User Interface - Feature #2479

Feature # 2252 (Closed): implement GUI client support

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

DEFINE IMAGE and image phrase support for both button and image

01/06/2015 05:53 PM - Greg Shah

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Eugenie Lyzenko		**% Done:**	0%
Category:			**Estimated time:**	0.00 hour
Target version:				
billable:	No		**version:**	
vendor_id:	GCD			
Description				
Related issues:				
Related to User Interface - Feature #3332: Add support for load offset and lo...			**Closed**	

History

#1 - 01/06/2015 05:53 PM - Greg Shah

Full Image phrase support for DEFINE BUTTON and DEFINE IMAGE widgets:

size-phrase({SIZE | SIZE-CHARS | SIZE-PIXELS} width BY height)
CONVERT-3D-COLORS
STRETCH-TO-FIT
RETAIN-SHAPE
TRANSPARENT

#2 - 01/07/2015 08:54 AM - Greg Shah

- Subject changed from image phrase support for both button and image to DEFINE IMAGE and image phrase support for both button and image

It is important to note that full DEFINE IMAGE conversion support is part of this task. Currently, we have some provisional support in post-parse-fixups and annotations, but the actual conversion of DEFINE IMAGE is not yet done.

The image-phrase can include an image-size-phrase which uses IMAGE-SIZE, IMAGE-SIZE-CHARS and IMAGE-SIZE-PIXELS. It also can include a FROM clause. Please see the progress.g for details.

I think the DEFINE IMAGE LIKE clause will need work too.

#3 - 01/12/2015 10:45 AM - Greg Shah

Please add CREATE IMAGE support by adding the widget to the list DynamicWidgetFactory.java.

#4 - 01/15/2015 06:04 PM - Eugenie Lyzenko

Greg,

I've found I've missed one rule file update for image widget conversion support: convert/frame_generator.xml. Without this we can not handle/convert DEFINE IMAGE statement. I had this working just forgot to include into respective update. Sorry. I have recovered this from my memory but one issue is still remaining:

The simple code:

```
...
message "Hit a key to start".
pause.

def image moon_pic file "man_on_moon.png".
...
```

converts into(the def image option handling correct in generated UI file):

```
...
            message("Hit a key to start");
            pause();
            "\"man_on_moon.png\""; <--there must be no this line

            FrameElement[] elementList0 = new FrameElement[]
            {
               new WidgetElement(fFrame.widgetMoonPic())
            };
...
```

Can you tell me what rule file I need to change to get rid of this line? Of course if it will not take significant time.

#5 - 01/15/2015 06:08 PM - Greg Shah

> I've found I've missed one rule file update for image widget conversion support: convert/frame_generator.xml. Without this we can not handle/convert DEFINE IMAGE statement. I had this working just forgot to include into respective update.

No problem. Please include it with your next update.

> Can you tell me what rule file I need to change to get rid of this line?

See this in frame_generator.xml:

```
        <!-- Hide static data of DEFINE BUTTON statement -->
        <rule>(type == prog.define_button or
             type == prog.define_rectangle) and
           parent.type == prog.statement
         <action>copy.parent.setHidden(true)</action>
        </rule>
```

Add type == prog.define_image to that and I think the problem will be resolved.

#6 - 01/15/2015 06:35 PM - Eugenie Lyzenko

> Add type == prog.define_image to that and I think the problem will be resolved.

Thank, this works.

#7 - 01/30/2015 05:25 AM - Eugenie Lyzenko

Question. Do we need th support for FROM {X n Y n} | {ROW n COLUMN n} clause of the image phrase?

This sets the origin inside image data to start display image, so we can cut some data from the loaded image. Currently there are no even proper runtime methods to convert this option.

#8 - 01/30/2015 05:35 AM - Greg Shah

Yes, please implement this.

#9 - 01/30/2015 04:51 PM - Eugenie Lyzenko

- File evl_upd20150130a.zip added

This update for your review includes conversion support implementation candidate for size-phrase and image-phrase in DEFINE IMAGE and DEFINE BUTTON statements. All options for these phrases are supported.

The runtime implementation is not here, just stubs to compile converted code.

