

User Interface - Feature #1788

implement define rectangle and rectangle widget support

10/30/2012 09:41 AM - Greg Shah

Status:	Closed	Start date:	01/30/2014
Priority:	Normal	Due date:	
Assignee:	Marius Gligor	% Done:	100%
Category:		Estimated time:	60.00 hours
Target version:	GUI Support for a Complex ADM2 App	vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to User Interface - Feature #2425: implement GUI rectangle widget			Closed

History

#1 - 10/31/2012 02:54 PM - Greg Shah

- Target version set to Milestone 12

#2 - 01/29/2014 06:14 PM - Greg Shah

- Assignee set to Marius Gligor

- Start date set to 01/30/2014

- Estimated time changed from 32.00 to 60.00

For now this will only be implemented in ChUI mode.

Start by writing 4GL testcases to show the features of the rectangle widget. In particular:

- How do you use DEFINE RECTANGLE to define such widgets with different options, sizes and so forth.
- How does this render visually?
- How does this affect layout of other widgets?
- How does the resulting widget behave in regard to input? As far as I understand it is an output-only widget and doesn't have any event processing. Confirm this.

Once the behavior has been fully explored using testcases, document the findings here. Those findings essentially form the specification for the changes you will write to implement rectangle support.

For an idea of how we write such testcases, please review my recent testcase work in [testcases/uast/library_calls/](#). This is a bit different, but it shows a pretty good example of the kinds of things we do to fully understand a 4GL feature. Luckily, rectangle support will be much less work than the [#1634](#) that goes with the library_calls testcases.

Please note that I do expect this will take more time than normal, since it is your first deep work in the 4GL.

#3 - 04/03/2014 02:32 AM - Marius Gligor

- Status changed from New to WIP

#4 - 04/03/2014 06:27 PM - Greg Shah

Also I looked over P2J project code in order to understand how to implements P4 widgets on P2J and I found a class `com.goldencode.p2j.ui.RectangleWidget`. Seems that someone already started the implementation for RECTANGLE.

That is some stubbed out support on the server side. I think it is not complete. There is no client-side implementation yet.

Do you have a manual or a tutorial for developers which explains how to implements P4GL widgets on P2J?

Unfortunately, no. The best thing to do is write a simple testcase that uses a DEFINE BUTTON and uses that button in a frame. Then convert it and review the generated frame definition/business logic. Then follow the code in the runtime on both the server side and the client side for the button widget.

#5 - 04/04/2014 11:08 AM - Marius Gligor

Today I did a lot of RECTANGLE tests using the customer's P4GL environment. Also I converted a small procedure working with BUTTON and FRAME widgets. Bellow are the results of my researches after this tests. Please let me know if what I stated bellow are true or false.

1. RECTANGLE widget looks like BUTTON widget except:

- Rectangles are readonly widgets.
 - Rectangles does not accept user inputs except mouse events to move or resize.
ChUI implementation does not support mouse events so we can state that rectangles have no inputs.
 - Rectangles cannot gain the focus. We cannot navigate to rectangles using key navigation, no tabstop properties are defined for rectangles.
 - No action triggers could be associated with rectangles.
 - Optional rectangle could have a border (edge).
 - Rectangles can be used only on the FOREGROUND or BACKGROUND of a FRAME widget.

2. Server side implementation reside in package: `com.goldencode.p2j.ui`

Example:

```
@LegacyResource(resource = LegacyResource.BUTTON)
public class ButtonWidget extends ControlEntity
```

3. Client side implementation reside in package: `com.goldencode.p2j.ui.client`

Example:

```
public abstract class Button extends AbstractWidget
```

4. ChUI implementation reside in package: `com.goldencode.p2j.ui.chui`

Example:

```
public class ButtonImpl extends Button
```

5. The following classes are used in creation process of widgets:

- `com.goldencode.p2j.ui.client.ChuiWidgetFactory`

- `com.goldencode.p2j.ui.client.UiUtils`

Example from: `UiUtils.reconstructComponent(com.goldencode.p2j.ui.ComponentConfig)`

```
case ComponentConfig.BUTTON :  
    clazz = Button.class;  
    break;
```

6. `ComponentConfig` - is used to store widget attributes.

7. In order to generate Java code for RECTANGLE widgets we have to write some rules.

#6 - 04/07/2014 08:10 AM - Greg Shah

I think this is generally correct. Since we will be implementing the GUI as well, please do plan to support the mouse events.

I think it is possible to support mouse events on some ChUI terminals. We have not implemented that support yet and we don't plan to do it soon.

#7 - 04/07/2014 02:36 PM - Marius Gligor

- *File rectangle-widget.pdf added*

Here is the RECTANGLE widget description extracted from P4GL manuals and some remarks:

- Some attributes are used by RECTANGLE widget only most are common to other widgets as well.
- Some attributes are available only for Character interface others only for Graphical interface but most are available on both interfaces.
- Some attributes are available on all platforms others are platform specific like TOOL TIP and HWND which are available on Windows OS only.
- RECTANGLE widgets could be static created using DEFINE RECTANGLE or dynamic created using CREATE RECTANGLE

#8 - 04/08/2014 02:43 PM - Marius Gligor

- *File rectangle_uast_1.zip added*

Today I tried to convert some small RECTANGLE test cases using the current implementation for `RectangleWidget` and the current defined rules.

