

## User Interface - Feature #2333

Feature # 2252 (Closed): implement GUI client support

### implement the GUI status line widget

07/09/2014 11:02 AM - Greg Shah

<b>Status:</b>	Closed	<b>Start date:</b>	07/09/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Marius Gligor	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	40.00 hours
<b>Target version:</b>	GUI Support for a Complex ADM2 App	<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			

### History

#### #1 - 09/11/2014 09:26 AM - Marius Gligor

- Status changed from New to WIP
- Assignee set to Marius Gligor

#### #2 - 09/15/2014 11:47 AM - Marius Gligor

- File status-bar.png added
- File mag\_upd20140915a.zip added

Here are the changes which I did so far. Some of the changes like color inheritance could be applied to other widgets as well.

1. I changed the values for BGCOLOR and FGCOLOR with integer types instead the ChUI Color instance.
2. Colors could be inherited by widgets from their parents. For example a the status bar inherit the background and foreground colors from his parent window if no custom colors are already explicit specified for status bar.
3. I added inside GuiOptions a global Inset object used when draw widgets. The Inset define the inside margins for a widget. In the status bar case I used insets to set the margins between status bar border and the status bar text.
4. Before the status bar text is about to draw a clip rectangle is defined which protect against drawing text outside the status area when text width is longer than status bar area on which the text is draw. Also when the parent window is resized the status bar width is also resized and the clip rectangle is recalculated.
5. I optimized the status bar draw code by eliminating any unnecessary objects creation inside the draw method.
6. On Windows OS when a window having a status bar is resizable a small triangle is draw on status bar at the bottom-right corner. I added a new graphical primitive to draw polygons in order to implements this feature. (see the attached picture)
7. I added also a flush graphical primitive which could be used whenever we need to force the graphical driver to copy the background paint buffer to screen.
8. In WindowLayout#doLayout I fixed a small bug related to Window resize. When a Window is resized we must also resize the ScreenBitmap size which keeps track of all the clipped regions. Otherwise it is possible to have widgets which are not draw at all! Here we should have a precise conversion between pixels and characters and vice versa.
9. Doing tests I observed that the drawing window looks different in Linux OS compared with Windows OS. In Windows looks better. In Linux not. Also when starting a GUI client it seems that many instances of P2J client are created!!!

### #3 - 09/15/2014 03:17 PM - Greg Shah

Code Review 0915a

The changes look very good.

1. Are the GuiOptions insets only useful for the status line? If so, then it should be named something status-line specific.
2. For fillPolygon(), are there cases where the number of points (nPoints) is different than the length of the two arrays (xPoints and yPoints)?

Doing tests I observed that the drawing window looks different in Linux OS compared with Windows OS. In Windows looks better. In Linux not.

Also when starting a GUI client it seems that many instances of P2J client are created!!!

Please do fix these problems.

### #4 - 09/16/2014 10:20 AM - Marius Gligor

- File mag\_upd20140916a.zip added

1. Insets could be used by other widgets as well like message area, window title bar. Even window widgets could use insets and we could eliminate the global GuiOption.BORDER\_SIZE setting. I added an insets property to AbstractWidget class the ancestor of all widgets because each widgets should have different insets.
2. For fillPolygon graphical primitive I used the same signature like Graphics2D.fillPolygon method which require 3 parameters. However I simplified our method signature to use the size of coordinates as a third parameter when call Graphics2D.fillPolygon.
3. The drawing differences between Windows and Linux platforms comes from GUI coordinate units conversions. For example in our GUI driver when the height of the status bar widget is calculated the computation is done in pixels. Then the height is converted from pixels units in character units which in current implementation are also integer values. When the widget is draw the height in character units is converted back to pixels units. Because the character units are integer values the conversion does not produce the same value in pixels as was initial calculated. For example the status bar height in pixels is equivalent to 1.4 character units but since the character units are integers the stored value is 1. Even more the height in pixels depends on font height used to draw status text and should be different in Window and in Linux for the same font. As a result the status bar could be 1.6 characters in Windows but only 1.3 characters in Linux. When 1 character is converted back in pixels the height is less than initial value and the height is too small. Using 2 characters as height for status bar is too high.

In conclusions all character units MUST be of type decimal and the conversion between pixels and character units and vice versa should be precise as much as possible.