#10 - 02/03/2015 08:20 PM - Eugenie Lyzenko

- File evl_upd20150203a.zip added

This update for review includes suggest for runtime implementation for image phrase for image widget. In general it works but needs more debugging to be sure the offsets, width, height option combinations works as expected.

#11 - 02/04/2015 08:19 PM - Eugenie Lyzenko

- File evl_upd20150204b.zip added

This update adds missing in previous update ImageConfig class. Also it resolves the offset-size phrase issues found in investigation of the 4GL image handling.

The finding is there are two values for image width, one is used to image drawing and another is used to place the image widget alongside the other widgets. The second one is exactly the image internal natural value. The first one can be defined by size phrase and if not is equal to actual image width.

The same is true for image height too.

#12 - 02/06/2015 03:48 PM - Eugenie Lyzenko

- File evl_upd20150206a.zip added

This update for review presents the approach for image-phrase implementation candidate. Needs to check and debug but base concept is here. The idea is to introduce the new externalizable class ButtonImageDefinition which is data structure for every up, down or insensitive images()including image name. The button config class has instance of this class for every image that can be used. The server side button widget just sets appropriate option and push config to client. The client side button part checks if the size/offset option changed and sets the appropriate value in embedded image object before painting. Then the image handles everything as in image widget itself.

#13 - 02/10/2015 07:38 PM - Eugenie Lyzenko

- File evl_upd20150210a.zip added

This drop includes merging with the recent code base and fix for handling case when requested image size is bigger than the image to draw for button widget. The remaining case is opposite - when requested size is smaller - not so obvious so continue working to find best approach.

#14 - 02/11/2015 01:15 PM - Eugenie Lyzenko

- File evl_upd20150211a.zip added

The update for review includes candidate for image phrase handling for button widget. Need the tests to not getting regressions for image widget.

#15 - 02/11/2015 04:40 PM - Eugenie Lyzenko

- File evl_upd20150211b.zip added

The second update for today has small fixes for image widget bugs/regressions. I guess now if there are no objections/notes the update stage is ready to be tested to commit image-phrase implementation and then go further.

#16 - 02/11/2015 07:53 PM - Greg Shah

Code Review evl_upd20150211b.zip

This is really good.

1. I suspect the IMAGE-SIZE/IMAGE-SIZE-CHARS should be decimal type, not integer. This has to be honored all the way from the frame defs to the drawing. The fractions are actually meaningful.

2. Although it is probably OK, I find it confusing that the image phrase generates different setters for the same options, depending on whether it is on a button or image widget.

3. frame_generator.xml has some hard coded tabs.

4. ImageGuiImpl has 2 history entries that should be merged into a single entry.

5. ImageWidget needs the copyright date to be updated.

Is there anything left to do for this task?

#17 - 02/12/2015 05:13 AM - Eugenie Lyzenko

> Is there anything left to do for this task?

The following options need to be implemented:
CONVERT-3D-COLORS
STRETCH-TO-FIT
RETAIN-SHAPE
TRANSPARENT

#18 - 02/12/2015 07:00 AM - Eugenie Lyzenko

- File evl_upd20150212a.zip added

> 2. Although it is probably OK, I find it confusing that the image phrase generates different setters for the same options, depending on whether it is on a button or image widget.

There are two considerations here I've considered:
1. Button widget can have three different images embedded, while image widget - only one. For every of 3 button image we need to have separate setters.
2. Button has size setter that separate from image phrase, so we can have 4 points where button size can be defined, one related to general button size, other 3 - came from used images sizes. We need to have all and on runtime we have to choose the size that is maximum of them.

The update included for review fixes other notes.

#19 - 02/12/2015 07:17 AM - Greg Shah

Eugenie Lyzenko wrote:

> 2. Although it is probably OK, I find it confusing that the image phrase generates different setters for the same options, depending on whether it is on a button or image widget.

> There are two considerations here I've considered:
> 1. Button widget can have three different images embedded, while image widget - only one. For every of 3 button image we need to have separate setters.

Instead of having 3 sets of setters, you could have 1 set and generate an enum value that distinguishes between ButtonWidget.UP_IMG/ButtonWidget.DOWN_IMG/ButtonWidget.INSENSITIVE_IMG.

Of course, that still doesn't fix the confusion with ImageWidget emitting set[Width|Height]Chars() instead of setImage[Down|Up|Ins]Chars().

