User Interface - Feature #2676

implement the equivalent to configurable support for a windows theme

09/03/2015 04:20 PM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Ovidiu Maxiniuc	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Cleanup and Stabilization for GUI		
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to User Interface - Feature #2614: implement GUI runtime support for ...	**Closed**	
Related to User Interface - Feature #2252: implement GUI client support	**Closed**	**03/24/2014**
Related to User Interface - Feature #3684: implement a web theme inspired by ...	**Closed**	

History

#1 - 09/03/2015 04:34 PM - Greg Shah

See #2606 notes 34 through 53.

For this task, we are only going to add support for the "windows classic" theme. We will default to settings that match that theme. Each of these values should also be customizable via directory configuration. At the start of the user's session the "effective" theme should be calculated and these values used consistently throughout our GUI.

We will need to decide if we are going to provide a way to define named themes in the directory and then reference them in specific default/servers/groups/users.

I don't want to require the defaults to be specified in the directory. Themes we support directly should have the default values in code and then we should provide a way to override just those portions that need changing.

We should support choosing colors, fonts and sizes for the various Windows UI elements that we support.

For the aspects of a theme that require different drawing (e.g. rounded window borders in windows 7 basic), we will probably want to implement a pluggable approach. We may even want to allow a new theme (which has no preconfigured base theme in our code) to specify their plugin via the directory. One tricky part is that the client will have to have the class available. Normally we don't have the application code on the client system.

In the future, we will provide more pre-configured themes (windows 7 basic is probably the next most useful one and windows xp is also potentially useful). But in all likelihood, the additional themes are not needed right now.

#2 - 10/07/2015 08:15 AM - Greg Shah

As part of our implementation of the GUI widgets, we have already provided hard coded support for the Windows Classic theme. The default system colors are set to that theme and all the widget drawing is done in a compatible way. But right now this support is hard coded in place. We have no

ability to plug in alternate themes.

Contrary to the statement in note 1 above, we are going to implement the following:

1. The current Windows Classic theme support will be moved into a separate loadable/plugin form. It will still be the default theme. This includes the system color settings and the portions of the drawing code that are theme-specific.

2. The default system colors will be pluggable and thus loaded as part of setting the theme.

3. We will implement one other theme (to be determined based on customer discussions) in this pluggable form.

4. All SVG-based drawing elements (whether these are theme-related or not) which honor theme/system color overrides, will need to be edited after loading from the jar but before rendering as an image. In other words, our SVG drawing should honor the current color settings in the same way that the corresponding WIN32 control would honor it. Please see #2614-39 through 41 for some useful details.

#3 - 03/23/2016 04:23 PM - Greg Shah

- Parent task deleted (#2252)

- Target version changed from Milestone 12 to 23

#4 - 08/18/2016 10:44 AM - Greg Shah

See #2676-15 (actually read all notes from 6 through 15) to see a potential hidden dependency in our drawing on color overrides/themes.

#5 - 11/16/2016 01:21 PM - Greg Shah

- Target version changed from 23 to Cleanup and Stabilization for GUI

#6 - 02/23/2017 05:12 PM - Greg Shah

The immediate objective is to implement the plugin abstraction layer (separating out the current classic theme) while also adding a new Windows 8/10 theme plugin.

#7 - 02/23/2017 05:15 PM - Greg Shah

Please design the plugin approach so that we can sub-class a theme plugin and override just those parts that we need to change.

#8 - 02/24/2017 02:11 PM - Eric Faulhaber

- Assignee set to Ovidiu Maxiniuc

#9 - 02/24/2017 02:52 PM - Constantin Asofiei

From Greg:

> how much of the color side of the theming is configurable today via the directory vs. being hard-coded?

The color-table and system-colors both can be defined in the directory. Only these two colors in BrowseGuiImpl I think don't have an equivalent in the system-color table:

```
/** Default separator color. */
private static final ColorRgb SEPARATOR_DEFAULT_COLOR = new ColorRgb(192, 192, 192);
```

```
/**
 * Default separator color if background browse color matches {@link #SEPARATOR_DEFAULT_COLOR}.
 */
private static final ColorRgb SEPARATOR_DEFAULT_COLOR2 = new ColorRgb(128, 128, 128);
```

#10 - 02/24/2017 03:00 PM - Ovidiu Maxiniuc

Greg Shah wrote:

> Please design the plugin approach so that we can sub-class a theme plugin and override just those parts that we need to change.

I intend to create a manager/factory that will handle the theme initialization and rendering. It will be configured in the directory with the classname of the theme. This Theme will be responsible for loading all necessary resources (like images, icons) and providing other information needed for layout and rendering. I am thinking to provide these themes as separately packaged: class with additional code and its resources in separate jars. The theme manager should be able to load them if they are added to client's classpath. Of course, the ClassicTheme is always available in main jar and the manager will default to it if the one configured in directory cannot be found/loaded. This way, the newly added Theme implementations can always extend the ClassicTheme to benefit form the existing code.

I have some questions, the main one being how deep should we go? Which window style should be supported?
In windows some things are clear, but on other OS (including linux and mac os) the caption buttons aligned to right of titlebar, title of frame may be centered.
Even on windows 7 and above, the caption buttons are not of same size, the close button is wider. Also, in some themes, these buttons are aligned to the top of the window instead of centered vertically on titlebar.
Although windows10 should be easy to render, windows vista/7 aero style has transparency for non-client area of the window. Probably this is not our target, at least for the moment.

Also, I want to be sure I understand this: the goal of this task is no only to support differently colored widgets in windows classic theme (changing color palette), but rather unlimited support for rendering windows frames and all usual components (radio-buttons/scroll bars/menus/etc). This is difficult to configure in a few lines of xml of the directory compared to a simple palette.

#11 - 02/27/2017 05:29 AM - Greg Shah

> I intend to create a manager/factory that will handle the theme initialization and rendering. It will be configured in the directory with the classname of the theme. This Theme will be responsible for loading all necessary resources (like images, icons) and providing other information needed for layout and rendering.

I think you are on the right track here.

> I am thinking to provide these themes as separately packaged: class with additional code and its resources in separate jars.

Do you plan for one class with a list of widget-specific methods or for separate rendering helper classes for each widget type?

> The theme manager should be able to load them if they are added to client's classpath. Of course, the ClassicTheme is always available in main jar and the manager will default to it if the one configured in directory cannot be found/loaded. This way, the newly added Theme implementations can always extend the ClassicTheme to benefit form the existing code.

Yes, very good.

> I have some questions, the main one being how deep should we go? Which window style should be supported?
> In windows some things are clear, but on other OS (including linux and mac os) the caption buttons aligned to right of titlebar, title of frame may be centered.
> Even on windows 7 and above, the caption buttons are not of same size, the close button is wider. Also, in some themes, these buttons are aligned to the top of the window instead of centered vertically on titlebar.
> Although windows10 should be easy to render, windows vista/7 aero style has transparency for non-client area of the window. Probably this is not our target, at least for the moment.

At this time, only implement pluggable theming for being able to fully swap out classic vs windows 8/10. This means that all differences between these two themes must be handled in a pluggable manner, but you do not need to worry about anything for windows 7/vista.

> Also, I want to be sure I understand this: the goal of this task is no only to support differently colored widgets in windows classic theme (changing color palette), but rather unlimited support for rendering windows frames and all usual components (radio-buttons/scroll bars/menus/etc). This is difficult to configure in a few lines of xml of the directory compared to a simple palette.

Yes. The rendering itself needs to be pluggable. Buttons/scroll bars etc... that render differently would be handled by helper methods that are widget-specific and the widget drawing methods would be left with the generic code that is common to all themes.

#12 - 02/27/2017 08:06 AM - Ovidiu Maxiniuc

Greg Shah wrote:

> I am thinking to provide these themes as separately packaged: class with additional code and its resources in separate jars.

Do you plan for one class with a list of widget-specific methods or for separate rendering helper classes for each widget type?

For the moment, each widget type will have a method in the Theme class for rendering. I expect that if such method receives the BaseConfig as parameter, it should extract much if not all information about rendering. Additional parameters can be added as needed. If such a class is too crowded, it can be refactored at any time. I am thinking of the alternative solution of declaring some render interfaces (like CheckboxRenderer) and getter in Theme class. In a very simple case, the new Theme could implement all these interfaces and in getters to return itself.

> At this time, only implement pluggable theming for being able to fully swap out classic vs windows 8/10. This means that all differences between these two themes must be handled in a pluggable manner, but you do not need to worry about anything for windows 7/vista.

Will it be possible to swap the current theme at runtime? I will not do anything in this direction for the moment, except if this is something trivial like: layout/repaint for all windows.

> Also, I want to be sure I understand this: the goal of this task is no only to support differently colored widgets in windows classic theme (changing color palette), but rather unlimited support for rendering windows frames and all usual components (radio-buttons/scroll bars/menus/etc). This is difficult to configure in a few lines of xml of the directory compared to a simple palette.

> Yes. The rendering itself needs to be pluggable. Buttons/scroll bars etc... that render differently would be handled by helper methods that are widget-specific and the widget drawing methods would be left with the generic code that is common to all themes.

OK

#13 - 02/27/2017 08:20 AM - Greg Shah

> Will it be possible to swap the current theme at runtime? I will not do anything in this direction for the moment, except if this is something trivial like: layout/repaint for all windows.

