 1: 'timeout before the specific screen buffer became available (Mismatched data at line 0, col
umn 1. Expected 'M' (0x004D at relative Y 0, relative X 1) and found ' ' (0x0000 at relative Y 0, relative X 1
), template: screens/navigation/syman_menu.txt.)'
----
396     tc_job_002     TIMCO TC-JOB-002 testcase.     FAILED     NONE     BACKOUT     Driver 2     40.954

failure in step 40: 'Unexpected EOF in actual at page # 1.'

524     tc_dan_004     TIMCO TC-DAN-004 testcase.     FAILED     SEQUENTIAL     BACKOUT     Driver 4     182.8
70

failure in step 16: 'timeout before the specific screen buffer became available (Mismatched data at line 0, co
lumn 10. Expected ' ' (0x0020 at relative Y 0, relative X 10) and found 'M' (0x004D at relative Y 0, relative
X 10), template: screens/tc/dan/004/dan_004_step3.txt.)'

525     tc_dan_005     TIMCO TC-DAN-005 testcase.     FAILED_DEPENDENCY     SEQUENTIAL     BACKOUT
    0.000

dependency chain: tc_dan_004

528     tc_dan_010     TIMCO TC-DAN-010 testcase.     FAILED_DEPENDENCY     SEQUENTIAL     BACKOUT
    0.000

dependency chain: tc_dan_004 --> tc_dan_005

529     tc_dan_011     TIMCO TC-DAN-011 testcase.     FAILED_DEPENDENCY     SEQUENTIAL     BACKOUT
    0.000

dependency chain: tc_dan_004 --> tc_dan_005 --> tc_dan_010
```

These runs where not at the best time of the day. Restarted main-regression,

#266 - 02/26/2016 03:50 AM - Igor Skornyakov

Fixed a bug revealed by automated regression testeting
Committed to a task branch 2565a revno 11053

Regression test restarted

#267 - 02/26/2016 07:59 AM - Igor Skornyakov

Fixed a bug revealed by automated regression testeting
Committed to a task branch 2565a revno 11054

Regression test restarted

#268 - 02/26/2016 09:12 AM - Greg Shah

Code Review Task Branch 2565a Revision 11054

1. The GUI-specific additions to ChuiWidgetFactory are inappropriate. This is the same problem as mentioned before where you are adding dependencies that should not be there.

- remove the GUI package imports
- clean up the javadoc (it should not reference GUI classes and it should not reference GUI in the text at all)

2. ChuiWidgetFactory is missing a history entry.

3. I don't want to leave commented code in WidgetFactoryAdapter for createScrollPane(). If it doesn't need to be there, remove it.

#269 - 02/26/2016 09:14 AM - Greg Shah

What issues remain to be fixed from manual testing or automated regression testing?

#270 - 02/26/2016 09:16 AM - Igor Skornyakov

Greg Shah wrote:

> What issues remain to be fixed from manual testing or automated regression testing?

I have good reasons to expect that regression test will pass now. From the manual testing I'm working on the issue described in the note 261 and hope to finish soon today.

#271 - 02/26/2016 09:21 AM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11054

> 1. The GUI-specific additions to ChuiWidgetFactory are inappropriate. This is the same problem as mentioned before where you are adding dependencies that should not be there.
>
> - remove the GUI package imports
> - clean up the javadoc (it should not reference GUI classes and it should not reference GUI in the text at all)
>
> 2. ChuiWidgetFactory is missing a history entry.
>
> 3. I don't want to leave commented code in WidgetFactoryAdapter for createScrollPane(). If it doesn't need to be there, remove it.

Done. Committed to the task branch 2565a revno 11055,

#272 - 02/26/2016 09:26 AM - Igor Skornyakov

There is a number of methods in the WidgetFactoryAdapter which just throw new RuntimeException("Not implemented.") but are implemented both in ChuiWidgetFactory and GuiWidgetFactory. Do we really need them?
Thank you.

#273 - 02/26/2016 09:32 AM - Greg Shah

Code Review Task Branch 2565a Revision 11055

The javadoc changes requested in note 268 were not done.

> clean up the javadoc (it should not reference GUI classes and it should not reference GUI in the text at all)

Please search on ScrollPaneGuiImpl and GUI to see what I am talking about.

#274 - 02/26/2016 09:33 AM - Greg Shah

> There is a number of methods in the WidgetFactoryAdapter which just throw new RuntimeException("Not implemented.") but are implemented both in ChuiWidgetFactory and GuiWidgetFactory. Do we really need them?

No. Anything that is unused can be removed.

#275 - 02/26/2016 09:42 AM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11055

Sorry. Fixed now. I've also removed obsolete methods from the WidgetFactoryAdapter.

Committed to the task branch 2565a revno 11057.

#276 - 02/26/2016 09:45 AM - Greg Shah

Code Review Task Branch 2565a Revision 11057

In ChuiWidgetFactory please search on the text "GUI" and change it to "UI".

#277 - 02/26/2016 09:50 AM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11057
>
> In ChuiWidgetFactory please search on the text "GUI" and change it to "UI".

Done.
Committed to the task branch 2565a revno 11058.

#278 - 02/26/2016 11:45 AM - Igor Skornyakov

At the last run only the following regression tests failed:

```
394     tc_dc_slot_029     TIMCO TC-DC-SLOT-029 testcase.     FAILED     NONE     BACKOUT     Driver 4     800
0.410

failure in step 20: 'timeout before the specific screen buffer became available (Mismatched data at line 4, co
lumn 0. Expected ' ' (0x0000 at relative Y 4, relative X 0) and found 'L' (0x004C at relative Y 4, relative X
0), template: screens/tc/dc/slot/029/dc_slot_029_step7.txt.)'

396     tc_job_002     TIMCO TC-JOB-002 testcase.     FAILED     NONE     BACKOUT     Driver 3     57.346

failure in step 40: 'Unexpected EOF in actual at page # 1.'

398     tc_job_clock_002     TIMCO TC-JOB-CLOCK-002 testcase.     FAILED     NONE     BACKOUT     Driver 3
 204.962

failure in step 20: 'timeout before the specific screen buffer became available (Mismatched data at line 5, co
lumn 18. Expected ' ' (0x0000 at relative Y 5, relative X 18) and found '┌' (0x250C at relative Y 5, relative X 18), template: screens/tc/job/clock/002/job_clock_002_step6_2.txt.)'
```

```
496     tc_item_master_104     TIMCO TC-ITEM-MASTER-104 testcase.     FAILED     NONE     BACKOUT     Driver 7
     879.647
```

```
failure in step 45: 'Mismatched data at absolute row 55763, relative row 25, page # 962, page size 58, last page false, column index 21. Expected '8' (0x38) and found 'N' (0x4E).'
```

Only tc_job_00 failed in all runs. I consider automated regression tests passed,

#279 - 02/26/2016 11:48 AM - Greg Shah

OK, thanks.

When you have the fix for the GUI issue, we will evaluate if the changes needed will require a new round of automated regression testing.

Also, there is about to be a merge to trunk of a set of javadoc fixes that will likely require some merging on your part. It will be important for you to merge these very carefully. This will happen before you will be scheduled to merge to trunk.

#280 - 02/29/2016 08:20 AM - Igor Skornyakov

The issue with mouse drug is fixed.
Committed to the task branch 2565a revno 11059

Only JS code was affected.

Resuming manual GUI tests.

#281 - 02/29/2016 09:39 AM - Greg Shah

Code Review Task Branch 2565a Revision 11059

1. Are you intending to assign me.currentHovered and me.currentMousePtr from inside the Window constructor? This seems wrong, especially since later in Window functions, you refer to this.currentHovered. Generally, we should not have member data like this exposed as globally accessible on the p2j.screen object.

2. In Window.setCursorStyle(), you reference this.currentMousrPtr. Notice the "Mousr" instead of "Mouse". Is this intentional? It seems wrong. Again, it is confusing because it looks like you are trying to reference currentMousePtr, but that data is on the p2j.screen object and not on the Window instance.

3. Do you really need to publicly expose all the following member data in p2j.screen?