Just leave it for now.

> 2. Button has size setter that separate from image phrase, so we can have 4 points where button size can be defined, one related to general button size, other 3 - came from used images sizes. We need to have all and on runtime we have to choose the size that is maximum of them.

I understand. This is required.

#20 - 02/12/2015 07:24 AM - Greg Shah

Code Review evl_upd20150212a.zip

1. All row/col values need to be decimal/double as well. I should have seen this last night. For example: your new ImageWidget.setImageOffset*().

2. Doesn't the client-side drawing layer need changes to handle the decimal/double changes properly?

#21 - 02/12/2015 07:24 AM - Greg Shah

Eugenie Lyzenko wrote:

> Is there anything left to do for this task?

> The following options need to be implemented:
> CONVERT-3D-COLORS
> STRETCH-TO-FIT
> RETAIN-SHAPE
> TRANSPARENT

OK, please get the current update ready and into testing. Then you can work on these as a separate update.

#22 - 02/12/2015 12:02 PM - Eugenie Lyzenko

> Instead of having 3 sets of setters, you could have 1 set and generate an enum value that distinguishes between
> ButtonWidget.UP_IMG/ButtonWidget.DOWN_IMG/ButtonWidget.INSENSITIVE_IMG.

To do this the call:

```
...
execLib("set_attr",...)
...
```

will not work, right?

Can you point me the place to read to implement 2 parameters setter or the example for something already implemented this way.

#23 - 02/12/2015 04:10 PM - Greg Shah

Eugenie Lyzenko wrote:

> Instead of having 3 sets of setters, you could have 1 set and generate an enum value that distinguishes between ButtonWidget.UP_IMG/ButtonWidget.DOWN_IMG/ButtonWidget.INSENSITIVE_IMG.

> To do this the call:
> [...]
> will not work, right?

> Can you point me the place to read to implement 2 parameters setter or the example for something already implemented this way.

Don't worry about this for now. Just leave it as it is.

For future reference, you can look at how we handle the view-as alert-box button type (search on BTN_OK in ui_statements.rules.

#24 - 02/12/2015 04:42 PM - Eugenie Lyzenko

- File evl_upd20150212b.zip added

OK. This update for review changes row/column offsets to double and adjust client side to use new double data.

#25 - 02/13/2015 09:22 AM - Greg Shah

Code Review evl_upd20150212b.zip

It looks good, get it both conversion and runtime regression tested.

#26 - 02/13/2015 12:38 PM - Eugenie Lyzenko

Greg Shah wrote:

> Code Review evl_upd20150212b.zip

> It looks good, get it both conversion and runtime regression tested.

The conversion testing completed without regressions. Generated code is identical. Continue with the runtime testing.

#27 - 02/16/2015 08:51 AM - Eugenie Lyzenko

The runtime testing completed without regressions(separate run for CTRL-C 3 way tests). The results are in 10754_5c96d7c_20150216_evl.zip.

So the update evl_upd20150212b.zip is ready to be committed and distributed.

#28 - 02/16/2015 09:38 AM - Greg Shah

Check it in.

#29 - 02/16/2015 09:51 AM - Eugenie Lyzenko

Committed in bzr as 10757.

#30 - 02/18/2015 06:06 PM - Eugenie Lyzenko

- File evl_upd20150218a.zip added

This update for your review adds transparency runtime implementation for image widget. Also it prepares other boolean option to be implemented: STRETCH-TO-FIT, RETAIN-SHAPE and CONVERT-3D-COLORS. Continue with size/stretch related(STRETCH-TO-FIT, RETAIN-SHAPE).

#31 - 02/19/2015 09:45 AM - Greg Shah

Code Review evl_upd20150218a.zip

It looks good.

#32 - 02/19/2015 08:01 PM - Eugenie Lyzenko

- File evl_upd20150219a.zip added

This update adds conversion rule change to support RETAIN-SHAPE sub-option and runtime implementation for STRETCH-TO-FIT and RETAIN-SHAPE.