Good question. Right now, it is NOT a requirement. However, if it is trivial, then it would be useful. At a minimum, it would make for an impressive demo to show people the difference that can be seen by using different themes (especially in the embedded web environment).

#14 - 03/05/2017 12:39 PM - Greg Shah

-------- Forwarded Message --------
Subject: Re: status report
Date: Wed, 1 Mar 2017 23:23:30 +0200
From: Ovidiu Maxiniuc ...
To: Greg Shah ...

Hi,

I created 3209e3 from 3209e and committed my changes. There are some hours of work, but the client already looks changed.
By default, the client starts with Windows classic theme. To change it I use

```
./client.sh -xclient:driver:theme=com.goldencode.p2j.ui.client.gui.theme.Windows10Theme
```

I altered the client.sh script like this:

```
1. process options
   while getopts ":d:stu2z:l:h:c:x:" opt; do
   case $opt in
   d  ) debug=$OPTARG ;;
   s  ) suspend="y" ;;
   t  ) prog="echo "$prog ; test="y" ;;
   l  ) log=$OPTARG ;;
   h  ) host="net:server:host="$OPTARG ;;
   i  ) instance=$OPTARG ;;
   c  ) cfg=$OPTARG ;;
   2  ) srvr="-server" ;;
   u  ) up=$up"/"$up ;;
   x  ) swing="$swing $OPTARG" ;;
   z  ) codepath=$OPTARG ;;
   \? ) for i in "${usage[@]}"; do
   echo -e $i
   done
   exit 1 ;;
   esac
   done
   shift $(($OPTIND - 1))
```

Thanks,
Ovidiu

#15 - 03/05/2017 12:39 PM - Greg Shah

-------- Forwarded Message --------
Subject: Re: status report
Date: Thu, 2 Mar 2017 22:43:15 +0200
From: Ovidiu Maxiniuc ...
To: Greg Shah ...

Hi,

I updated 3209e3 with W10 support for Scrollbars, MessageBox and EditBoxes. Some other issues were fixed.
The previous commit was broken so the build was not adding the resource files.

For tomorrow: Combo-boxes and Menus and maybe Sliders. Finally the cleanup and javadoc that are heavily missing.

For some time in the future: current implementation lacks support for events like mouse hovering visual effects. I added these for all buttons, but some work may need to support all widgets if we are to properly support new Windows themes.

Thanks,
Ovidiu

#16 - 03/05/2017 12:40 PM - Greg Shah

-------- Forwarded Message --------
Subject: Re: status report
Date: Fri, 3 Mar 2017 09:39:33 +0200
From: Ovidiu Maxiniuc ...
To: Eric Faulhaber ...

Eric,

Revision revision 11223 is based on the frozen 3209e.
What is interesting is that W10 (and W8 / 8.1) uses a single user-chosen color to compute the entire palette. It would be nice to find/guess/approximate the functions that create the whole palette. This means that the whole theme can be changed using a single parameter! For the moment I hardcoded constants for each drawn subcomponent.

Thanks,
Ovidiu

On 03/03/2017 12:44 AM, Eric Faulhaber wrote:

> Ovidiu,
>
> Sounds cool, I'm anxious to try it out! If it's not already, would you please rebase 3209e3 to the latest 3209?
>
> Thanks,
> Eric

#17 - 03/06/2017 06:06 AM - Ovidiu Maxiniuc

Task branch 3209e3 was rebased to 3209e/11225.

#18 - 03/06/2017 11:24 AM - Ovidiu Maxiniuc

- File hote_gui_embed_Screenshot_2017-03-06_18-07-20.png added

- File hotel_gui_embed_Screenshot_2017-03-06_18-13-15.png added

- File hotel_gui_embed_Screenshot_2017-03-06_17-23-17.png added

- File hotel_gui_embed_Screenshot_2017-03-06_17-15-18.png added

Attached are a few screen-shots of the embedded mode client captured while the new Windows10 Theme activated (refid #2676).
I used the hotel_gui revno 99 and 3209e3/11228 (rebased from 3209e/11224).

#19 - 03/07/2017 09:55 AM - Eric Faulhaber

Nice work so far.

Nothing against your implementation, but the serif font of the W10 theme looks out of place to me in the context of the surrounding content, plus it looks like we have some problems fitting the new font into the existing widget real estate.

How much effort is it to "subclass" a theme? I'd like to experiment with a non-standard theme that is based on the flatter look of the W10 theme.

#20 - 03/07/2017 09:59 AM - Eric Faulhaber

Is the darker blue of the selected browse row something still on the todo list to be changed?

#21 - 03/07/2017 10:21 AM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> Nice work so far.

Thanks!

> Nothing against your implementation, but the serif font of the W10 theme looks out of place to me in the context of the surrounding content, plus it looks like we have some problems fitting the new font into the existing widget real estate.

I am aware of this font issue. I did not dig at all here. I only know that the new font(s) need to be imported somehow. I will ask details from Constantin. Then I will try to use it/them with the new theme.

> How much effort is it to "subclass" a theme? I'd like to experiment with a non-standard theme that is based on the flatter look of the W10 theme.

That's simple, theoretically. Just pick up an existing one (com.goldencode.p2j.ui.client.gui.theme.Windows10Theme or ClassicTheme from same package) and overwrite the required methods. The code is quite messy for the moment. The methods are named like draw<widget-type><component>() or get<widget-type><attribute>(). I will perform cleanup and add javadoc after I get an acceptable version.

After build and configure client to load the respective Theme you should see you changes. I am also thinking to add a Win8 theme, which is not so flat as Win10. In fact this is why I preferred W10 for startup: being flat is easier to paint, lots of components paints by just fillRect- them with appropriate color. However, there are some differences in layout, that need to be computed.

#22 - 03/08/2017 01:50 PM - Ovidiu Maxiniuc

Task branch 3209e3 was rebased to latest trunk and updated. Current revision is 11147.

#23 - 03/14/2017 02:57 PM - Ovidiu Maxiniuc

Task branch 3209e3 was rebased to latest trunk (11143) and updated. Current revision is 11152.

#24 - 03/20/2017 04:26 PM - Constantin Asofiei

Ovidiu, this is the first half review for 3209e3 rev 11157. I assume you have some 3209e2 changes merged here - the Swing Window/Frame and window resize/location changes - because I think I saw them in another task?

Greg: I suggest adding a way to specify the default theme via the directory, and not just rely on the ThemeManager.crtTheme being hard-coded to some value. If client:driver:theme is not set, then the directory value is checked, and if that is not set either, we can default to classic or win10?

Following review is without src/com/goldencode/p2j/ui/client/gui; I'll finish them tomorrow, because I want to follow the logic, too:

1. ColorRgb - the java.awt import is not needed
2. ThinClient:16254 - you are aligning the || operator like this:

```
          (Keyboard.isMnemonic(evt.keyCode()) ||
           !(src instanceof FillIn            ||
             src instanceof Editor            ||
             src instanceof SelectionList     ||
             src instanceof ComboBox          ||
             src instanceof Browse)) &&
```

   but there is an inner paranthesys and your formattig suggets the || is on the same level...
3. Button - clean hte unused imports (I think the imports have not changed at all... as java.awt.event is not used. And if you remove the imports, there is no functional change in the class.
4. Dimension, DropDown, Label, MouseHandler, p2j.screen.js, AbstractWidget - there is no functional change in these classes
5. MenuElement - is missing history entry
6. I see you've edited the fwd.ico and there is no mention in this task that it was required...
7. ui/client/GuiButton
   - is missing copyright header, javadoc for c'tor and other methods
   - should be in the ui.client.gui package
8. you have comments like this deactivatedWindow.repaint(); // only need border/titlebar - why not add a Window.repaintDecorations() which repaints only the border and titlebar?
9. please explain the changes in chui.driver.swing classes - you replaced Frame with Window?
10. I assume you've renamed AbstractClient.java to AbstractClientFrame.java? Please update the Module in the header and make this explicit in the history entry.
11. there are places where the imports are grouped and split with an empty line, and you removed that - please don't remove that, is easier to follow.
12. GuiWebSocket.isFontInstalled - when is it possible for the result not be a Boolean?

```
         if (res instanceof Boolean)
         {
            return (Boolean) res;
         }
         else
         {
            // log exception and return negative
            new RuntimeException("Not a Boolean: " + String.valueOf(res)).printStackTrace();
            return false;
         }
```

    Do you have an explicit recreate or did you add it just as a precaution?
13. p2j.canvas_renderer.js - I see you made some changes in drawText:
    - is this.ctx.textBaseline = 'alphabetic'; cross-browser compatible?
    - how did you get this formula?

```
      if (centered)
      {
         y = y + Math.round(txtHeight / 2) - Math.round(txtHeight / 6) + 1;
      }
```

#25 - 03/20/2017 04:35 PM - Greg Shah

> Greg: I suggest adding a way to specify the default theme via the directory, and not just rely on the ThemeManager.crtTheme being hard-coded to some value. If client:driver:theme is not set, then the directory value is checked, and if that is not set either, we can default to classic or win10?

Yes. The order of precedence:

1. client:driver:theme
2. directory (lookup for user account, then group, then server, then default)
3. default set in ThemeManager code

win10 should be the default theme if nothing else is specified.

#26 - 03/21/2017 06:44 AM - Ovidiu Maxiniuc

Constantin Asofiei wrote:
Thank you for your review!