```
/** locked mouse pointer */
   me.lockedMousePointer = null;

   /** the id of the widget which caused the mouse pointer lock*/
   me.activeDropDown = -1;
   /** most recent active window  (non-overlay)*/
   me.activeWindow = null;

   /** wait state flag */
   me.isWaitState = false;

   /** most recent mouse position */
   me.lastMousePos = null;
```

It should be private state instead, unless there is a really good reason.

4. Do MouseHandler.sendMouseExit(), MouseHandler.sendMouseEnter() and MouseHandler.sendMouseEvent() need to be exposed as methods of MouseHandler or can they be hidden and only used inside the object.

5. MouseHandler.sendMouseExit(), MouseHandler.sendMouseEnter() and MouseHandler.sendMouseEvent() are missing javadoc.

6. Please look at your web client (bzr diff -r10973) and cleanup the coding standards problems.

#282 - 02/29/2016 10:19 AM - Igor Skornyakov

Greg Shah wrote:

> Code Review Task Branch 2565a Revision 11059

Fixed. Committed to the task branch 2565a revno 11060.

BTW: the #2996 is fixed as well.

#283 - 02/29/2016 11:46 AM - Constantin Asofiei

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
> > Please check the LOAD-MOUSE-POINTER for combo-box/button - is not working anymore, in rev 11052.
>
> Sorry Constantin. In my test (attached) it works fine both in Swing and Web.

Does IBEAM still work in your tests? I was testing IBEAM for a combo-box in demo-widgets.p, and is no longer working. I don't think is working either in your mouse.p test - I get a Unable to open file for ... error message if IBEAM is used.

#284 - 02/29/2016 11:55 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Does IBEAM still work in your tests? I was testing IBEAM for a combo-box in demo-widgets.p, and is no longer working. I don't think is working either in your mouse.p test - I get a Unable to open file for ... error message if IBEAM is used.

Thank you Constantin. It was a typo in the MousePointer class. It is fixed now.

Committed to the task branch 2565a revbo 11061.

#285 - 02/29/2016 12:47 PM - Igor Skornyakov

As far as I can see all standard manual GUI tests run identically in 2565a and trunk (WebGUI) with one exception: the mouse selection in the EDITOR (./demo/demo_widgets.p) doesn't work in 2565a. Mouse wheel doesn't work in both.

Investigating.

#286 - 02/29/2016 01:43 PM - Igor Skornyakov

The issue from the note 285 is fixed.

Committed to the task branch 2565a revno 11063.

What's remained is the manual tests of FSD (note 224)

#287 - 02/29/2016 02:15 PM - Greg Shah

In what revision did your manual testing of the Swing GUI client pass all standalone GUI tests?

#288 - 02/29/2016 02:21 PM - Igor Skornyakov

Greg Shah wrote:

> In what revision did your manual testing of the Swing GUI client pass all standalone GUI tests?

It was revno 11046.

#289 - 02/29/2016 02:28 PM - Igor Skornyakov

As far as I can see FSD tests passed OK after adding the following to the directory.xml:

```
            <node class="string" name="searchpath-fixed">
              <node-attribute name="value" value=".:/home/ias/0/testcases/uast/button/images:/home/ias/0/testcases/uast/image/icons"/>
            </node>
```

#290 - 02/29/2016 02:44 PM - Greg Shah

>> In what revision did your manual testing of the Swing GUI client pass all standalone GUI tests?
>
> It was revno 11046.

There has been a substantial amount of change in GUI code since then. I think it is important to re-run those tests on 11063 to ensure there has been no regressions.

What revision was used to pass MAJIC regression testing?

#291 - 02/29/2016 02:50 PM - Igor Skornyakov

Greg Shah wrote:

> There has been a substantial amount of change in GUI code since then. I think it is important to re-run those tests on 11063 to ensure there has been no regressions.

OK, I will re-test

> What revision was used to pass MAJIC regression testing?

I do not remember exactly but after it was passed the changes where made in JS code and WebMouseHandler only. However I can restart testing right now.

#292 - 02/29/2016 03:00 PM - Greg Shah

Before you do this testing, please rebase. It should be the last one before your merge.

Although I'd prefer not to redo the testing, it seems necessary.

#293 - 02/29/2016 03:01 PM - Igor Skornyakov

Greg Shah wrote:

> Before you do this testing, please rebase. It should be the last one before your merge.
>
> Although I'd prefer not to redo the testing, it seems necessary.

OK.

#294 - 02/29/2016 03:27 PM - Igor Skornyakov

Rebased task branch 2565a from P2J trunk rev 10975. The latest revision is now 11066.

The only conflicts during rebase where in history records and duplicated fields/methods in SliderConfig and SliderWidget

#295 - 02/29/2016 03:52 PM - Igor Skornyakov

Fixed javadocs

Committed to the task branch 2565a revno 11067.

#296 - 03/01/2016 04:49 AM - Igor Skornyakov

Manual Swing GUI tests passed OK.

#297 - 03/01/2016 05:39 AM - Igor Skornyakov

Manual Web GUI tests passed OK

#298 - 03/01/2016 08:17 AM - Greg Shah

How is MAJIC regression testing going?

#299 - 03/01/2016 08:23 AM - Igor Skornyakov

Greg Shah wrote:

> How is MAJIC regression testing going?

The night run ended with a lot of connection failures at the CTRL-C part. The main part ended with 3 failed tests: tc_codes_employees_021, tc_codes_employees_024 and tc_job_002.

The next run hanged at the CTRL-C part. Now the test is running again.

#300 - 03/01/2016 09:32 AM - Igor Skornyakov

All gso_ctrlc_tests passed, all gso_ctrlc_3way_tests failed.

#301 - 03/01/2016 09:33 AM - Greg Shah

Are you confident that you have not lost any of Eugenie's javadoc changes?

#302 - 03/01/2016 09:37 AM - Igor Skornyakov

Greg Shah wrote:

> Are you confident that you have not lost any of Eugenie's javadoc changes?

I hope so. There where no any conflicts with javadocs. And I've checked the Oracle javadoc output after rebase. Apart from generated classes there where complains only regarding my code which I've fixed.

#303 - 03/01/2016 09:37 AM - Constantin Asofiei

Review for 2565a rev 11067:

1. Boolean ClientExports.setWaitState - please review all usage of this API; if the API returns null (as TC.setWaitState(MousePointer) does), then a if (lt.client.setWaitState(...)) call will throw a NPE. Can 4GL SET-WAIT-STATE return unknown? If so, you need to handle this. Otherwise, you only need TRUE and FALSE values returned by the ClientExports.setWaitState API, so set the return type to boolean.
2. ComboBoxguiImpl.simpleListMouseHoverAction is never used
3. MouseHoverAction.setCursor - the window local variable is never used
4. MouseOps has no functional change, remove the history entry and the copyright year change
5. p2j.screen.js - on lines 439 and 442, you are defining local variables, not members for the Window object.
6. p2j.screen.js - on line 2375, the setWaitState function is missing javadoc

Otherwise, I'm still a little worried about AbstractContainer.findMouseSource change: if you have not already, please test mouse scroll/selection for all widgets which support this: selection (on mouse drag) for EDITOR/FILL-IN and mouse scroll support for EDITOR, combo DROP-DOWN, SELECTION-LIST, BROWSE.

#304 - 03/01/2016 09:40 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Review for 2565a rev 11067:
>
> 1. Boolean ClientExports.setWaitState - please review all usage of this API; if the API returns null (as TC.setWaitState(MousePointer) does), then a if (lt.client.setWaitState(...)) call will throw a NPE. Can 4GL SET-WAIT-STATE return unknown? If so, you need to handle this. Otherwise, you only need TRUE and FALSE values returned by the ClientExports.setWaitState API, so set the return type to boolean.
> 2. ComboBoxguiImpl.simpleListMouseHoverAction is never used
> 3. MouseHoverAction.setCursor - the window local variable is never used
> 4. MouseOps has no functional change, remove the history entry and the copyright year change
> 5. p2j.screen.js - on lines 439 and 442, you are defining local variables, not members for the Window object.
> 6. p2j.screen.js - on line 2375, the setWaitState function is missing javadoc

> Otherwise, I'm still a little worried about AbstractContainer.findMouseSource change: if you have not already, please test mouse scroll/selection for all widgets which support this: selection (on mouse drag) for EDITOR/FILL-IN and mouse scroll support for EDITOR, combo DROP-DOWN, SELECTION-LIST, BROWSE.

OK, thank you.

#305 - 03/01/2016 09:46 AM - Eugenie Lyzenko

- File majic_regression_testing_ctrlc3way.xml.zip added

Igor Skornyakov wrote:

> All gso_ctrlc_tests passed, all gso_ctrlc_3way_tests failed.

To simplify 3-way tests if there are failures with CTRL-C part you can use the script attached. Run it in the majic_baseline directory:

```
./run.sh majic_regression_testing_ctrlc3way.xml
```

This executes 3-way tests in standalone session. Just make sure there are no client zombies before execution.

#306 - 03/01/2016 09:47 AM - Igor Skornyakov

Eugenie Lyzenko wrote:

> Igor Skornyakov wrote:
>
> > All gso_ctrlc_tests passed, all gso_ctrlc_3way_tests failed.
>
> To simplify 3-way tests if there are failures with CTRL-C part you can use the script attached. Run it in the majic_baseline directory:
> [...]
> This executes 3-way tests in standalone session. Just make sure there are no client zombies before execution.

Thank you Greg!

#307 - 03/01/2016 10:34 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Review for 2565a rev 11067:
>
> 1. Boolean ClientExports.setWaitState - please review all usage of this API; if the API returns null (as TC.setWaitState(MousePointer) does), then a if (lt.client.setWaitState(...)) call will throw a NPE. Can 4GL SET-WAIT-STATE return unknown? If so, you need to handle this. Otherwise, you only need TRUE and FALSE values returned by the ClientExports.setWaitState API, so set the return type to boolean.
> 2. ComboBoxguiImpl.simpleListMouseHoverAction is never used
> 3. MouseHoverAction.setCursor - the window local variable is never used
> 4. MouseOps has no functional change, remove the history entry and the copyright year change
> 5. p2j.screen.js - on lines 439 and 442, you are defining local variables, not members for the Window object.
> 6. p2j.screen.js - on line 2375, the setWaitState function is missing javadoc

Fixed.

Committed to the task branch 2565a revno 11068.

#308 - 03/01/2016 10:52 AM - Greg Shah

Please post the findMouseSource() testing results when you have them.

#309 - 03/01/2016 10:52 AM - Igor Skornyakov

Greg Shah wrote:

> Please post the findMouseSource() testing results when you have them.

Sure.

#310 - 03/01/2016 11:05 AM - Igor Skornyakov

Greg Shah wrote:

> Please post the findMouseSource() testing results when you have them.

Gentlemen, I don't mean to criticize, but don't you think that the logic based on positive/negative/absent widget id is too fragile? May be it is better to use marker interfaces or widget type/options mask instead?

#311 - 03/01/2016 11:14 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Otherwise, I'm still a little worried about AbstractContainer.findMouseSource change: if you have not already, please test mouse scroll/selection for all widgets which support this: selection (on mouse drag) for EDITOR/FILL-IN and mouse scroll support for EDITOR, combo DROP-DOWN, SELECTION-LIST, BROWSE.

As far as I can see mouse scroll/selection works identically both in trunk and 2565a (Swing). As far as I understand the AbstractContainer.findMouseSource change doesn't affect Web GUI.

BTW: selection on mouse drag for FILL-IN doesn't work neither in trunk nor on 2565a.

#312 - 03/01/2016 11:59 AM - Igor Skornyakov

Only tc_codes_employees_021 and tc_job_002 failed in two runs of the main part.

#313 - 03/01/2016 12:00 PM - Constantin Asofiei

Igor Skornyakov wrote:

> BTW: selection on mouse drag for FILL-IN doesn't work neither in trunk nor on 2565a.

Mouse-drag selection is not yet implemented for FILL-IN.

Rev 11068 is OK.

#314 - 03/01/2016 12:06 PM - Eugenie Lyzenko

Igor Skornyakov wrote:

> Only tc_codes_employees_021

Unfortunately this one is not expected to be failed - need to get it passed to be sure.

#315 - 03/01/2016 12:08 PM - Greg Shah

> > Only tc_codes_employees_021
>
> Unfortunately this one is not expected to be failed - need to get it passed to be sure.

What is the failure here?

#316 - 03/01/2016 12:08 PM - Igor Skornyakov

Eugenie Lyzenko wrote:

> Igor Skornyakov wrote:
>
> > Only tc_codes_employees_021
>
> Unfortunately this one is not expected to be failed - need to get it passed to be sure.

I understand. The regression test restated. Based on my experience this test fails pretty often.

#317 - 03/01/2016 12:10 PM - Greg Shah

> The regression test restated.

We don't want to do a full regression testing run to see if this will pass. Please run in single test mode or manually execute the test.

#318 - 03/01/2016 12:11 PM - Igor Skornyakov

Greg Shah wrote:

> > > Only tc_codes_employees_021
> >
> > Unfortunately this one is not expected to be failed - need to get it passed to be sure.
>
> What is the failure here?

