

User Interface - Feature #2545

Feature # 2252 (Closed): implement GUI client support

Feature # 2446 (Closed): implement BUTTON and IMAGE GUI widgets (runtime and conversion support)

add support for specially named button images

04/09/2015 07:56 AM - Greg Shah

Status:	Closed	Start date:	04/09/2015
Priority:	Normal	Due date:	
Assignee:	Eugenie Lyzenko	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	vendor_id:	GCD
billable:	No		
Description			

History

#1 - 04/09/2015 07:57 AM - Greg Shah

For details see [#2480](#) notes 84 and 85. This task is to implement support for those images.

#2 - 04/14/2015 08:32 PM - Eugenie Lyzenko

The base for embedded images is ready but the investigation shows we do not have the functionality to load them from jar file. It sounds strange but if the approach I offer for these images is:

- store the arrow images on the client side p2j.jar file.
- use the current tools to load button images. The schema we now use is: load image from application jar file, if not found - use client side file system.
- if we do not find the image on local client side file system - try to load from p2j.jar on the client side.
- I suggest to store the images as 32bpp icons at place where we have "gclogo.ico" file.
- use ImageHelper to add missing functionality.

That's the plan.

#3 - 04/15/2015 10:28 AM - Greg Shah

In the 4GL, does one specify the special image name as a quoted string ("btn-up-arrow") or as unquoted text btn-up-arrow?

In the 4GL, is it possible to load your own file btn-up-arrow in preference to the prepackaged one? For example, if you place a file of that same name in the prospath, does it load?

#4 - 04/15/2015 10:58 AM - Eugenie Lyzenko

In the 4GL, does one specify the special image name as a quoted string ("btn-up-arrow") or as unquoted text btn-up-arrow?

Only quoted string("btn-up-arrow") is possible, otherwise even 4GL check syntax does not pass.

In the 4GL, is it possible to load your own file btn-up-arrow in preference to the prepackaged one? For example, if you place a file of that same name in the prospath, does it load?

It is not possible to replace image btn-*arrow with another image with same name. 4GL always load prepackaged images if the file name is specified as "btn-*arrow".

#5 - 04/15/2015 11:02 AM - Greg Shah

- store the arrow images on the client side p2j.jar file.

Yes.

- use the current tools to load button images. The schema we now use is: load image from application jar file, if not found - use client side file system.
- if we do not find the image on local client side file system - try to load from p2j.jar on the client side.

This approach would fail to match the described behavior. Instead, when on GUI only, why not match the special image names and when matched, bypass the app jar load and the local client side file system search?

- I suggest to store the images as 32bpp icons at place where we have "gclogo.ico" file.

Yes.

- use ImageHelper to add missing functionality.

What is missing?

#6 - 04/15/2015 11:24 AM - Eugenie Lyzenko

What is missing?

Loading an image(not default icon for main window) from p2j.jar file.

#7 - 04/15/2015 12:23 PM - Greg Shah

OK, then the only thing I want you to change in the plan is to bypass app jar and local client filesystem searches with the special image names.

#8 - 04/15/2015 08:43 PM - Eugenie Lyzenko

- File *evl_upd20150415a.zip* added

The update has basic implementation for prepackaged button images support. Several notes for what is ready:

- in 4GL prepackaged image names is case insensitive, so the P2J implementation do the same, because icon names are in low case - the requested image name is converted to low case while loading.
- the prepackaged images has priority comparing to application jar images(server side) and file system(client side).
- if for some reasons it is not possible to get image from p2j.jar - we will not search file system for possible alternative.

The prepackaged images has special drawing style so I need to integrate it into current ButtonGuiImpl. Moreover for buttons with prepackaged images it is possible to set only image/image-up. No image-down and/or image-insensitive available to be set. So continue working.

#9 - 04/16/2015 09:40 PM - Eugenie Lyzenko

- File *evl_upd20150416a.zip* added

The next step for review in prepackaged images support:

- the DEFINE BUTTON rule has been changes to add missing SIZE-PIXELS phrase. SIZE and SIZE-CHARS already was there
- minor cleanup for image icons, not all pixel had correct color
- the ButtonWidget has protection from setting down and insensitive images when we have prepackaged images
- the GUI class ButtonGuiImpl has modified to support prepackaged images drawing. To mimic the button background the prepackaged images are considering as transparent.