This is one of the most important part of GUI driver implementation and as I understood is already in work and will be available soon.

**#5 - 09/16/2014 10:50 AM - Greg Shah**

Code Review 0916a

The changes look good. Please go ahead into runtime regression testing.

In conclusions all character units MUST be of type decimal and the conversion between pixels and character units and vice versa should be precise as much as possible.

This is one of the most important part of GUI driver implementation and as I understood is already in work and will be available soon.

Correct.

Also when starting a GUI client it seems that many instances of P2J client are created!!!

What have you found about this issue?

**#6 - 09/16/2014 11:03 AM - Marius Gligor**

When GUI client starts after authentication the Ask.p is tried to be executed on server side but crash in StandardServer.standardEntry line 364 having stopDisp = 1 which forces a logoff and then re-logon (continued operation). The GUI client remain inside ClientCore.start loop because running is always false.

**#7 - 09/16/2014 11:05 AM - Greg Shah**

What causes the server-side crash?

**#8 - 09/16/2014 11:09 AM - Marius Gligor**

I found the following error: Attempt to instantiate classclass com.goldencode.p2j.ui.client.Frame which has no proper mapping

**#9 - 09/16/2014 11:10 AM - Greg Shah**

Please post the full stack trace.

**#10 - 09/16/2014 11:12 AM - Marius Gligor**

```
[09/16/2014 18:11:26 EEST] (StandardServer.invoke:SEVERE) {00000005:00000016:bogus} Abnormal end!  
java.lang.reflect.InvocationTargetException  
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)  
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)  
at java.lang.reflect.Method.invoke(Method.java:601)  
at com.goldencode.p2j.util.Utils.invoke(Utils.java:1238)  
at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)  
at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)  
at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)  
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

```
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:601)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:694)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
at java.lang.Thread.run(Thread.java:722)
Caused by: java.lang.RuntimeException: Attempt to instantiate classclass com.goldencode.p2j.ui.client.Frame which has no proper mapping
at com.goldencode.p2j.ui.client.driver.WidgetFactoryAdapter.create(WidgetFactoryAdapter.java:64)
at com.goldencode.p2j.ui.client.UiUtils.reconstructComponent(UiUtils.java:360)
at com.goldencode.p2j.ui.client.WidgetRegistry.reconstructWidget(WidgetRegistry.java:82)
at com.goldencode.p2j.ui.client.WidgetRegistry.pushDefinition(WidgetRegistry.java:191)
at com.goldencode.p2j.ui.chui.ThinClient$18.run(ThinClient.java:7007)
at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:12050)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:11999)
at com.goldencode.p2j.ui.chui.ThinClient.pushOneDef(ThinClient.java:6995)
at com.goldencode.p2j.ui.chui.ThinClient.pushScreenDefinition(ThinClient.java:6970)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:601)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:694)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:257)
at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1128)
at com.goldencode.p2j.net.Queue.transact(Queue.java:585)
at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:223)
at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:163)
at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1424)
at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:97)
at com.sun.proxy.$Proxy4.standardEntry(Unknown Source)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:285)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:94)
at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:227)
at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:423)
at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:1)
at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:290)
```

**#11 - 09/16/2014 11:15 AM - Greg Shah**

OK, the failure is actually on the client side and it is just that our GUI support does not yet handle frames. So when the frame definition is pushed down and instantiated, the client aborts.

This is expected considering the current state of the GUI. No more work needs to be done there.

**#12 - 09/17/2014 10:16 AM - Marius Gligor**

- % Done changed from 0 to 100

0916a passed regression tests. Committed revision 10613.

**#13 - 09/17/2014 10:17 AM - Marius Gligor**

- Status changed from WIP to Review

**#14 - 09/17/2014 10:35 AM - Greg Shah**

I am keeping this open only because we still have the drawing differences between Windows and Linux. When the [#2323](#) changes are released, please do fix the drawing as part of this task.

**#15 - 10/08/2014 09:22 AM - Greg Shah**

- Status changed from Review to Closed

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

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

**Files**

---

mag_upd20140915a.zip	63.8 KB	09/15/2014	Marius Gligor
status-bar.png	7.09 KB	09/15/2014	Marius Gligor
mag_upd20140916a.zip	70.8 KB	09/16/2014	Marius Gligor