```
46     CHECK-SCREEN-BUFFER

wait = true; millis = '180000'; failing screen =

03/01/2016                  EMPLOYEE MASTER (5 of 5)                    09:19:26
┌────────────────────────────────────────────────────────────────────────────────┐
│           Employee: 1029    ********** JR., *** E.                             │
│              Skill:                                                            │
│          Crew Code:                                                            │
│              Shift:   0     Days-Off:                                          │
└────────────────────────────────────────────────────────────────────────────────┘
 Start      Start End        End         Crew Crew       Skil     Shift      Days
 Date       Time  Date       Time  Dept  Code Eff Date   Code Sft Eff Date   Off
 __________ _____ __________ _____ ____  ____ __________ ____ ___ __________ ____

>07/02/1990 07:00 07/02/1990 15:30 7000 D40  01/01/1990 7300  11 01/01/1990 67
 11/01/1995 07:00 11/01/1995 15:30 7000 40D  01/01/1990 7300  11 01/01/1990 67
 09/25/1999 07:00 09/25/1999 15:30 1500 070  01/01/1990 1540  11 01/01/1990 67
 05/18/2002 07:00 05/18/2002 15:30 1100 050  01/01/1990 1130  11 01/01/1990 67
 10/18/2003 07:00 10/18/2003 15:30 7000 070  01/01/1990 1540  11 01/01/1990 67
 12/20/2003 07:00 12/20/2003 15:30 7000 070  01/01/1990 7500  11 01/01/1990 67
 06/12/2004 07:00 06/12/2004 15:30 7000 060  01/01/1990 7500  11 01/01/1990 67
 11/01/2004 00:00 11/01/2004 00:00      62E  01/01/1990 1130  11 01/01/1990 67
 11/02/2004 00:00 11/02/2004 00:00      62E  01/01/1990 1130  11 01/01/1990 67

(N)ext (P)rev (U)pdate (A)dd (C)opy (F)ind (D)el (O)utput (T)ext (X)Crew (H)elp
(E)xtend (1)Up (2)Down (3)Pg Back (4) Pg Frwd (5)Beg (6)End (R)eturn: F

    FAILED     180.055 timeout before the specific screen buffer became available (Mismatched data at line 3,
column 22. Expected '1' (0x0031 at relative Y 3, relative X 22) and found ' ' (0x0000 at relative Y 3, relativ
e X 22), template: screens/tc/codes/employees/021/codes_employees_021_step10.txt.)
```

#319 - 03/01/2016 12:12 PM - Greg Shah

What is the expected text?

#320 - 03/01/2016 12:14 PM - Igor Skornyakov

Greg Shah wrote:

> What is the expected text?

```
01/07/2010                  EMPLOYEE MASTER (5 of 5)                    09:51:50
┌──────────────────────────────────────────────────────────────────────────────┐
│           Employee: 1029    ******* JR., ********* E.                        │
│              Skill: 1130    IT SOFTWARE DEVEL                                │
│          Crew Code: 62E     MGR BUS SYS ANA - GAITHER                        │
│              Shift:  11     Days-Off: 67                                     │
└──────────────────────────────────────────────────────────────────────────────┘
 Start      Start End        End        Crew Crew       Skil     Shift      Days
 Date       Time  Date       Time  Dept Code Eff Date   Code Sft Eff Date   Off
 __________ _____ __________ _____ ____ ____ __________ ____ ___ __________ ____