> Ovidiu, this is the first half review for 3209e3 rev 11157. I assume you have some 3209e2 changes merged here - the Swing Window/Frame and window resize/location changes - because I think I saw them in another task?

Yes, I built the implementation on 3209e2. For at least two reasons:

1. I was hoping that it would be merged to trunk by the time this task is finished;
2. the old swing implementation was incorrect, 95% or more of the times, I get black boxes instead of windows/dialogs/menus. I still try to use the trunk to compare the new renderings and I have to reopen windows tens of times until I get the image.

> Following review is without src/com/goldencode/p2j/ui/client/gui; I'll finish them tomorrow, because I want to follow the logic, too:
>
> 1. ColorRgb - the java.awt import is not needed

Indeed, removed.

> 1. ThinClient:16254 - you are aligning the || operator like this:
> [...]
> but there is an inner paranthesys and your formattig suggets the || is on the same level...

Correct, fixed.

> 1. Button - clean hte unused imports (I think the imports have not changed at all... as java.awt.event is not used. And if you remove the imports, there is no functional change in the class.

That's correct. I tried to add some code there, but then I extracted it in a new class (GuiButton). I reverted Button to original state.

> 1. Dimension, DropDown, Label, MouseHandler, p2j.screen.js, AbstractWidget - there is no functional change in these classes

Reverted.

1. MenuElement - is missing history entry

The remaining changes are also non functional here, just updated to CGD standard code style. I will revert this file, too.

1. I see you've edited the fwd.ico and there is no mention in this task that it was required...

The old icon file was non-standard 32x23 pixel size. The new one is 32x32.

1. ui/client/GuiButton
   - is missing copyright header, javadoc for c'tor and other methods
   - should be in the ui.client.gui package

Done: moved and javadocs added.

1. you have comments like this deactivatedWindow.repaint(); // only need border/titlebar - why not add a Window.repaintDecorations() which repaints only the border and titlebar?

Thanks for suggestion. I will add such method. I already have thought about something similar. I just need to invalidate the decoration/titlebar/border areas, instead of repainting the whole window.

1. please explain the changes in chui.driver.swing classes - you replaced Frame with Window?

The new implementation uses JFrame s for top-level frames and JDialog s for dialogs. This way we have a natural relation between windows, so there is no need to do Z order manager by hand, using artificial always-on-top property. Because of the abstraction, the ContentPane may receive either of those objects as *application window*. For this I used Window which is the closest common ancestor of JFrame and JDialog.

1. I assume you've renamed AbstractClient.java to AbstractClientFrame.java? Please update the Module in the header and make this explicit in the history entry.

This is almost true. As noted above, the AbstractClient is now split in two classes: AbstractClientFrame that handles parentless window behaviour and AbstractClientDialog which handles windows that have parents (dialogs and menus). I will add the necessary documentation to both files.

1. there are places where the imports are grouped and split with an empty line, and you removed that - please don't remove that, is easier to follow.

I think that is the recommended coding style for import statements.

1. GuiWebSocket.isFontInstalled - when is it possible for the result not be a Boolean?
   [...]
   Do you have an explicit recreate or did you add it just as a precaution?

I encountered a runtime class cast exception. I could not find the cause for it so I added the 'trap'. It didn't fire ever since.

1. p2j.canvas_renderer.js - I see you made some changes in drawText:
   - is this.ctx.textBaseline = 'alphabetic'; cross-browser compatible?

Yes, it should. As you know, browsers may deviate from the standard. In regard to textBaseline property, only the hanging and ideographic are known to have issues, but these alignment are strange. However, the alphabetic alignment is the default value for textBaseline (see https://www.w3.org/TR/2dcontext/#text-styles), and is identical to baseline standard from typography.

> - how did you get this formula?
> [...]

Unfortunately, as you know, the canvas and generally the web API offer little support, if any, for text measuring. The expression you see was obtained empirically, using multiple fonts with close sizes.

#27 - 03/21/2017 09:43 AM - Ovidiu Maxiniuc

The revision 11157 passed the devsrv01 and full ETF tests (no surprise here). However, the latest revision is 11159 (includes changes from 1st review + other improvements).

#28 - 03/21/2017 03:50 PM - Constantin Asofiei

Ovidiu, second part of review:

1. AlertBoxGuiImpl$MessagePanel:447 - there is a left-over javadoc comment
2. LabelGuiImpl - remove the java.awt import
3. I see some more generic menu-related refactoring/changes related to dimension, etc - please checkout menu related tests from the testcases/ project and see if they work OK (both popup and menubar).

Otherwise, the changes look good. And to resume what we've discussed:

1. continue testing widgets side-by-side with windev01 and FWD classic theme (especially where the drawing changes are more than just moving drawing code from the widget to the theme).
2. as there are differences between classic and win10 themes, system fonts and system colors need to be at the theme. See FontTable.readSystemFontTable and try to make this theme-dependent. Same for SysColor.
3. Greg, I assume we should allow different clients having different themes on the same FWD server, right? (this is linked to previous point's system font/color tables)

And something I forgot to mention: I think Stanislav has some WIP changes related to BROWSE; you mentioned you are working on fixing misc issues related to it, please check with Stanislav so you don't duplicate work.

#29 - 03/22/2017 08:25 AM - Greg Shah

> Greg, I assume we should allow different clients having different themes on the same FWD server, right? (this is linked to previous point's system font/color tables)

Yes, correct.

#30 - 03/26/2017 11:19 PM - Eric Faulhaber

Slightly off-topic for this task, but since we're trying to get branch 3209e3 into the trunk next, please note I put in a fix for a NUM-FIELDS buffer handle attribute runtime bug I found that was triggered by some data export ABL code I wrote for the hotel gui sample. The fix is in rev. 11165. It should be very safe, but I ran it through full search ETF testing anyway. It passed.

It is in standard regression testing now. Come to think of it, we added support for this attribute after the regression test environment was settled. My change is isolated to the new code for the attribute, so I just realized this is overkill, but I'll let it complete nonetheless.

#31 - 03/28/2017 03:36 AM - Constantin Asofiei

Ovidiu, some notes about rev 11166:

1. ColorTable is a server-side class, using ThemeManager.getCurrentTheme().getDefaultSystemPalette(); will not work, as this code is client-side, right? Same issue is seen in FontTable, using ThemeManager.getCurrentTheme().getStandardFontTable(); will not work. You need to invoke a client-side API to retrieve these tables and save-them in the work area.
2. FontTable - you have code like wa.getStandardFontTable() repeating in multiple places, save the reference and use that

#32 - 03/28/2017 03:37 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

Constantin Asofiei wrote:

> Ovidiu, some notes about rev 11166:
>
> 1. ColorTable is a server-side class, using ThemeManager.getCurrentTheme().getDefaultSystemPalette(); will not work, as this code is client-side, right? Same issue is seen in FontTable, using ThemeManager.getCurrentTheme().getStandardFontTable(); will not work. You need to invoke a client-side API to retrieve these tables and save-them in the work area.
> 2. FontTable - you have code like wa.getStandardFontTable() repeating in multiple places, save the reference and use that

Constantin,
I updated the 3209e3 (to 11168). I think I fixed issues with ColorTable and FontTable. My previous implementation used Themes on both server and client side. I moved them to client only and use method to remotely send messages. I had some difficulties with fonts so in the end, I had to make the client to push the preferred fonts to server at initialization time and stored there, in the user's context. Could you throw one more look over these changes? Thanks!

#33 - 03/28/2017 04:24 PM - Greg Shah

Please run this branch through ChUI regression testing.

#34 - 03/30/2017 02:53 PM - Ovidiu Maxiniuc

Greg Shah wrote:

> Please run this branch through ChUI regression testing.

3209e3 updated. Current revision is 11172.

The locked AWT thread @ window max/restore operation was caused by BATCH bracketing that blocked the thread until BATCH is finished. I don't like this approach, I think that Swing low level drawing need to be rewritten. Maybe on a dedicated task.

devsrv01 tests passed. The single issue reported by email was caused by a failure of harness to capture the correct screen.
All ETF tests passed.

#35 - 03/30/2017 02:57 PM - Greg Shah

Do you believe that the branch is safe to be merged to trunk?

#36 - 03/31/2017 11:05 AM - Ovidiu Maxiniuc

Greg Shah wrote:

> Do you believe that the branch is safe to be merged to trunk?

I've run this morning some tests and I found some regressions related to recent UI changes. They are fixed now (rev 11174). I am continuing testing, but I think chances to find show-stopper issues are low now.

#37 - 03/31/2017 11:27 AM - Greg Shah

The only other thing we are looking to include is the embedded mode overlay fix. Please coordinate with Sergey, Constantin and Hynek about potentially including that fix in 3209e3. If that cannot be done soon, then we will still want to merge 3209e3 to trunk ASAP.

I'm relying upon your judgement for when it is safe to merge into the trunk.

#38 - 03/31/2017 04:57 PM - Ovidiu Maxiniuc

At the end of the day I deemed it safe. I spotted no regressions in 3209e3. In consequence, the task branch was merged to trunk as revision 11145. The task branch and the devsrv01 results were archived. The standard notification message was sent to team.

#39 - 04/25/2017 01:45 PM - Greg Shah

Ovidiu: could you please provide an example of the directory configuration for specifying the classic theme?

#40 - 04/25/2017 02:22 PM - Ovidiu Maxiniuc

Greg Shah wrote:

> Ovidiu: could you please provide an example of the directory configuration for specifying the classic theme?