The default values for both STRETCH-TO-FIT and RETAIN-SHAPE attributes are false/no. The RETAIN-SHAPE is meaningful only if STRETCH-TO-FIT == true. The idea of implementing is: when we fit the picture to display area we have two scaling factors, for X and Y. If we need to keep aspect ration - we have to choose only one for both direction and this scaling is the smaller one from two different to have at least one size to be exact the same as display area size.

Continue with debugging and CONVERT-3D-COLORS option runtime implementation.

#33 - 02/20/2015 02:22 PM - Greg Shah

Code Review evl_upd20150219a.zip

It is good.

#34 - 02/20/2015 03:40 PM - Eugenie Lyzenko

- File evl_upd20150220a.zip added

This updates for review includes runtime implementation of the CONVERT-3D-COLORS option for image and button widgets. The idea is to use another RGB filter to process image before drawing. Had to add creation BufferedImage object after filter apply because if we have two sequential

filters, transparent and convert 3d - we need BufferedImage as the filter input, not Image produced by Toolkit.getDefaultToolkit().createImage(ip).

The button widget's NO-CONVERT-3D-COLORS attribute is also now supported. When the button image is loading the value of this attribute is passing to embedded image config as ImageConfig.convert3d == !ButtonConfig.noConvert3D. The other reaction is happening on underlying image widget level.

Also the code to detect if we use icon as image has been added. The reason is the options STRETCH-TO-FIT, RETAIN-SHAPE have meaning only if the loaded image is not an icon. The simple check is to find if the image filename has ".ico" part or not.

If there are no notes, today I'll make more testing and if there are no issues - looks like the change is ready to be tested.

#35 - 02/20/2015 03:52 PM - Greg Shah

Code Review evl_upd20150220a.zip

1. In Convert3DColorsFilter, are the input colors to be converted really hard coded in the 4GL?

2. In ImageGuiImpl, the checking of the .ico in the filename seems fragile. The filename doesn't have to have that, but can still be loaded as an icon. Likewise, if we load from a server-side resource, this filename extension would not be visible. Don't we already go through a special path to load an icon? Can't we just set the isIcon as part of that process?

#36 - 02/20/2015 04:06 PM - Eugenie Lyzenko

> 1. In Convert3DColorsFilter, are the input colors to be converted really hard coded in the 4GL?

I think so, from OpenEdge: ABL Reference, page 412.

> 2. In ImageGuiImpl, the checking of the .ico in the filename seems fragile. The filename doesn't have to have that, but can still be loaded as an icon. Likewise, if we load from a server-side resource, this filename extension would not be visible. Don't we already go through a special path to load an icon? Can't we just set the isIcon as part of that process?

Yes, this is weak place. And yes, we already have isIcon == true when we load icon by LOAD-*ICON. But what if we will use LOAD-IMAGE() or define image and just use "some/path/filename.ico" file. This means we working with image but really the data will be icon. And for these cases we can check the filename extension to have info about file data internals.

#37 - 02/20/2015 04:32 PM - Greg Shah

OK. I don't think this needs full runtime testing. The GUI stuff won't get loaded in MAJIC testing. Even the server-side Image*.java should not be loaded. I think this is safe enough to check in.

#38 - 02/20/2015 04:34 PM - Greg Shah

Have you run your testcases in both windev01 and in P2J? Does our implementation match exactly?

#39 - 02/20/2015 04:41 PM - Eugenie Lyzenko

OK. I don't think this needs full runtime testing. The GUI stuff won't get loaded in MAJIC testing. Even the server-side Image*.java should not be loaded. I think this is safe enough to check in.

Agreed. I just would like to make conversion test. I've changed the rule file and want to make sure there will be no regression her. Is it OK?

#40 - 02/20/2015 04:57 PM - Eugenie Lyzenko

Have you run your testcases in both windev01 and in P2J? Does our implementation match exactly?

Yes, I'm comparing windev01 and P2J versions layout for every test. There are some mismatches when the units are characters. For pixel based units the match is OK.

And I have found regression from recent size related option changes to the previously implemented button relates IMAGE-SIZE options. So need to debug and fix it.

#41 - 02/20/2015 04:59 PM - Greg Shah

I've changed the rule file and want to make sure there will be no regression her. Is it OK?

Good point. Yes, conversion testing is needed.

