

User Interface - Bug #2678

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

scrollbar button arrow drawing is not 100% identical

09/03/2015 04:38 PM - Greg Shah

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

History

#1 - 09/03/2015 04:39 PM - Greg Shah

See [#2606](#) notes 33 through 61. We are going to try the SVG approach.

Please implement this same approach for the min/max/close buttons as well.

#2 - 09/04/2015 12:30 PM - Eugenie Lyzenko

The task branch 2678a has been created from trunk revision 10931.

#3 - 09/04/2015 01:23 PM - Eugenie Lyzenko

Task branch 2678a for review updated to revision 10932.

This is the SVG related code moved from 2606a revision 10945.

Please implement this same approach for the min/max/close buttons as well.

OK.

#4 - 09/04/2015 08:27 PM - Eugenie Lyzenko

- File `sel_list_p2j_test5_3_0904.jpg` added

Task branch 2678a for review updated to revision 10933.

This update for review contains first working release of the SVG images integration into P2J. The debugging shows the JPEG converter does not work with internal exception in batik code. The PNG converter works fine. The picture with vector based approach is attached here to illustrate.

The image caching is still not ready. The idea is to use "name"+"width"+"height" string as the object key and image data as object inside the HashMap to store already converted images. The theme name should also be considered here.

The application min/max/close button images are on the next days plan.

#5 - 09/05/2015 07:31 PM - Eugenie Lyzenko

Task branch 2678a for review updated to revision 10934.

This update adds caching ability for SVG images handling. Also the unnecessary batik jar files were removed from project:

```
lib/batik-rasterizer-1.8.jar  
lib/batik-squiggle-1.8.jar  
lib/fop-transcoder-allinone-1.1.jar  
lib/js.jar  
lib/xalan-2.7.0.jar  
lib/xerces_2_5_0.jar
```

BTW, some of them conflict with conversion engine so xalan-2.7.0.jar and xerces_2_5_0.jar must be removed mandatory.

#6 - 09/06/2015 09:04 PM - Eugenie Lyzenko

- File *sel_list_p2j_test5_4_0906.jpg* added

Task branch 2678a for review updated to revision 10935.

This update adds min/max/close/restore buttons support. Old painting code has been removed from CaptionButton class. The screenshot with working result is attached here. The cache system is working. In general the look is pretty good. But we can play with SVG source files to have better look. Good news is - no need to modify code(may be it make sense to try TIFF transcoder to check if the result will be better - this will change the code).

#7 - 09/07/2015 09:48 PM - Eugenie Lyzenko

- File *sel_list_p2j_test5_4_0907.jpg* added

Task branch 2678a for review updated to revision 10936.

This update adds SVG based image handling to the ComboBoxGuiImpl. The image pointers checking is added to avoid NPE. Also the image source *.svg files reworked to produce better raster results.

Another finding is we can not use TIFF transcoder due to image processing library. The batik converts to TIFF but ImageIO from TwelveMonkeys we use - does not. So currently only option is to use PNG result.

And I can consider this update as release candidate. We do not need a runtime testing because only GUI code is involved but I prefer to make conversion tests to make sure the conversion process does not interfere with the new libraries added.

#8 - 09/08/2015 07:30 AM - Eugenie Lyzenko

Some news. According to this:

<https://github.com/haraldek/TwelveMonkeys>

the recent TwelveMonkeys must support TIFF and even SVG. So probably we can use only this package, no need to use batik additionally. Testing.

#9 - 09/08/2015 08:22 AM - Eugenie Lyzenko

The findings:

1. Using TIFF does not have advantages, the look is the same as for PNG.
2. Handling SVG from TwelveMonkeys requires batik package to be installed. So no help here too.

Conclusion - we can leave with current implementation.

#10 - 09/08/2015 10:05 AM - Greg Shah

Code Review Task Branch 2678a Revision 10936

Good work!

1. Using the latest revision, please post screen-shots of each resulting image and a comparison image of the same 4GL program running on windev01. The resulting images should be identical for the default sizes.
2. The new code in ImageHelper really doesn't reuse any functionality in the ImageHelper. ImageHelper is in the ui/client/ package because it is used by code there. But the SVG image processing is only going to be used deep in the GUI client. Please create a new class in ui/client/gui/ for the SVG code from ImageHelper.
3. Does getScalableImage() need to close its InputStream?
4. Are the xml-apis-1.3.04.jar, xml-apis-ext-1.3.04.jar and xmlgraphics-commons-2.0.jar from Apache commons? I want to confirm that we can include all of this code. Batik itself is Apache licensed, so it is OK. I just want to make sure all its dependencies are also Apache licensed.

#11 - 09/08/2015 02:26 PM - Eugenie Lyzenko

Greg Shah wrote:

Code Review Task Branch 2678a Revision 10936

Good work!

1. Using the latest revision, please post screen-shots of each resulting image and a comparison image of the same 4GL program running on windev01. The resulting images should be identical for the default sizes.
2. The new code in ImageHelper really doesn't reuse any functionality in the ImageHelper. ImageHelper is in the ui/client/ package because it is used by code there. But the SVG image processing is only going to be used deep in the GUI client. Please create a new class in ui/client/gui/ for the SVG code from ImageHelper.