Greg, please see the newly added directory.xml.classic.template to hotel_gui project (revision 129).
A note: I used theme as container thinking that in a future revision we will add support for additional configuration of the theme or move in this collection some other existing settings like: color-table, custom-fonts, window-system.

#41 - 04/25/2017 02:31 PM - Ovidiu Maxiniuc

Greg,
As I notified you by email, I added in 2676a/11159 a new theme named Windows8Theme. It further extends Windows10Theme and it is packed as a separate resource, windows8.jar. It proves the plug-in capability of the Theme s but, more important for me, it help me compare pixel-by-pixel the current theme with the baseline screen-shots from reference system (which is a Windows Server 2012 R2 system, with a GUI similar to Windows8.1).

#42 - 05/02/2017 05:59 AM - Constantin Asofiei

Ovidiu, this is a review for 2676a:

- Windows8Theme
  - not all fields/methods have javadoc added
  - line 900 - fields need to be before methods
- SwingEmulatedWindow
  - you have changes related to window z-order management. As I recall, this can be done by the business logic, too: as you've no-op'ed moveToTop and moveToBottom, how will this work?
- SwingGuiDriver
  - again, you've removed explicit window management. This will affect explicit move to top/bottom and also window family support. Also, z-order is maintained internally by FWD, and the driver-level can't handle it properly (even Swing).
- AbstractGuiDriver - has no functional changes
- ComboBoxGuiImpl, MenuItemGuiImpl, WindowBorder, WindowLayout, WindowManager - missing history entry
- EditorGuiImpl
  - you've moved focus logic from mouseClicked to mousePressed - are you sure that a click event will generate a mouse press event first? As I recall, only a click event is generated, in case of a click...
- FillInGuiImpl
  - merge the history entries
- ScrollBarGuiImpl
  - you have this in the history Fixed height of scrollbars (left/top insets were substracted twice) - but there is no change in width() or height(), just a note that // note: super.height() already substracts Y offset (location.y). Shouldn't this be fixed?
- Tooltip
  - how thread-safe is this code?

```
      hideTask = new TimerTask()
      {
         @Override
         public void run()
         {
            if (!tipOn)
            {
               return; // no need to cleanup
            }
            oWindow.repaint(cleanupRect);
            tipOn = false;
```

```
            // removing myself, preventing double firing
            hideTask = null;
        }
    };
```

Shouldn't this post a task via tc.invokeLaterOS? As the timer will run on a different thread.

- NativeInsets - toString is missing javadoc

#43 - 05/02/2017 09:12 AM - Ovidiu Maxiniuc

Constantin,
Thanks for digging my new code. I committed most of the fixes as rev 11162. I will give some thought over the Tooltip code. It apparently work, but indeed, it is not thread-safe.

Constantin Asofiei wrote:

> Ovidiu, this is a review for 2676a:
>
> - Windows8Theme
> - not all fields/methods have javadoc added
> - line 900 - fields need to be before methods

Added.

- SwingEmulatedWindow
  - you have changes related to window z-order management. As I recall, this can be done by the business logic, too: as you've no-op'ed moveToTop and moveToBottom, how will this work?

The windows are grouped in tree at swing/native level. Previous implementation tried to impose temporary invalid order (while sorting) that made the taskbar button to flash. The native windows know their Z-order relative to their parent. This is not true for web side, where the Z-order must be imposed.

- SwingGuiDriver
  - again, you've removed explicit window management. This will affect explicit move to top/bottom and also window family support. Also, z-order is maintained internally by FWD, and the driver-level can't handle it properly (even Swing).

I understand there are window combinations where driver/Swing combination failed to order windows properly? Where can I find testcase?

- AbstractGuiDriver - has no functional changes

Just realized that getUiTheme() is not used anymore. Removed.

- ComboBoxGuiImpl, MenuItemGuiImpl, WindowBorder, WindowLayout, WindowManager - missing history entry

Added. Rolled back changes in WindowLayout.

- EditorGuiImpl
  - you've moved focus logic from mouseClicked to mousePressed - are you sure that a click event will generate a mouse press event first? As I recall, only a click event is generated, in case of a click...

Sorry, you're wrong. From Java8 docs: void mouseClicked(MouseEvent e) is *Invoked when the mouse button has been clicked (pressed and released) on a component.*

- FillInGuiImpl
  - merge the history entries

Done.

- ScrollBarGuiImpl
  - you have this in the history Fixed height of scrollbars (left/top insets were substracted twice) - but there is no change in width() or height(), just a note that // note: super.height() already substracts Y offset (location.y). Shouldn't this be fixed?

They are fixed at lines 256 & 260 when wOff and hOff were computed. They are meant as offsets but they were computed as total difference from super. Since the super already substracts the 3d border width (offset), it lead to incorrect size (double gap at bottom/right).

- Tooltip
  - how thread-safe is this code?
  [...]
  Shouldn't this post a task via tc.invokeLaterOS? As the timer will run on a different thread.

I will review this code.

- NativeInsets - toString is missing javadoc

Added.

#44 - 05/02/2017 09:14 AM - Ovidiu Maxiniuc

Ovidiu Maxiniuc wrote:

> Thanks for digging my new code. I committed most of the fixes as rev 11162.

The correct revision is 11163.

#45 - 05/09/2017 03:32 PM - Constantin Asofiei

Ovidiu, please check something: if the Swing **ChUI** window is closed via ALT-F4 (or its close button), the java process does not terminate; in this case, if the Swing frame is closed, it must terminate...

#46 - 05/10/2017 03:42 PM - Ovidiu Maxiniuc

ChUI window closing issue fixed.
I rewrote the tooltip class but I still have some problems with repaint event that is discarded and doesn't clear the tooltip.

The branch was also rebased. The current revision is 11167.

#47 - 06/13/2017 12:31 PM - Constantin Asofiei

Ovidiu, please let me know what is left with this task.

What I'm aware of in 2676a rev 11170:

- WindowLayout, FrameGUIImpl - no functional change
- WindowTitleBar - missing history entry
- Windows10Theme - line 333 - some indentation is weird
- in Tooltip, the fields TooltipWorker.cancelled, TooltipWorker.displayed shouldn't be volatile (or have some other form of synchronization)?
- about the window-parenting issue: please see the uast/window_parenting/tree-chaining2.p test; unfortunately, there is no easy way to automate this. What you need to do is to minimize i.e. the root, a leaf, etc, unminimize the root, etc, and see how FWD and 4GL behaves. A key note is that the z-order is preserved for the entire minimized tree, and also minimized state for children (minimized before the parent), when a parent is unminimized.

Next steps would be:

- ChUI runtime testing
- test the GUI code with the large GUI app scenarios, and check for UI drawing regressions.

#48 - 06/13/2017 04:04 PM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

Ovidiu, please let me know what is left with this task.

I am not aware of anything else at this moment. Except for the issue you discovered. Thanks for the review.

What I'm aware of in 2676a rev 11170:

- WindowLayout, FrameGUIImpl - no functional change

Reverted.

- WindowTitleBar - missing history entry

I was about to revert this one too, but I saw some discrepancies between the titles of the windows of your testcase FWD & ABL. I will do a short investigation here.

- Windows10Theme - line 333 - some indentation is weird

Reverted contrast colors and reset to standard indentation.

- in Tooltip, the fields TooltipWorker.cancelled, TooltipWorker.displayed shouldn't be volatile (or have some other form of synchronization)?

volatile should be enough in this case.

- about the window-parenting issue: please see the uast/window_parenting/tree-chaining2.p test; unfortunately, there is no easy way to automate this. What you need to do is to minimize i.e. the root, a leaf, etc, unminimize the root, etc, and see how FWD and 4GL behaves. A key note is that the z-order is preserved for the entire minimized tree, and also minimized state for children (minimized before the parent), when a parent is unminimized.

Indeed, this is not working correctly. I will fix it tomorrow.

Next steps would be:

- ChUI runtime testing

Already done this: it passed (as expected).

- test the GUI code with the large GUI app scenarios, and check for UI drawing regressions.

Is there a list of scenarios I can follow?

#49 - 06/14/2017 04:16 PM - Ovidiu Maxiniuc

Ovidiu Maxiniuc wrote:

- WindowTitleBar - missing history entry

I was about to revert this one too, but I saw some discrepancies between the titles of the windows of your testcase FWD & ABL. I will do a short investigation here.

Found two issues here: one (that I fixed) was that the layout of the WindowTitleBar was not always computed so, in some conditions, the text appeared cut or not centered. A second one is really strange: the title of the parent-less window from your testcase should be unknown since their character expression is a concatenation that contains handle value of their unknown parent. If you run it against the ABL test machine, they are not!

- in Tooltip, the fields TooltipWorker.cancelled, TooltipWorker.displayed shouldn't be volatile (or have some other form of synchronization)?

Your example revealed a flaw in tooltip management. I need to review the hiding procedure because when multiple windows are involved, the wrong one is probably selected in driver so this does not work correctly.

- about the window-parenting issue: please see the uast/window_parenting/tree-chaining2.p test; unfortunately, there is no easy way to automate this. What you need to do is to minimize i.e. the root, a leaf, etc, unminimize the root, etc, and see how FWD and 4GL behaves. A key note is that the z-order is preserved for the entire minimized tree, and also minimized state for children (minimized before the parent), when a parent is unminimized.

Indeed, this is not working correctly. I will fix it tomorrow.