#42 - 02/20/2015 05:00 PM - Greg Shah

Eugenie Lyzenko wrote:

> Have you run your testcases in both windev01 and in P2J? Does our implementation match exactly?

> Yes, I'm comparing windev01 and P2J versions layout for every test. There are some mismatches when the units are characters. For pixel based units the match is OK.

> And I have found regression from recent size related option changes to the previously implemented button relates IMAGE-SIZE options. So need to debug and fix it.

Please do fix both of these issues (character unit mismatches and the regression for IMAGE-SIZE options) as part of this task.

#43 - 02/20/2015 05:21 PM - Eugenie Lyzenko

> Please do fix both of these issues (character unit mismatches and the regression for IMAGE-SIZE options) as part of this task.

I have one question here. Looks like the character units mismatches are caused the fact we have different character width and height in pixels for windev01 and P2J. And I think this can be not always a bug. For different systems we have different font metrics. The only we can verify here is proportions, meaning coordinates in character units should correspond with total window size in character units. Moreover we can not provide for all possible systems to have exact character size in pixels, we can request the system to have 8x16 font but this will not guarantee the final font will be 8x16, we get the best match instead. What do you think about our strategy here?

#44 - 02/22/2015 07:03 AM - Eugenie Lyzenko

- File evl_upd20150222a.zip added

The update fixes the size/location regression for images used to draw the buttons. Plus slightly enhanced icon detection approach for image widgets. The already detected icons new excluded from the process. The conversion testing is in progress.

The character size issue/mismatch is still untouched. So what is our plan here? Need to fix in this task, or separately?

#45 - 02/22/2015 02:53 PM - Eugenie Lyzenko

The conversion testing completed, generated codes are identical. No regressions.

So the update is ready to be committed if we separate character size fix to another task.

#46 - 02/22/2015 03:32 PM - Greg Shah

Yes, go ahead and commit it.

#47 - 02/22/2015 04:05 PM - Eugenie Lyzenko

Greg Shah wrote:

> Yes, go ahead and commit it.

Committed in bzr as 10771.

#48 - 02/23/2015 07:19 AM - Eugenie Lyzenko

The remaining thing I'm going to implement is having getters/setters for all attributes in image and button widgets that can be query set outside of the DEFINE IMAGE or DEFINE BUTTON statements:

```
CONVERT-3D-COLORS
STRETCH-TO-FIT
RETAIN-SHAPE
TRANSPARENT
```

For example:

```
...
bVar = button:convert-3d-colors.
button:convert-3d-colors = not bVar.
...
```

#49 - 02/23/2015 12:20 PM - Eugenie Lyzenko

- File evl_upd20150223a.zip added

This update for review adds boolean attributes setters/getters outside of the DEFINE IMAGE or DEFINE BUTTON statements. Including dynamic widget case when we reference the widget by handle. Also fixes the NPE in Button.java when width calculating for dynamic button without having text label defined.

The UI artifact has been detected while implementation and testing this feature. The X,Y coordinates can not be set for dynamic button or image(not sure about all other dynamic widgets). The dynamic widgets are painting at top left corner of the frame. This can be fixed as part of the other UI issues related task I think because this is not DEFINE IMAGE specific issue.

#50 - 02/23/2015 12:37 PM - Eugenie Lyzenko

- File deleted (evl_upd20150223a.zip)

#51 - 02/23/2015 12:39 PM - Eugenie Lyzenko

- File evl_upd20150223a.zip added

Sorry the update 0223a was inconsistent(has not final versions). Please see the new drop.

#52 - 02/23/2015 01:07 PM - Greg Shah

Code Review evl_upd20150223a.zip

The changes look good.

Normally, I would ask you to get these tested, but Igor has overlapping changes being tested in #2498 and he has had difficulty getting them through testing. I'd like to see if we can get his changes checked in before your update gets tested.

Thanks.

#53 - 02/23/2015 01:29 PM - Eugenie Lyzenko

> Normally, I would ask you to get these tested, but Igor has overlapping changes being tested in #2498 and he has had difficulty getting them through testing. I'd like to see if we can get his changes checked in before your update gets tested.
>
> Thanks.

I see. Let's wait while Igor's changes become committed.