There are some commented out code in ButtonGuiImpl - this is not final version(I need to test all possible button styles for compatibility with prepackaged images) and possibly move common code in separate methods. So continue working.

#10 - 04/17/2015 08:28 PM - Eugenie Lyzenko

- File *evl_upd20150417a.zip* added

This update for review is merged with the recent code base full support for prepackaged button images. 4 more icons have been added to present the disabled state buttons. Also several small fixes for button disabled state and image based button size computing etc. The disabled prepackaged

images have been loaded on the client side simultaneously with main images. So it is not required another server-client conversation round.

Note for prepackaged images L&F. The button picture is the handcrafted copy of the same button from Windows 4GL. Hope it is not copyrighted.

Due to rule change conversion testing is required.

#11 - 04/18/2015 04:33 PM - Eugenie Lyzenko

All my button tests work fine with this update. So waiting for Hynek to checking in. Or may be there are the notes from review.

#12 - 04/20/2015 02:03 PM - Greg Shah

Code Review evl_upd20150417a.zip

1. In ButtonWidget change the import `com.goldencode.p2j.ui.client.*`; into `import com.goldencode.p2j.ui.client.ImageHelper;`. This is an exception to our normal rule of using wildcards. Normally, we would not want code in the parent package to be importing code from a child package. In this case, the reuse is useful but we don't want it to lead to more usage of that child package.

2. About this:

The button picture is the handcrafted copy of the same button from Windows 4GL. Hope it is not copyrighted.

The pixels of the Windows images are copyrighted. We CANNOT use the clipboard to copy/paste the pixels into our own file. We CANNOT take a screen shot of the images and then save the pixels off into a file. But we CAN duplicate the exact result of the images by hand-editing the pixels in an image editor (e.g. GIMP). In other words, Microsoft and/or Progress cannot stop us from making something ourselves that looks exactly the same. You cannot protect the look of the result, only the actual file/pixels that they have in their implementation are protected by copyright. We don't infringe on their copyright as long as we are by-hand creating images that look the same.

Did you copy/paste the image pixels/files or did you manually draw the pixels using an image editor?

#13 - 04/20/2015 02:38 PM - Eugenie Lyzenko

- File `evl_upd20150420a.zip` added

1. In ButtonWidget change the import `com.goldencode.p2j.ui.client.*`; into `import com.goldencode.p2j.ui.client.ImageHelper;`. This is an exception to our normal rule of using wildcards. Normally, we would not want code in the parent package to be importing code from a child package. In this case, the reuse is useful but we don't want it to lead to more usage of that child package.

OK. The modified file is in update attached.

The pixels of the Windows images are copyrighted. We CANNOT use the clipboard to copy/paste the pixels into our own file. We CANNOT take a screen shot of the images and then save the pixels off into a file. But we CAN duplicate the exact result of the images by hand-editing the

pixels in an image editor (e.g. GIMP). In other words, Microsoft and/or Progress cannot stop us from making something ourselves that looks exactly the same. You cannot protect the look of the result, only the actual file/pixels that they have in their implementation are protected by copyright. We don't infringe on their copyright as long as we are by-hand creating images that look the same.

Did you copy/paste the image pixels/files or did you manually draw the pixels using an image editor?

The only I can say was "stolen" is the size of the images (16x16) and how it looks. Just copy/paste does not work because the quality of this operation is not good enough. So I had to manually create image in GIMP and edit every pixel to have exactly the same result. Then export to 32bpp icon image from GIMP. Is it acceptable approach?

#14 - 04/20/2015 02:58 PM - Greg Shah

Code Review evl_upd20150420a.zip

It is good.

#15 - 04/20/2015 02:59 PM - Greg Shah

The only I can say was "stolen" is the size of the images (16x16) and how it looks.

This isn't stealing. Such things are unprotectable elements of the images. In other words, copyright law does not extend to these things.

Just copy/paste does not work because the quality of this operation is not good enough.

Good, because copy/paste would have been illegal.

So I had to manually create image in GIMP and edit every pixel to have exactly the same result. Then export to 32bpp icon image from GIMP. Is it acceptable approach?

Yes, this is exactly what was needed. There is nothing wrong with this.