This testcase really changed the way I see windows in ABL. I played a little today and I can say that this is a little different from window-application I worked with until now.

I tried to revert to old implementation, but it does not match my observations on ABL test machine. The following process is recursive: if a parent is iconified, all children are hidden (window state is preserved). The process stops for a child is a leaf (does not have other child by itself) or is iconified (its children are already hidden). When restored, the process is reversed. All children are made visible, in recursive manner. We stop the recursion again, when a child is leaf or when it is minimized. In later case, its children will be made visible when the child will be restored expressly, again in a recursive manner.
The iconified (minimized) state of the windows is crucial here and must not be altered. When a top-level window is hidden, it is normally preserved, even if the window is removed from taskbar.

Well, there is something else I did not test yet. How this system behaves if the windows are reparented while they (or their parent) is iconified.

#50 - 06/14/2017 04:21 PM - Constantin Asofiei

Ovidiu Maxiniuc wrote:

> ... the title of the parent-less window from your testcase should be unknown since their character expression is a concatenation that contains handle value of their unknown parent. If you run it against the ABL test machine, they are not!

What does it display? Empty text?

- in Tooltip, the fields TooltipWorker.cancelled, TooltipWorker.displayed shouldn't be volatile (or have some other form of synchronization)?

> Your example revealed a flaw in tooltip management. I need to review the hiding procedure because when multiple windows are involved, the wrong one is probably selected in driver so this does not work correctly.

You can save the window on which the tooltip is drawn, when the tooltip is first shown.

#51 - 06/15/2017 08:44 AM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

> What does it display? Empty text?

The 1st window is labeled as "Window 1 has parent". Like the string(?) = "".
However, if I rewrite the line:
h[i]:title = "Window " + string(i) + " has parent " + string(parents[i]).
as

```
DEF VAR tt AS CHAR.
tt =  "Window " + string(i) + " has parent " + string(parents[i]).
MESSAGE tt tt = ?.
h[i]:title = tt.
```

Then the 1st window's title is indeed, empty or better said unknown since the output of message is ? yes. Interesting or what?

> Your example revealed a flaw in tooltip management. I need to review the hiding procedure because when multiple windows are involved, the wrong one is probably selected in driver so this does not work correctly.

> You can save the window on which the tooltip is drawn, when the tooltip is first shown.

I save the owner widget (the one that generated the tooltip when mouse was hovering it). Then, at the clear time, I use WindowManager.resolveTopLevelWindow(owner) to find the window to be processed. I am investigating now this method which I suspect might not return the correct value.

#52 - 06/16/2017 03:46 PM - Constantin Asofiei

Ovidiu Maxiniuc wrote:

> Constantin Asofiei wrote:
>
> > What does it display? Empty text?
>
> The 1st window is labeled as "Window 1 has parent". Like the string(?) = "".
> However, if I rewrite the line:
> h[i]:title = "Window " + string(i) + " has parent " + string(parents[i]).
> as
> [...]
> Then the 1st window's title is indeed, empty or better said unknown since the output of message is ? yes. Interesting or what?

Yes, that's weird. Please defer this.

> > Your example revealed a flaw in tooltip management. I need to review the hiding procedure because when multiple windows are involved, the wrong one is probably selected in driver so this does not work correctly.
>
> > You can save the window on which the tooltip is drawn, when the tooltip is first shown.
>
> I save the owner widget (the one that generated the tooltip when mouse was hovering it). Then, at the clear time, I use WindowManager.resolveTopLevelWindow(owner) to find the window to be processed. I am investigating now this method which I suspect might not return the correct value.

Do you have a status on this and the other window mngmnt issue?

#53 - 06/19/2017 08:45 AM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

> Your example revealed a flaw in tooltip management. I need to review the hiding procedure because when multiple windows are involved, the wrong one is probably selected in driver so this does not work correctly.

You can save the window on which the tooltip is drawn, when the tooltip is first shown.

I save the owner widget (the one that generated the tooltip when mouse was hovering it). Then, at the clear time, I use WindowManager.resolveTopLevelWindow(owner) to find the window to be processed. I am investigating now this method which I suspect might not return the correct value.

Do you have a status on this and the other window mngmnt issue?

The tooltip issue is fixed. I fixed the windows management, too, but I need to test this more, including the virtual desktop.

#54 - 06/23/2017 05:20 AM - Hynek Cihlar

Ovidiu, is there any estimate when this issue will be merged to trunk?

I am asking because I am implementing some additional window attributes and the implementations will most likely conflict with your changes. So I am planning to skip those conflicting parts. The referenced issue is #3284.

#55 - 06/23/2017 10:32 AM - Ovidiu Maxiniuc

Hynek Cihlar wrote:

> Ovidiu, is there any estimate when this issue will be merged to trunk?
>
> I am asking because I am implementing some additional window attributes and the implementations will most likely conflict with your changes. So I am planning to skip those conflicting parts. The referenced issue is #3284.

I understand. I have an issue with cascaded minimize/restore windows events. They do not match the ABL pattern. I focused temporarily on other tasks but I came back to this one. Hope to get them right today. Then to do the UI tests to make sure nothing else regressed.

#56 - 06/23/2017 03:49 PM - Ovidiu Maxiniuc

I have adjusted the Window management so it duplicate the behaviour I noticed in testcases executed on ABL.
Rebased and committed to 11172.
I am starting heavy testing with GUI scenarios.

#57 - 06/30/2017 03:27 PM - Ovidiu Maxiniuc

- File Tooltip leftover in Web driver.png added

I used the whole set of GUI demo scenarios to test the 11172, comparing in parallel, the ABL on dedicated server, different themes over swing and web desktop.
I encountered some issues but all of them were duplicable with the current trunk. Some scenarios were not matching the screen-shots but they matched the ABL (time intervals?). I have investigated some of those issues but I considered that it is too dangerous to include them in this stable revision.

A single issue I could not identify and proceed to fixing it. Please see this leftover from a tooltip that was already hidden:
!Tooltip leftover in Web driver.png!
Apparently this is a flaw in the clipping engine. The strange thing is that it is not a 'clean' cut that leave the area unpainted at all, instead it looks like a sub-pixel issue (you may need to zoom on it to see the colours are not part of out palette). I only could see this in the titlebar of a window. If tooltip is displayed in 'client area' of the window it is correctly removed from screen. Also this only happen on web driver (I only tested with Firefox 53.0).

In conclusion, I consider the branch stable enough to be merged to trunk.

#58 - 07/04/2017 08:59 AM - Constantin Asofiei

Ovidiu, I'm OK with merging 2676a to trunk - no further testing is needed, as it doesn't collide with 11154 trunk.

#59 - 07/04/2017 01:20 PM - Ovidiu Maxiniuc

2676a was merged to trunk as revision 11155. Test results and the branch itself were archived. Notification email was sent to team.

#60 - 07/04/2017 03:44 PM - Eugenie Lyzenko

- File dnd_test0_1_fwd_20170703a.jpg added

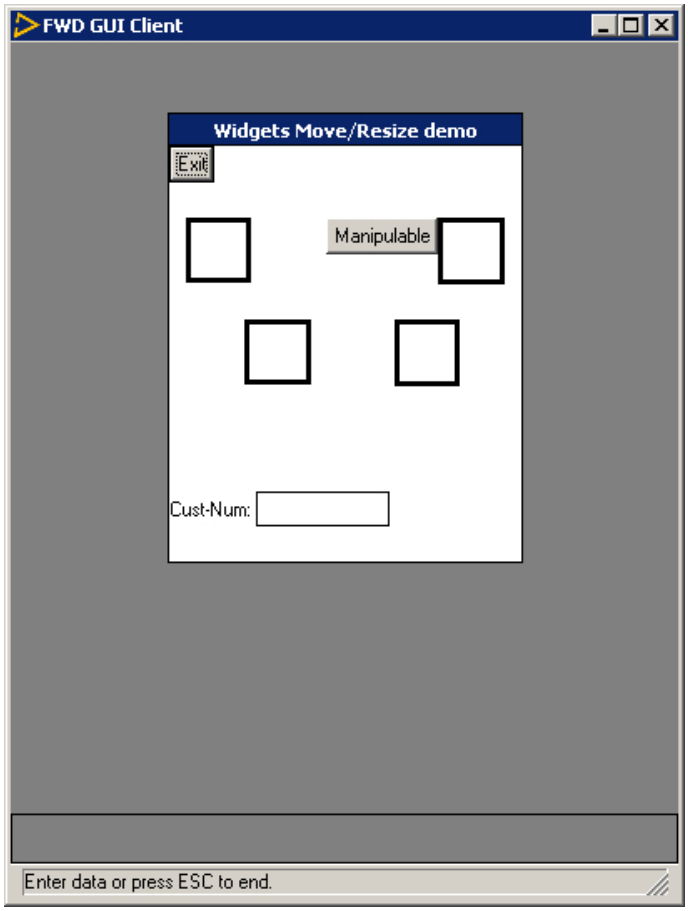
- File dnd_test0_1_fwd_classic_20170704a.jpg added