OK.

3. Does getScalableImage() need to close its InputStream?

4. Are the xml-apis-1.3.04.jar, xml-apis-ext-1.3.04.jar and xmlgraphics-commons-2.0.jar from Apache commons? I want to confirm that we can include all of this code. Batik itself is Apache licensed, so it is OK. I just want to make sure all its dependencies are also Apache licensed.

OK. I'll check.

#12 - 09/08/2015 05:11 PM - Eugenie Lyzenko

- File *sel_list_p2j_test5_5_4gl.jpg* added

- File *sel_list_p2j_test5_5_0908.jpg* added

1. Using the latest revision, please post screen-shots of each resulting image and a comparison image of the same 4GL program running on windev01. The resulting images should be identical for the default sizes.

This is the test selection_list/sel_list_test5_5.p(just uploaded to 1315 revision). It is better to use 8x magnification. You can see all arrows images, min/max/close. The images size is the same but in P2J you can see some smudge from batik renderer. Also our min/max/close buttons itself is a bit bigger than in 4GL. Also I need the test to check the restore buttons and disabled min/max/close.

Another point is our thumb button in scrollbar has lines inside but the 4GL version does not have them.

#13 - 09/08/2015 06:32 PM - Eugenie Lyzenko

- File *LICENSE.xmlgraphics-commons-2.0.txt* added

- File *README.xml-apis-1.3.04.txt* added

- File *README.xml-apis-ext-1.3.04.txt* added

4. Are the xml-apis-1.3.04.jar, xml-apis-ext-1.3.04.jar and xmlgraphics-commons-2.0.jar from Apache commons? I want to confirm that we can include all of this code. Batik itself is Apache licensed, so it is OK. I just want to make sure all its dependencies are also Apache licensed.

According to the files from batik package xmlgraphics-commons-2.0.jar is under Apache license. Others - W3C license. Is it acceptable or not?

#14 - 09/08/2015 08:47 PM - Eugenie Lyzenko

Task branch 2678a for review updated to revision 10937.

Notes resolution.

2. The new code in ImageHelper really doesn't reuse any functionality in the ImageHelper. ImageHelper is in the ui/client/ package because it is used by code there. But the SVG image processing is only going to be used deep in the GUI client. Please create a new class in ui/client/gui/ for the SVG code from ImageHelper.

OK. Moved to gui/VectorGraphicsHelper.java.

3. Does getScalableImage() need to close its InputStream?

Yes, the close() call has been added.

Preparing to rebase.

#15 - 09/08/2015 10:41 PM - Eugenie Lyzenko

Rebased task branch 2678a from P2J trunk revision 10932, new branch revision is 10938.

#16 - 09/09/2015 08:22 AM - Greg Shah

- File LICENSE.dom-software.txt added

The W3C license is acceptable.

1. It is officially approved by OSI: <http://opensource.org/licenses/alphabetical>

2. It is a permissive license. If we don't make any modifications, then the fact that the distribution itself contains the LICENSE.dom-software.txt is sufficient to meet the license terms.

We are OK with the jars you have included.

#17 - 09/09/2015 08:23 AM - Greg Shah

Code Review Task Branch 2678a Revision 10938

The code changes are good.

#18 - 09/09/2015 08:28 AM - Greg Shah

I am OK with you merging to trunk.

However, before you do that I would like to discuss the drawing differences that still exist:

1. The scrollbar button images are fuzzy in P2J but sharp in the 4GL. Is there some antialiasing going on?
2. The min/max/close button images are larger/thicker and fuzzy where they are smaller/thinner and sharp in the 4GL. Is this a result of the titlebar height being too large (is there some kind of scaling going on here)? I guess the same anti-aliasing may also be in effect.

What are your thoughts?

#19 - 09/09/2015 08:42 AM - Hynek Cihlar

Result of the scaled images (title bar and scroll bars) should also be tested. Scaling was one of the motivations to use SVG.

#20 - 09/09/2015 09:31 AM - Eugenie Lyzenko

Greg Shah wrote:

I am OK with you merging to trunk.

However, before you do that I would like to discuss the drawing differences that still exist:

1. The scrollbar button images are fuzzy in P2J but sharp in the 4GL. Is there some antialiasing going on?
2. The min/max/close button images are larger/thicker and fuzzy where they are smaller/thinner and sharp in the 4GL. Is this a result of the titlebar height being too large (is there some kind of scaling going on here)? I guess the same anti-aliasing may also be in effect.

What are your thoughts?

1. The fuzzy/smudge effect is the result of how the batik makes conversion and I think it is antialiasing.
2. I also see the min/max/close button are larger. The images looks the same size but the buttons itself are larger. Windows 4gl size is 16x14, while our p2j - 19x17. The titlebar(area with filled background blue color) height: Windows 4gl is 18, while in P2J - 23. It is important because the min/max/close buttons drawing is not only image painting but plus button frame lines painting and the button border is not the part of the button image.