1. Some tests are converted without errors others not. For example the following statements are converted without errors:

```
define rectangle but size-char 20 by 6.  
    DISPLAY but.
```

However I saw that the converted code looks like a BUTTON conversion because I found a `setLabel` statement inside the converted code but `RECTANGLE` widgets has no label attribute.

```
public static class Rect0Frame0Def
    extends WidgetList
{
    RectangleWidget but = new RectangleWidget();

public void setup(CommonFrame frame)
{
    frame.setDown(1);
    but.setLabel("but");
}

{
    addWidget("but", "", but);
}
}
```

2. When I added a `NO-FILL` option the following statements are no longer converted.

```
DEFINE RECTANGLE poly SIZE-CHARS 30 BY 3 NO-FILL.
    DISPLAY poly.
```

I attached the results of my conversions.

Now I'm working to understand how conversion process works, rules which are applied and the structure of generated files. I think that finally I have to do some changes on the existing rules for `RECTANGLE` widgets.

#9 - 04/09/2014 01:01 PM - Marius Gligor

Doing more test cases conversions I think that I found a bug in conversion process. Let me explain:
Trying to convert procedures using BUTTON widgets I found:

1. Statements like: DEFINE BUTTON butt LABEL "Test". are converted without errors.
2. Adding other options for button like SIZE phrase DEFINE BUTTON butt LABEL "Test" SIZE 10 BY 1. are not converted.
An exception is throw (see StringHelper.sweep):

EXPRESSION EXECUTION ERROR:

```
-----  
tw.graft("set_widget_attribute", null, lastId,  
"attribute", setter,  
"instance", wname,  
"vartype", atype,  
"varvalue", aval)  
^ { No replacement value found for key: instance }
```

Doing some debugs I found that "wname" is null. Here is the stack trace:

```
Conversion [Java Application]  
com.goldencode.p2j.convert.ConversionDriver at localhost:57613  
Thread [main] (Suspended (breakpoint at line 427 in TemplateWorker$Template))  
TemplateWorker$Template.graftAt(String, Aast, Aast, int, Object[]) line: 427  
TemplateWorker$Template.graftAt(String, Aast, long, int, Object[]) line: 309  
TemplateWorker$Template.graft(String, Aast, long, Object[]) line: 249  
CE4564.execute() line: not available [local variables unavailable]  
Expression.execute() line: 323
```

Expression.execute() try to execute:

```
tw.graft("set_widget_attribute", null, lastId,  
"attribute", setter,  
"instance", wname,  
"vartype", atype,  
"varvalue", aval)
```

After the call to CE4564.execute() the "wname" is null which means "instance" = null

The rule for SIZE phrase is already defined in rules/convert/frame_generator.xml line 1583: <function name="process_button_options">

The problem is on line 1602: <action>wname = node.getAnnotation("javaname")</action> where node.getAnnotation returns a null value for "wname".
I fixed this bug by replacing the "node" with "butdef" as follow: <action>wname = butdef.getAnnotation("javaname")</action>

Having this fix in place all defined options (attributes) for BUTTON widget are converted without error.

I found the same bug for RECTANGLE widget as well in line 1711. I provided the same fix in line 1711 and the RECTANGLE widget attributes are now converted without error.

#10 - 04/14/2014 01:23 PM - Marius Gligor

- % Done changed from 0 to 80
- File mag_upd20140414a.zip added

Here is the RECTANGLE widget implementation for character interface and some remarks.

1. The implemented options for character interface which has been tested are presented bellow:

```
DEFINE RECTANGLE rectangle
[ NO-FILL ]
[ EDGE-CHARS width ]
[ GRAPHIC-EDGE ]
[ DCOLOR expression ]
[ PFCOLOR expression ]
[ { SIZE | SIZE-CHARS } width BY height ]
```

2. If GRAPHIC-EDGE is defined the border of the RECTANGLE could be a horizontal line, a vertical line, a box or no border depending on rectangle width and height.

3. On P4GL manual I found that EDGE-CHARS, SIZE or SIZE-CHARS should be represented as DECIMAL values. This means that we could specify something like: EDGE-CHARS 1.75 and the Progress will round up the value to the nearest integer. I tested on the Progress 4GL environment and I found that is true. I saw that on other widgets implementations like BUTTON integer values are used for SIZE and SIZE-CHARS. However in my RECTANGLE implementation I used double values and a round up method to convert the value to an integer. Is this OK or I should use integer values like on button implementation?

#11 - 04/14/2014 02:50 PM - Marius Gligor

- File mag_upd20140414b.zip added

I fixed a bug related to colours drawing on RECTANGLE widget according to my latest tests.