- File dnd_test0_1_fwd_win10_20170704a.jpg added

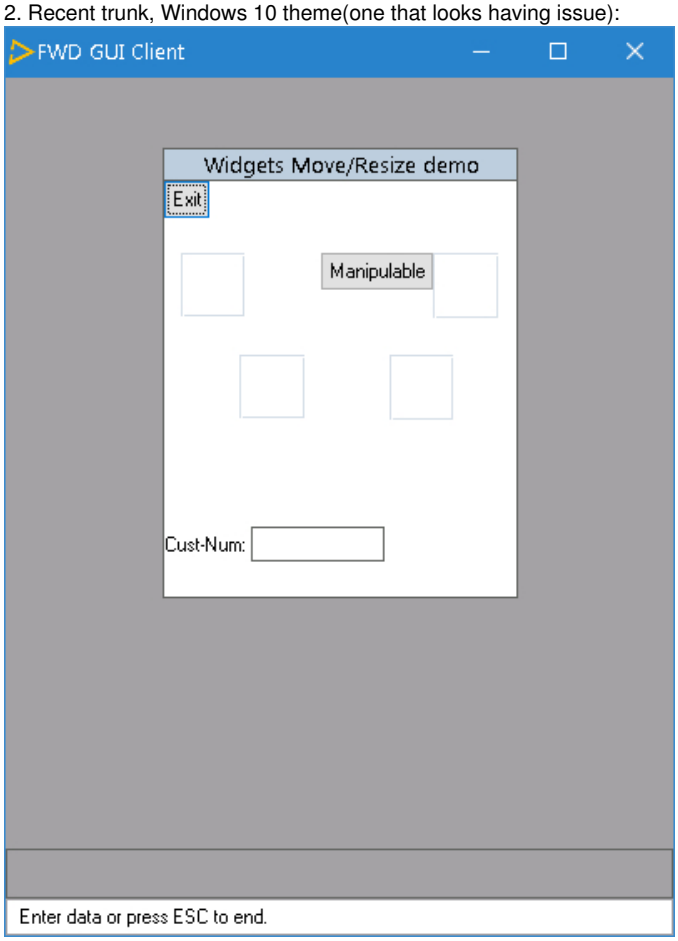
- File dnd_test0_1_4gl.jpg added

I agree, the buttons and other widgets look differently in Windows10 theme. But I do not think the GUI rectangle should change.

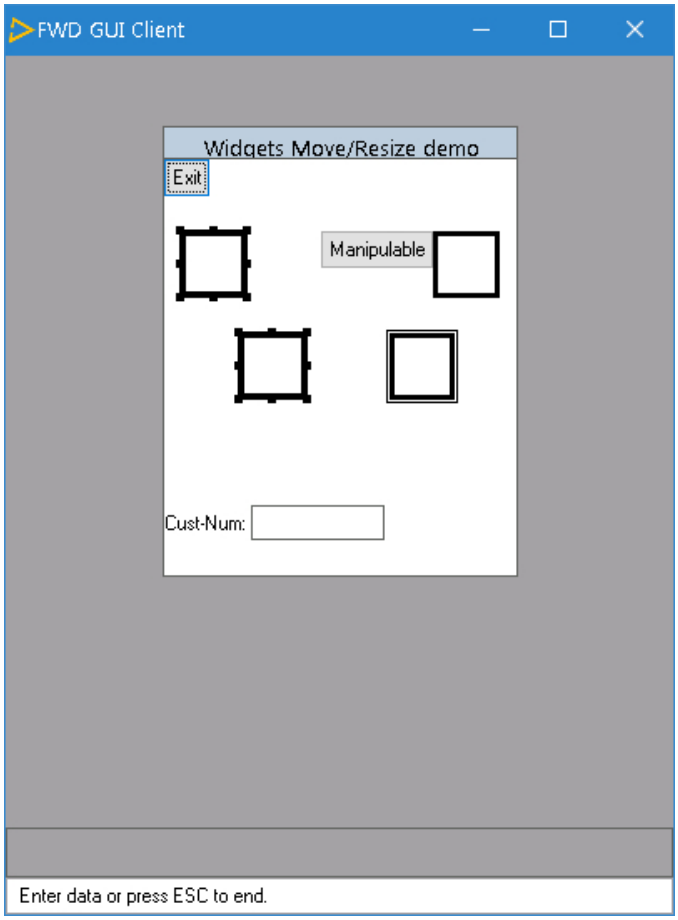
Consider:
1. Recent trunk, classic theme:



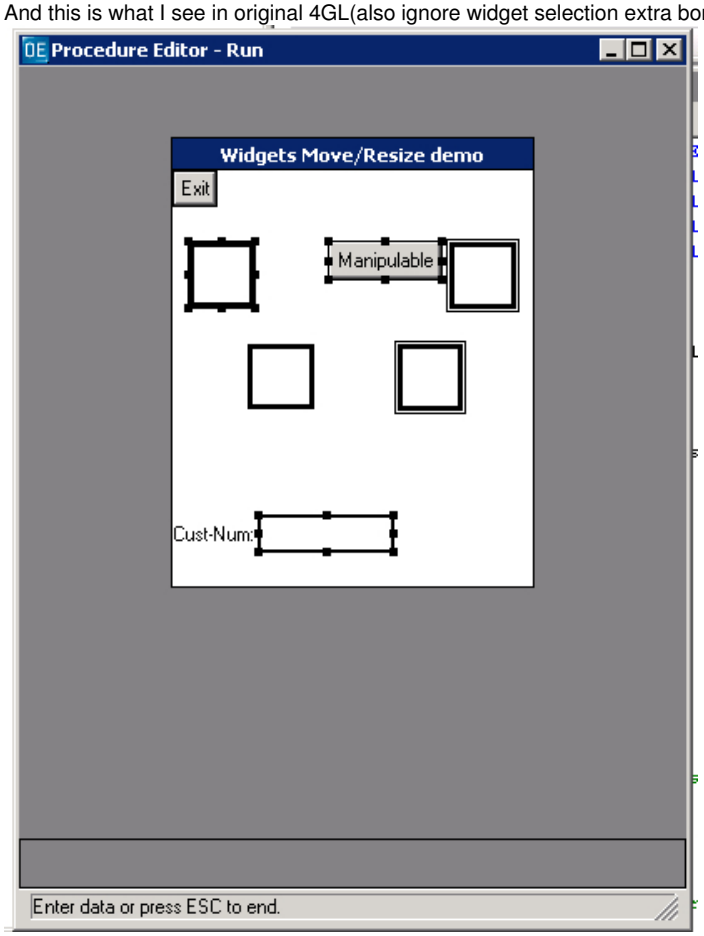
2. Recent trunk, Windows 10 theme(one that looks having issue):



This is Windows10 theme before 11155(please ignore borders around rectangles, this is the not related to your change, only rectangle color and thick):



And this is what I see in original 4GL(also ignore widget selection extra border):



Hope this helps. Again, this is standalone GUI regular rectangles.

#61 - 07/04/2017 03:49 PM - Ovidiu Maxiniuc

Eugenie, please provide the exact 4GL code for this, I will run it on Win8.1 machine and continue investigations.
Thanks!

#62 - 07/04/2017 03:55 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> Eugenie, please provide the exact 4GL code for this, I will run it on Win8.1 machine and continue investigations.
> Thanks!

The test is uast/drag_n_drop/dnd_test0.p accessible in testcases repo. Please refresh your local copy.

#63 - 07/05/2017 03:56 PM - Ovidiu Maxiniuc

Eugenie, you were right, in some cases the line was not correctly drawn on Win8/10. I created a complex testcase (see uast/rectangle/rectangle-tests.p) and fixed it. I also updated the classic theme based on this testcase. Please let me know if you agree to the changes I committed to new branch 2676b (rev 11156). In any case, I owe you a beer. Thanks!

#64 - 07/05/2017 05:03 PM - Eugenie Lyzenko

- File rectangle_tests_4gl_classic_20170705a.jpg added

- File rectangle_tests_fwd_classic_20170705a.jpg added

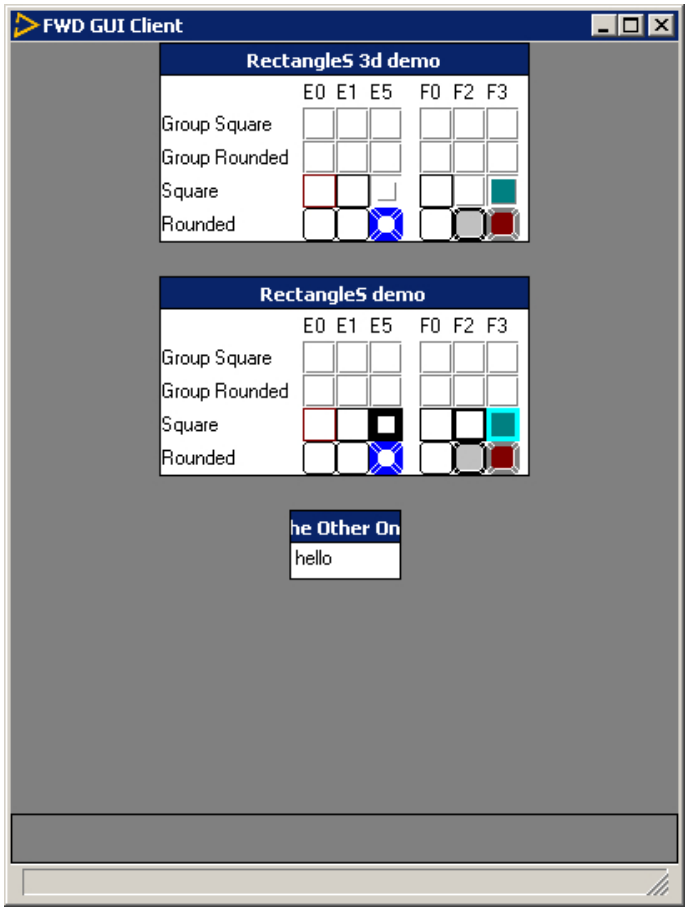
- File rectangle_tests_fwd_win10_20170705a.jpg added