The most important thing is antialiasing I guess. If we can turn it off - we will have sharp images I think.

#21 - 09/09/2015 10:00 AM - Greg Shah

Does Batik support the shape-rendering SVG property? The crisp-edges value might be what we are looking for.

<http://www.w3.org/1999/07/06/WD-SVG-19990706/render.html#RenderingProperties>

#22 - 09/09/2015 10:25 AM - Hynek Cihlar

Hynek Cihlar wrote:

Result of the scaled images (title bar and scroll bars) should also be tested. Scaling was one of the motivations to use SVG.

Btw. had noticed that some icons in the Windows 7 theme don't scale. For example scroll bar arrows do scale in Windows classic theme while they don't in Windows 7.

#23 - 09/09/2015 10:55 AM - Eugenie Lyzenko

Rebased task branch 2678a from P2J trunk revision 10933, new branch revision is 10939.

#24 - 09/09/2015 06:00 PM - Eugenie Lyzenko

Some findings. During the batik source code investigation I've found the PNGTranscoder creates ImageRenderer object to make SVG->PNG transformation. This renderer has a feature called RenderingHints with antialiasing turned on. That's why we have fussy image I think.

But the class PNGTranscoder provides method to override the default renderer - protected createRenderer. If we subclass PNGTranscoder with our new one, say SharpPNGTranscoder with only overridden method createRenderer and turn the antialias off for created renderer - I guess we get the sharp image on output. Is it acceptable approach?

#25 - 09/09/2015 06:04 PM - Greg Shah

SharpPNGTranscoder with only overridden method createRenderer and turn the antialias off for created renderer - I guess we get the sharp image on output. Is it acceptable approach?

Yes, it sounds good.

#26 - 09/09/2015 07:06 PM - Eugenie Lyzenko

- File sel_list_p2j_test5_5_0909.jpg added

It works. No fuzzy effect anymore. But this shows some SVG images are not good, need to rework. Or may be play with interpolation option for renderer.

#27 - 09/09/2015 11:46 PM - Eugenie Lyzenko

- *File sel_list_p2j_test5_5_0909_sharped.jpg added*

Task branch 2678a for review updated to revision 10940.

This update implements sharp images drawing. Almost all SVG images were reworked. The new class SharpPNGTranscoder turns the antialiasing off. Interpolation variations have the same result so it is unchanged. The output picture is also attached here.

#28 - 09/10/2015 07:48 AM - Greg Shah

Code Review Task Branch 2678a Revision 10940

The changes are good.

The only problem I see is that the bar on the top of the maximize button image is too thick. Fix that and I think you are done.

#29 - 09/10/2015 11:58 AM - Eugenie Lyzenko

- *File sel_list_p2j_test5_5_0910r.jpg added*

- *File sel_list_p2j_test5_5_0910.jpg added*

Task branch 2678a for review updated to revision 10941.

The only problem I see is that the bar on the top of the maximize button image is too thick. Fix that and I think you are done.

Fixed the maximize and restore buttons(we have the ability to see restore too). The fixed screens are attached.

#30 - 09/10/2015 12:46 PM - Greg Shah

Code Review Task Branch 2678a Revision 10941

It looks good.

Please merge this to trunk.

#31 - 09/10/2015 12:59 PM - Eugenie Lyzenko

Merged into trunk as revision 10934. Archived.

#32 - 09/10/2015 01:31 PM - Greg Shah

- *Status changed from WIP to Closed*

#33 - 11/10/2015 04:51 AM - Greg Shah

- *% Done changed from 0 to 100*

#34 - 11/16/2016 12:13 PM - Greg Shah

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

Files

sel_list_p2j_test5_3_0904.jpg	46.9 KB	09/05/2015	Eugenie Lyzenko
sel_list_p2j_test5_4_0906.jpg	56.3 KB	09/07/2015	Eugenie Lyzenko
sel_list_p2j_test5_4_0907.jpg	56.2 KB	09/08/2015	Eugenie Lyzenko
sel_list_p2j_test5_5_4gl.jpg	70 KB	09/08/2015	Eugenie Lyzenko
sel_list_p2j_test5_5_0908.jpg	51.1 KB	09/08/2015	Eugenie Lyzenko
LICENSE.xmlgraphics-commons-2.0.txt	11.1 KB	09/08/2015	Eugenie Lyzenko
README.xml-apis-1.3.04.txt	872 Bytes	09/08/2015	Eugenie Lyzenko
README.xml-apis-ext-1.3.04.txt	593 Bytes	09/08/2015	Eugenie Lyzenko
LICENSE.dom-software.txt	3.28 KB	09/09/2015	Greg Shah
sel_list_p2j_test5_5_0909.jpg	51.3 KB	09/09/2015	Eugenie Lyzenko
sel_list_p2j_test5_5_0909_sharped.jpg	51.2 KB	09/10/2015	Eugenie Lyzenko
sel_list_p2j_test5_5_0910.jpg	51.2 KB	09/10/2015	Eugenie Lyzenko
sel_list_p2j_test5_5_0910r.jpg	51.2 KB	09/10/2015	Eugenie Lyzenko