#12 - 04/15/2014 05:19 AM - Marius Gligor

- File mag_upd20140415a.zip added
- File rectangles.png added
- File rect-5.p added
- File chui_colors.p added

1. Add new options for DEFINE RECTANGLE like DCOLOR, PFCOLOR, BGCOLOR, FGColor, TOOLTIP. BGCOLOR, FGColor and TOOLTIP are for GUI only and ignored on ChUI.

```
DEFINE RECTANGLE rectangle
[ NO-FILL ]
[ { EDGE-CHARS width } | { EDGE-PIXELS width } ]
[ DCOLOR expression ]
[ BGCOLOR expression ]
[ FGColor expression ]
```

[GRAPHIC-EDGE]
[PFCOLOR expression]
[size-phrase]
[TOOLTIP tooltip]

2. DCOLOR and PFCOLOR can be static assigned on DEFINE RECTANGLE statement or assigned dynamic.
3. I fixed the color drawing on rectangles. DCOLOR is used as fill color and PFCOLOR is used for border (edge) color.

#13 - 04/15/2014 07:22 AM - Marius Gligor

- File *rect-6.png* added
- File *rect-6.p* added

An uast test having dynamic color assignments.

#14 - 04/16/2014 01:51 PM - Marius Gligor

- File *mag_upd20140416a.zip* added
- Status changed from WIP to Review

Final RECTANGLE implementation code.

#15 - 04/22/2014 11:20 AM - Greg Shah

Code Review 0416a

This is really good. Have you tested it on all 3 ChUI client types?

1. In *frame_generator.xml* function *process_rect_options* please change variable names *butdef* and *bchild* to something more rectangle-related (e.g. *rectdef* and *rchild*).
2. In *BaseConfig*, the *fgcolor* and *bgcolor* should probably be *Color* instances, not *int*.
3. In *BaseConfig*, why are the getters/setters for the new color attributes protected instead of public?
4. The *BaseEntity* copyright date needs update.
5. In *BaseEntity*, the *setBgcolor(int64 bgcolor)*, *getBgcolor()*, *setFgcolor(int64 fgcolor)* and *getFgcolor()* should be implemented instead of just throwing *UnsupportedOperationException*.
6. In *BaseEntity*, the *getTooltip()*, *getWidthPixels* and *getHeightPixels* methods are missing.

#16 - 04/22/2014 12:32 PM - Marius Gligor

Yes I tested on all 3 ChUI client types in Linux and in Windows OS.

#17 - 04/30/2014 12:09 PM - Marius Gligor

- File mag_upd20140430a.zip added

Fixed code review issues.

However FG_COLOR, BG_COLOR, TOOL_TIP and SIZE_PIXELS are attributes used on GUI interface only and the final implementation will be done when we'll implement the P2J GUI support.

#18 - 04/30/2014 12:16 PM - Greg Shah

Code Review 0430a

Everything looks good. The only open question is this:

3. In BaseConfig, why are the getters/setters for the new color attributes protected instead of public?

#19 - 04/30/2014 12:22 PM - Marius Gligor

- File mag_upd20140430b.zip added

Fixed.

#20 - 04/30/2014 12:54 PM - Greg Shah

Code Review 0430b

It looks good. Please get it regression tested (both conversion and runtime).

#21 - 05/06/2014 12:43 PM - Marius Gligor

I restarted the regression tests both CTRL-C part and MAIN part.

This time CTRL-C part took around 58 minutes.

Also are less failures compared to previous tests.

#22 - 05/06/2014 01:10 PM - Marius Gligor

- % Done changed from 80 to 100

0430b - Passed regression tests. Committed revision 10519.

#23 - 05/06/2014 01:36 PM - Greg Shah

- Status changed from Review to Closed

#24 - 11/16/2016 12:12 PM - Greg Shah

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

Files

rectangle-widget.pdf	207 KB	04/07/2014	Marius Gligor
rectangle_uast_1.zip	279 KB	04/08/2014	Marius Gligor
mag_upd20140414a.zip	47.9 KB	04/14/2014	Marius Gligor

mag_upd20140414b.zip	47.9 KB	04/14/2014	Marius Gligor
rectangles.png	13.2 KB	04/15/2014	Marius Gligor
mag_upd20140415a.zip	48.4 KB	04/15/2014	Marius Gligor
rect-5.p	489 Bytes	04/15/2014	Marius Gligor
chui_colors.p	704 Bytes	04/15/2014	Marius Gligor
rect-6.p	684 Bytes	04/15/2014	Marius Gligor
rect-6.png	13.2 KB	04/15/2014	Marius Gligor
mag_upd20140416a.zip	57.4 KB	04/16/2014	Marius Gligor
mag_upd20140430a.zip	57.6 KB	04/30/2014	Marius Gligor
mag_upd20140430b.zip	57.6 KB	04/30/2014	Marius Gligor