Ovidiu Maxiniuc wrote:

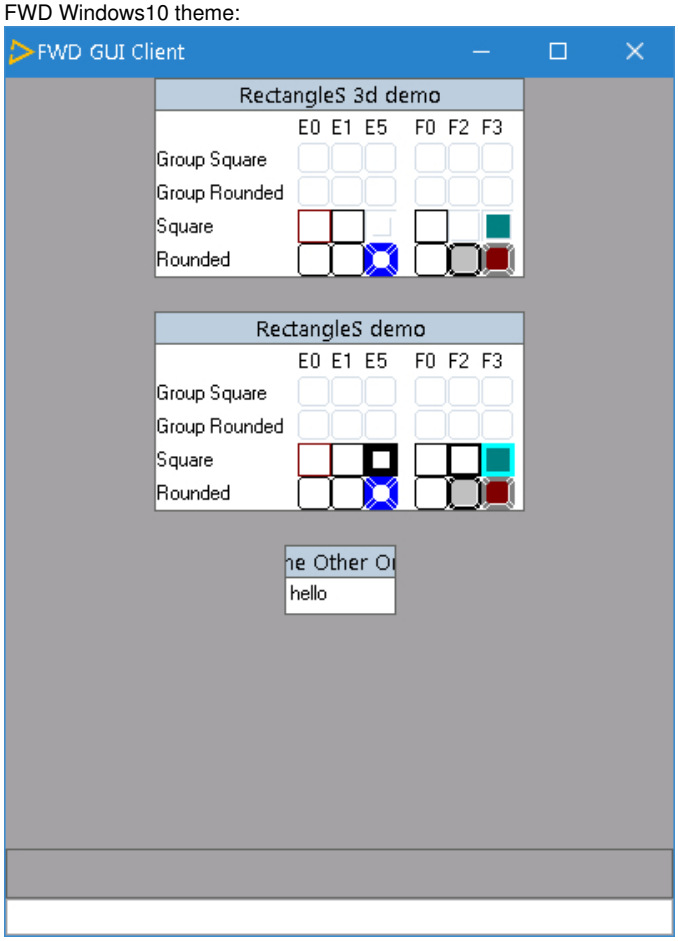
> Eugenie, you were right, in some cases the line was not correctly drawn on Win8/10. I created a complex testcase (see uast/rectangle/rectangle-tests.p) and fixed it. I also updated the classic theme based on this testcase. Please let me know if you agree to the changes I committed to new branch 2676b (rev 11156).

The rectangles looks good. For code changes review I need a bit more time, will answer later.

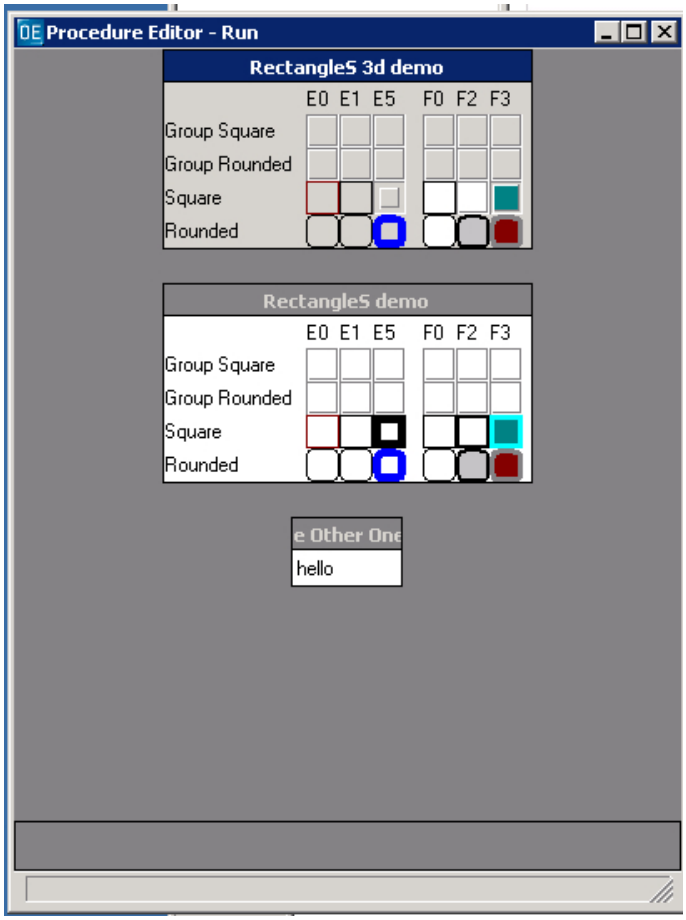
Not sure if this related to this branch changes or not but frame background looks regressed for 3D frames(in classic theme at least). Consider the following pictures:
FWD Classic theme:



FWD Windows10 theme:



4GL Classic theme:



The background of the 3D frame must have button face color, not white. What picture you have for 4GL Windows 8 system?

#65 - 07/05/2017 05:43 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> Please let me know if you agree to the changes I committed to new branch 2676b (rev 11156).

I have no objections for your code changes. But to be completely sure I recommend to run all other rectangle simple tests from uast/rectangle/ to compare output with what we have in 4GL system. Have you checked them?

#66 - 07/06/2017 07:25 AM - Ovidiu Maxiniuc

Eugenie, indeed, there are a couple of issues regarding to frames, but they are not strictly related to styles:

1. note the 4GL, only the 1st frame is selected (blue titlebar). In FWD, this is not the case. This is caused by the fact that the frame titles are rendered as active/selected if one of its widget has focus. Because the frames only contain labels and rectangles that are not focus traversable, FWD is unable to detect the active one, so it paints all the same.
2. the background colour of frames is rendered by BorderedPanelGuiImpl, see lines 233-251. It seems like the bgColor from GuiColorResolver is not correctly set (in case of a 3D frame, it should be 0xd6d3ce for classic theme and 0xf7f3f7 for win8/10 - perhaps xp as well, as they are defined by theme default palette COLOR_3DFACE/BACKGROUND_3DFACE_COLOR). Also the labels should inherit the background colour. In the case of GUI test scenarios, the colour do match because the widget are not put directly in frame, but there are other 3d widgets set as background.

One more question. One can see that our round rectangles have some rendering flaws. Can we fix that (remove the background points that slip through between the concentric round rectangles)? Maybe using a dedicated drawing primitive, if the web driver support it?

I am going now to test each procedure from uast/rectangle and will be back with results.

#67 - 07/06/2017 08:00 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> Eugenie, indeed, there are a couple of issues regarding to frames, but they are not strictly related to styles:
>
> 1. note the 4GL, only the 1st frame is selected (blue titlebar). In FWD, this is not the case. This is caused by the fact that the frame titles are rendered as active/selected if one of its widget has focus. Because the frames only contain labels and rectangles that are not focus traversable, FWD is unable to detect the active one, so it paints all the same.

Yes, I see this too. I think it is not painting issue, FWD can not detect active frame properly.

> 1. the background colour of frames is rendered by BorderedPanelGuiImpl, see lines 233-251. It seems like the bgColor from GuiColorResolver is not correctly set (in case of a 3D frame, it should be 0xd6d3ce for classic theme and 0xf7f3f7 for win8/10 - perhaps xp as well, as they are defined by theme default palette COLOR_3DFACE/BACKGROUND_3DFACE_COLOR). Also the labels should inherit the background colour. In the case of GUI test scenarios, the colour do match because the widget are not put directly in frame, but there are other 3d widgets set as background.

OK. I just wanted to say some time ago 3D frames background was correct so this is a kind of regression we need to fix.

> One more question. One can see that our round rectangles have some rendering flaws. Can we fix that (remove the background points that slip through between the concentric round rectangles)? Maybe using a dedicated drawing primitive, if the web driver support it?

Yes, I see this. When I worked on rectangle GUI implementation I found this is result of how Java2d internals draw the round rectangle. Not our bug.

To avoid this we need to replace java awt routine for drawing round rectangle. Actually I guess this means we will have to ignore java awt curved lines primitives and draw round rectangle by set of straight lines(and possibly including one pixel painting) approximation. This is complex and the question is if we really need this substitution or we can live with current implementation.