>07/02/1990 07:00 07/02/1990 15:30 7000 D40  08/04/2007 7300  11 01/01/1990 67
 11/01/1995 07:00 11/01/1995 15:30 7000 40D  08/04/2007 7300  11 01/01/1990 67
 09/25/1999 07:00 09/25/1999 15:30 1500 070  08/04/2007 1540  11 01/01/1990 67
 05/18/2002 07:00 05/18/2002 15:30 1100 050  08/04/2007 1130  11 01/01/1990 67
 10/18/2003 07:00 10/18/2003 15:30 7000 070  08/04/2007 1540  11 01/01/1990 67
 12/20/2003 07:00 12/20/2003 15:30 7000 070  08/04/2007 7500  11 01/01/1990 67
 06/12/2004 07:00 06/12/2004 15:30 7000 060  08/04/2007 7500  11 01/01/1990 67
 04/05/2005 07:00 04/05/2005 15:30 7000 62B  08/04/2007 7500  11 01/01/1990 67
 08/04/2007 07:00 08/04/2007 15:30 1100 62E  08/04/2007 1130  11 01/01/1990 67
```

#321 - 03/01/2016 12:16 PM - Greg Shah

OK, please run it in single mode or manually to see if it will work. The problem is probably not due to your changes, but let's confirm.

#322 - 03/01/2016 12:25 PM - Igor Skornyakov

Greg Shah wrote:

> OK, please run it in single mode or manually to see if it will work. The problem is probably not due to your changes, but let's confirm.

In the single mode run the test passed.

#323 - 03/01/2016 12:26 PM - Greg Shah

Great!

Please merge task branch 2565a revision 11068 to trunk and archive the branch.

#324 - 03/01/2016 12:27 PM - Igor Skornyakov

Greg Shah wrote:

> Great!
>
> Please merge task branch 2565a revision 11068 to trunk and archive the branch.

Without running ctrlc3way?

#325 - 03/01/2016 12:34 PM - Greg Shah

Yes. I think your changes (since the last time CTRL-C testing passed) are safe enough.

#326 - 03/01/2016 12:35 PM - Igor Skornyakov

Greg Shah wrote:

> Yes. I think your changes (since the last time CTRL-C testing passed) are safe enough.

OK. actually ctrlc3wau has passed now. Starting merge.

#327 - 03/01/2016 12:44 PM - Igor Skornyakov

The task branch 2565a revison 11068 has been merged to the trunk and committed as revision 10976.

#328 - 03/01/2016 01:03 PM - Igor Skornyakov

The task branch 2565a has been archived

#329 - 03/01/2016 01:05 PM - Greg Shah

- % Done changed from 0 to 100

- Status changed from New to Closed

- Assignee set to Igor Skornyakov

#330 - 03/03/2016 03:57 AM - Igor Skornyakov

Greg.
I've added the test for the LOAD-MOUSE-CURSOR and the test plan (uast/mouse/load-mouse-pointer.p and uast/mouse/mouse-cursor-test-plan.txt)

- revno 1478. May be it makes sense to add it to the set of standard manual GUI tests?

#331 - 03/03/2016 09:06 AM - Greg Shah

It is a good idea. I'll add it.

#332 - 11/16/2016 12:12 PM - Greg Shah

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

Files

predefined-mouse-cursors.p	1.22 KB	11/12/2015	Igor Skornyakov
br.p	1010 Bytes	11/20/2015	Igor Skornyakov
slider.p	241 Bytes	12/01/2015	Igor Skornyakov
mouse.p	6.06 KB	12/01/2015	Igor Skornyakov
mouse.p	6.1 KB	12/01/2015	Igor Skornyakov
cursor2.html	183 Bytes	01/20/2016	Igor Skornyakov
cursor2.css	5.65 KB	01/20/2016	Igor Skornyakov
mouse.p	7.06 KB	02/02/2016	Igor Skornyakov
mouse.p	7.1 KB	02/25/2016	Igor Skornyakov
majic_regression_testing_ctrlc3way.xml.zip	840 Bytes	03/01/2016	Eugenie Lyzenko