#54 - 02/24/2015 06:33 PM - Eugenie Lyzenko

- File evl_upd20150224a.zip added

This update for review merges the recent code base plus fixes the location issue for dynamic images and buttons. The idea is the coordinates for dynamic widgets is defined later than the widget initially placed in the frame. Moreover the coordinates for 4GL dynamic widgets must be explicitly defined, no automatic layout. So on the client side we need to know if the widget dynamic or not - to do this we need to store dynamic flag in widget config(and we even have the place in config). When it is time to draw the widget - the actual coordinates of the dynamic widgets will be taken from the other place than for static widgets - where we store them as result of the setRow()/setColumn() calls.

#55 - 02/25/2015 08:51 AM - Eugenie Lyzenko

The only thing I see is to unify CONVERT-3D-COLORS attribute for button and image widgets. Currently button has NO-CONVERT-3D-COLORS option and uses it to support CONVERT-3D-COLORS attribute. Looks like it makes sense to have the attribute CONVERT-3D-COLORS for button too and use it to store current NO-CONVERT-3D-COLORS option as NO-CONVERT-3D-COLORS == !CONVERT-3D-COLORS. What do you think?

#56 - 02/25/2015 08:53 AM - Greg Shah

Eugenie Lyzenko wrote:

> The only thing I see is to unify CONVERT-3D-COLORS attribute for button and image widgets. Currently button has NO-CONVERT-3D-COLORS option and uses it to support CONVERT-3D-COLORS attribute. Looks like it makes sense to have the attribute CONVERT-3D-COLORS for button too and use it to store current NO-CONVERT-3D-COLORS option as NO-CONVERT-3D-COLORS == !CONVERT-3D-COLORS. What do you think?

Agreed.

#57 - 02/25/2015 09:05 AM - Greg Shah

Code Review evl_upd20150224a.zip

Are ButtonGuiImpl.draw() and ImageGuiImpl.draw() really the proper locations to calculate location for a dynamic widget? I'm worried that:

- we will have to implement this separately for every widget, I'd prefer an approach that is entirely (or mostly) implemented in a base class
- this doesn't cover the ChUI case (for button)
- it seems like we may have other layout or state changes needed besides where we draw

Constantin/Hynek: please review and comment on the changes in ButtonGuiImpl and ImageGuiImpl and BaseEntity.

#58 - 02/25/2015 09:16 AM - Eugenie Lyzenko

One more possible TODO change is rework for image phrase setters for button related images we have discussed previously. To have single setter for all Up, Down, Insensitive images and setters for offset(chars|pixels)/size(chars|pixels). Do we need this rework now?

#59 - 02/25/2015 09:17 AM - Greg Shah

Eugenie Lyzenko wrote:

> One more possible TODO change is rework for image phrase setters for button related images we have discussed previously. To have single setter for all Up, Down, Insensitive images and setters for offset(chars|pixels)/size(chars|pixels). Do we need this rework now?

No, I don't want to do this now.

#60 - 02/25/2015 09:33 AM - Hynek Cihlar

Greg Shah wrote:

> Code Review evl_upd20150224a.zip
>
> Are ButtonGuiImpl.draw() and ImageGuiImpl.draw() really the proper locations to calculate location for a dynamic widget? I'm worried that:
>
> - we will have to implement this separately for every widget, I'd prefer an approach that is entirely (or mostly) implemented in a base class
> - this doesn't cover the ChUI case (for button)
> - it seems like we may have other layout or state changes needed besides where we draw
>
> Constantin/Hynek: please review and comment on the changes in ButtonGuiImpl and ImageGuiImpl and BaseEntity.

Ideally the draw() method should not perform any positional logic. The COL/ROW attributes are synced with the client widget instance as part of BaseConfig.applyConfig() execution - eventually AbstractWidget.setLocation() is called and I think this is the location where the conversion to native/physical units should happen (and vice versa in AbstractWidget.setPhysicalLocation()). Then it should be enough to only call physicalLocation() in draw().

Is there anything special on the widget position when it is dynamic?

#61 - 02/25/2015 12:07 PM - Eugenie Lyzenko

- File evl_upd20150225a.zip added

This update for review has general fix for dynamic widget location issue. All processing shifted to BaseConfig.applyConfig(). I have left BaseEntity change to pass dynamic flag to client side because if we will even not use this on client side(as it is for now) - we pass anyway this value to client side and I think it is better to have dynamic value correct for dynamic widget.