#16 - 04/22/2015 07:49 PM - Eugenie Lyzenko

- File *evl_upd20150422a.zip* added

The update has minor javadoc comment fixes and making some constant public to be used in `ComboBoxGuiImpl` when the purpose is common.

#17 - 04/22/2015 08:22 PM - Greg Shah

Code Review *evl_upd20150422a.zip*

Please move the public static final members above the private members. And the private static final members should be moved below the public static final and above the private members.

Otherwise I'm fine with the changes.

#18 - 04/22/2015 08:31 PM - Eugenie Lyzenko

- File *evl_upd20150422c.zip* added

Greg Shah wrote:

Code Review *evl_upd20150422a.zip*

Please move the public static final members above the private members. And the private static final members should be moved below the public static final and above the private members.

Done with update attached.

#19 - 04/22/2015 08:33 PM - Greg Shah

Code Review *evl_upd20150422c.zip*

It looks good.

#20 - 04/28/2015 08:02 PM - Eugenie Lyzenko

Now I can merge *evl_upd20150422c.zip* with 10841 and start testing, OK?

#21 - 04/29/2015 07:34 AM - Eugenie Lyzenko

- File *evl_upd20150429a.zip* added

This is merge with 10842 code base to be ready to regression testing.

#22 - 04/29/2015 09:35 AM - Greg Shah

Code Review *evl_upd20150429a.zip*

The merge looks good.

Constantin has an update that will be checked in before yours. He should be done on Friday. After that, you can merge once more (it should be easier) and then go ahead and test.

#23 - 05/01/2015 03:25 PM - Eugenie Lyzenko

- File *evl_upd20150501a.zip* added

This is the merge with 10845. Now the conversion testing is not required because it was already done (as part of the previous update to 10845) for the changes I've made.

#24 - 05/01/2015 03:34 PM - Greg Shah

Code Review *evl_upd20150501a.zip*

It's good.

Now the conversion testing is not required because it was already done (as part of the previous update to 10845) for the changes I've made.

Agreed.

#25 - 05/03/2015 09:40 AM - Greg Shah

Please merge to bzt 10846 and get this tested.

#26 - 05/03/2015 06:31 PM - Eugenie Lyzenko

- File *evl_upd20150503a.zip* added

The code is the merge with 10846 and small change to label cutting approach. With the recent changes the frame width calculation has been fixed to be compatible with P2J GUI. So I had to change new label size calculation too.

I'm starting the regression in a 1.5 hour but if there is any objections - let me know and I'll interrupt it.

#27 - 05/04/2015 09:27 AM - Greg Shah

Code Review *evl_upd20150503a.zip*

It looks good. Check it in when it passes testing.

#28 - 05/04/2015 09:45 AM - Eugenie Lyzenko

Greg Shah wrote:

Code Review *evl_upd20150503a.zip*

It looks good. Check it in when it passes testing.

The testing completed, no regressions found. The results: *10846_eff7cc1_20150504_evl.zip*. The test TC-MISC-REPORTS-015 had to be run in

standalone mode to get passed. Will be committed and distributed shortly.

#29 - 05/04/2015 10:00 AM - Eugenie Lyzenko

evl_upd20150503a.zip has been committed as 10847.

#30 - 05/04/2015 10:01 AM - Greg Shah

- Status changed from New to Closed

#31 - 11/16/2016 12:13 PM - Greg Shah

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

Files

evl_upd20150415a.zip	219 KB	04/16/2015	Eugenie Lyzenko
evl_upd20150416a.zip	255 KB	04/17/2015	Eugenie Lyzenko
evl_upd20150417a.zip	260 KB	04/18/2015	Eugenie Lyzenko
evl_upd20150420a.zip	260 KB	04/20/2015	Eugenie Lyzenko
evl_upd20150422a.zip	260 KB	04/22/2015	Eugenie Lyzenko
evl_upd20150422c.zip	260 KB	04/23/2015	Eugenie Lyzenko
evl_upd20150429a.zip	260 KB	04/29/2015	Eugenie Lyzenko
evl_upd20150501a.zip	276 KB	05/01/2015	Eugenie Lyzenko
evl_upd20150503a.zip	276 KB	05/03/2015	Eugenie Lyzenko