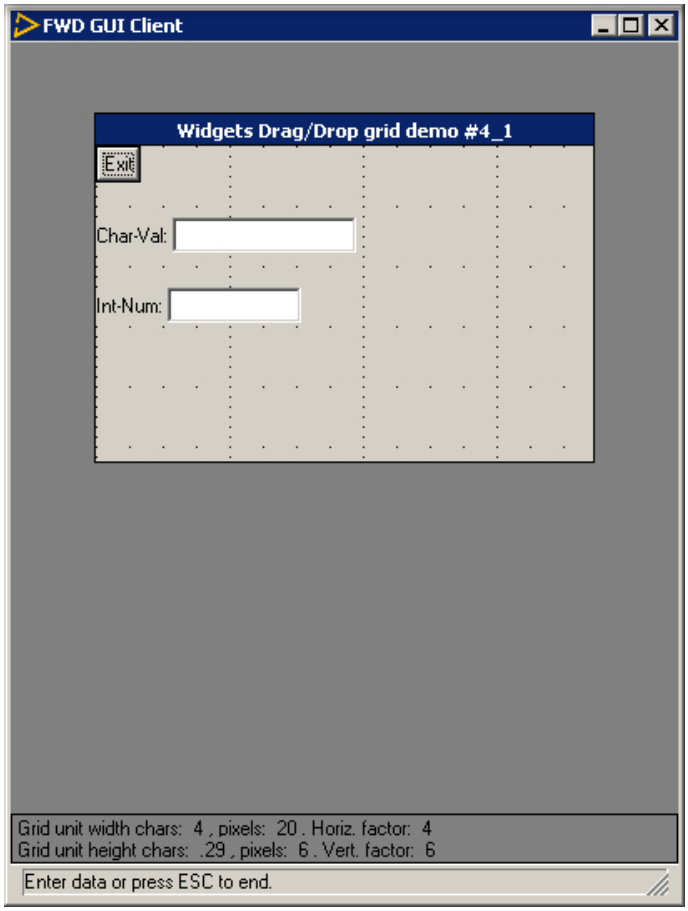
#68 - 07/06/2017 09:40 AM - Eugenie Lyzenko

- File dnd_test4_1_fwd_classic_20170706a.jpg added

Ovidiu Maxiniuc wrote:

> 1. the background colour of frames is rendered by BorderedPanelGuiImpl, see lines 233-251. It seems like the bgColor from GuiColorResolver is not correctly set (in case of a 3D frame, it should be 0xd6d3ce for classic theme and 0xf7f3f7 for win8/10 - perhaps xp as well, as they are defined by theme default palette COLOR_3DFACE/BACKGROUND_3DFACE_COLOR). Also the labels should inherit the background colour. In the case of GUI test scenarios, the colour do match because the widget are not put directly in frame, but there are other 3d widgets set as background.

OK. I've found the situation is a bit different. Frame 3D painting is generally OK. Consider the picture below:



The issue with 3D frame in your uast/rectangle/rectangle-tests.p test is related to conflict in frame definition I guess:

```
...
   WITH FRAME fr3d CENTERED THREE-D NO-LABELS TITLE BGCOLOR 16 "RectangleS 3d demo".
...
```

The two options: THREE-D and BGCOLOR 16 are conflicting here. I think THREE-D must have preference and setting frame BGCOLOR should be ignored.

#69 - 07/06/2017 04:44 PM - Ovidiu Maxiniuc

I finished running uast/rectangle/* tests. I noted two issues:

1. tests 6(1): the gap between the two rectangles is smaller in FWD (both Classic & Win8/10)
2. test 7_4: r2 is white at the center in ABL, and gray in FWD. This is not a rendering issue, the gc.bgColor is 0xd4d0c8 (R212, G208, B200). It has been set in other place.

Something else bothers me: since r11153 (or so) when running with Win10/win8 themes, the client is slightly larger, and fonts do not exactly match any more. I debugged and see that char-to-pixel ratio is no longer 21:5 but 24:6. The classic theme is not affected. How can I revert these?

#70 - 07/06/2017 05:25 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> Something else bothers me: since r11153 (or so) when running with Win10/win8 themes, the client is slightly larger, and fonts do not exactly match any more. I debugged and see that char-to-pixel ratio is no longer 21:5 but 24:6. The classic theme is not affected. How can I revert these?

I think installing and configuring MS fonts we refer as regular ones in server/fonts directory should help.

#71 - 08/10/2017 10:48 AM - Greg Shah

What is the status of 2676b? Does it need review? Are there open issues? What testing has been done?

#72 - 08/15/2017 08:50 AM - Ovidiu Maxiniuc

Greg Shah wrote:

> What is the status of 2676b?

I opened this branch to add fixes for UI glitches every time I spot them while working on other tasks.

> Does it need review?

Some other classes are affected (beside the basic color changes from Theme-s). This should be reviewed as well.

> Are there open issues?

Probably the answer to this question are the those from GUI Known issues.odt sent with notification email when the task branch 2676a was merged to trunk.

> What testing has been done?

The changes were tested at when they were added (ex: color of comboboxes and editor with task branch 3279a). They are mostly edge-case fixes. Conversion is not affected, only manual runtime tests are needed.

#73 - 08/15/2017 03:33 PM - Greg Shah

Code Review Task Branch 2676b Revision 11161

The changes are good.

I prefer to use the GuiDriver.setColor(ColorRgb) instead of GuiDriver.setColor(java.awt.Color). We really need to deprecate that method, because it adds confusion due to the AWT dependency. In ScrollableListGuiImpl there is now a reference to java.awt.Color.WHITE which suggests it is AWT or Swing code.

I have the same preference for the theme classes, but right now the use of java.awt.Color is quite deeply embedded there, so we will leave it. Let's avoid extending that into new classes (like ScrollableListGuiImpl).

#74 - 08/16/2017 09:38 AM - Ovidiu Maxiniuc

I removed the java.awt.Color from ScrollableListGuiImpl. Committed revision 11162. This branch will be merged soon into 1818a.

#75 - 08/16/2017 10:11 AM - Greg Shah

Code Review Task Branch 2676b Revision 11162

The changes are good.

#76 - 08/22/2017 08:59 AM - Greg Shah

The following is some useful information from Ovidiu about how he created the themes. windev01 and windev02 are customer test systems.

> I started to implement the Win10 theme a few of weeks (more likely a month) before I had access to windev02. I used my own testing machine (no 4GL, just running Win10) to capture the design/colors for windows and some widgets (like checkboxes, scrolls, radio-buttons).
>
> When I got access to windev02 (which runs 2012 R2 with a UI similar/identical to Win8) I created the new Windows8Theme. Because of the the already existing code, similarities between the Win8/Win10, and because Win10 theme is flatter than Win8, I decided to extends Win8 from Win10. At this moment I started to focus on Win8 look for which I had an exact 'target'. All previously resources (stored in ./src/com/goldencode/p2j/ui/client/gui/theme/windows10) have been recaptured from new machine (stored in ./src/com/goldencode/p2j/ui/client/gui/theme/windows8).
>
> You cannot configure windev02 (and also the AW VM) to run true Win10 interface, at least not without some kind of addon. However, while working with both windev01 and windev02 I learned that ABL does not paint itself the individual widgets, instead it relies on Windows API. As result, the widgets/components get rendered based on the configuration of the system it runs on.
>
> Because the widgets use multiple colors, they cannot be rendered with our draw primitives so they must be stored as image resources. For capturing these resources I created multiple instances of the widget. For example for radiobox: one disabled and one enabled. Then I made them selected. The difficult part was to capture them while highlighted by mouseover. After capturing, I cut each one using gimp graphic utility and saved as a full-color PNG with transparent background. After adding the image resource and linking in Windows8Theme rendering method I used again the gimp to overlap the zoomed original and FWD captured image to eventually fix offsets and colors.
>
> The colors for rectangles were also captured with color-picker form gimp and have been hardcoded in the Theme implementation.
>
> As noted in #1834, I made a mistake assuming that configuring Remmina to use 24/32 bit color depth will result in exact color match. I don't know why, but it does not. The difference is rather invisible to untrained eye (like 0xA0A0A0(■) -> 0xA2A0A3(■), here they are two grey boxes attached: (■■■)). You need to capture the colors locally on remote machine in order to get the correct palette. This require a little more work than using gimp and you cannot compare the final work (as there is no image transfer to your workstation).

#77 - 09/21/2018 12:23 PM - Greg Shah

- % Done changed from 0 to 100

- Status changed from WIP to Closed

#78 - 09/21/2018 12:23 PM - Greg Shah

- Related to Feature #3684: implement a web theme inspired by material design added

Files

hotel_gui_embed_Screenshot_2017-03-06_18-13-15.png	570 KB	03/06/2017	Ovidiu Maxiniuc
hotel_gui_embed_Screenshot_2017-03-06_17-23-17.png	693 KB	03/06/2017	Ovidiu Maxiniuc
hote_gui_embed_Screenshot_2017-03-06_18-07-20.png	700 KB	03/06/2017	Ovidiu Maxiniuc
hotel_gui_embed_Screenshot_2017-03-06_17-15-18.png	701 KB	03/06/2017	Ovidiu Maxiniuc
Tooltip leftover in Web driver.png	487 Bytes	06/30/2017	Ovidiu Maxiniuc
dnd_test0_1_fwd_classic_20170704a.jpg	40 KB	07/04/2017	Eugenie Lyzenko
dnd_test0_1_fwd_win10_20170704a.jpg	41.3 KB	07/04/2017	Eugenie Lyzenko
dnd_test0_1_fwd_20170703a.jpg	42.6 KB	07/04/2017	Eugenie Lyzenko
dnd_test0_1_4gl.jpg	51.4 KB	07/04/2017	Eugenie Lyzenko
rectangle_tests_fwd_classic_20170705a.jpg	64.1 KB	07/05/2017	Eugenie Lyzenko
rectangle_tests_fwd_win10_20170705a.jpg	72.4 KB	07/05/2017	Eugenie Lyzenko
rectangle_tests_4gl_classic_20170705a.jpg	72.6 KB	07/05/2017	Eugenie Lyzenko
dnd_test4_1_fwd_classic_20170706a.jpg	60.2 KB	07/06/2017	Eugenie Lyzenko