Also the update has rework for usage NO-CONVERT-3D-COLORS/CONVERT-3D-COLORS option and attribute.

#62 - 02/25/2015 02:11 PM - Greg Shah

Code Review evl_upd20150225a.zip

I'm fine with the changes.

Hynek: please look at the BaseConfig change and confirm if it is OK.

#63 - 02/25/2015 03:36 PM - Hynek Cihlar

Greg Shah wrote:

> Code Review evl_upd20150225a.zip
>
> I'm fine with the changes.
>
> Hynek: please look at the BaseConfig change and confirm if it is OK.

I think it is good. Just a minor thing - the call to setPhysicalLocation in IGI.initialize should be unnecessary as you already set the physical location thanks to the change in BC.applyConfig.

#64 - 02/25/2015 03:52 PM - Greg Shah

> the call to setPhysicalLocation in IGI.initialize should be unnecessary as you already set the physical location thanks to the change in BC.applyConfig.

Eugenie: please confirm that it is not needed. If so, please remove it. Then go ahead with both conversion and runtime regression testing.

#65 - 02/25/2015 03:58 PM - Eugenie Lyzenko

- File evl_upd20150225b.zip added

> Eugenie: please confirm that it is not needed. If so, please remove it. Then go ahead with both conversion and runtime regression testing.

Yes, this is not needed. The attached file has modified version. Starting the testing for evl_upd20150225b.zip

#66 - 02/25/2015 07:48 PM - Eugenie Lyzenko

The conversion testing completed. Codes are identical, no regression. Continue with the runtime testing.

#67 - 02/26/2015 11:47 AM - Eugenie Lyzenko

The main part of the regression tests completed OK. Waiting for CTRL-C to be done.

#68 - 02/26/2015 03:14 PM - Eugenie Lyzenko

Testing completed. The results file: 10778_5c96d7c_20150226_evl.zip.

So the update is ready to be committed and distributed.

#69 - 02/26/2015 03:20 PM - Greg Shah

Yes, please check it in.

#70 - 02/26/2015 03:38 PM - Eugenie Lyzenko

The update evl_upd20150225b.zip has been committed in bzr as 10780.

#71 - 02/26/2015 04:07 PM - Greg Shah

- Status changed from New to Closed

#72 - 09/05/2017 12:25 PM - Hynek Cihlar

- Related to Feature #3332: Add support for load offset and load size parameters of LOAD-IMAGE methods of BUTTON and IMAGE widgets added*

Files

evl_upd20150130a.zip	39.2 KB	01/30/2015	Eugenie Lyzenko
evl_upd20150203a.zip	75.1 KB	02/04/2015	Eugenie Lyzenko
evl_upd20150204b.zip	76.6 KB	02/05/2015	Eugenie Lyzenko
evl_upd20150206a.zip	86 KB	02/06/2015	Eugenie Lyzenko
evl_upd20150210a.zip	91.3 KB	02/11/2015	Eugenie Lyzenko
evl_upd20150211a.zip	91.7 KB	02/11/2015	Eugenie Lyzenko
evl_upd20150211b.zip	91.7 KB	02/11/2015	Eugenie Lyzenko
evl_upd20150212a.zip	91.6 KB	02/12/2015	Eugenie Lyzenko
evl_upd20150212b.zip	86.6 KB	02/12/2015	Eugenie Lyzenko
evl_upd20150218a.zip	76 KB	02/18/2015	Eugenie Lyzenko
evl_upd20150219a.zip	76.5 KB	02/20/2015	Eugenie Lyzenko
evl_upd20150220a.zip	83.3 KB	02/20/2015	Eugenie Lyzenko
evl_upd20150222a.zip	83.5 KB	02/22/2015	Eugenie Lyzenko
evl_upd20150223a.zip	93.8 KB	02/23/2015	Eugenie Lyzenko
evl_upd20150224a.zip	112 KB	02/24/2015	Eugenie Lyzenko
evl_upd20150225a.zip	119 KB	02/25/2015	Eugenie Lyzenko
evl_upd20150225b.zip	119 KB	02/25/2015	Eugenie Lyzenko