User Interface - Feature #2286

Feature # 2252 (Closed): implement GUI client support

design and implement basic GUI window widget support

04/16/2014 02:23 PM - Greg Shah

Status:	Closed	**Start date:**	04/16/2014
Priority:	Normal	**Due date:**	
Assignee:	Marius Gligor	**% Done:**	100%
Category:		**Estimated time:**	120.00 hours
Target version:	GUI Support for a Complex ADM2 App		
billable:	No	**vendor_id:**	GCD

Description

History

#1 - 04/16/2014 02:24 PM - Greg Shah

Put all details in the parent task #2252.

#2 - 04/17/2014 03:47 AM - Marius Gligor

- Status changed from New to WIP

#3 - 05/06/2014 08:53 AM - Marius Gligor

Before starting to implements the GUI model we have to describe and understand the ChUI model implementation.
Here is what I found looking inside the ChUI implementation.

A - On server side:
- StandardServer exports ServerExports services implemented in class LogicalTerminal

B - On client side:

- The core is the ThinClient which exports ClientExports services.

- ChUI widgets are pushed down from server to client and stored inside a WidgetRegistry structure which is a member of the ThinClient instance.

- The WidgetRegistry structure has a map collection of widgets. An entry in the map is of type <Integer, Widget>
Each entry in the map represent a widget. Each widget has an unique Integer ID (map entry key) and it is stored as an instance of Widget interface (map entry value).

- An OutputManager instance is used as an interface to low level UI primitives via an ScreenDriver interface.

- ScreenDriver instances drive an OutputPrimitives interface which provide primitive drawing services.

Example for ChUI:
ChuiScreenDriver implements ScreenDriver
SwingChuiDriver extends ChuiScreenDriver

ChuiWebDriver extends ChuiScreenDriver implements WebScreenDriver

- OutputPrimitives define the low level UI primitives methods.

Example for ChUI:
BasePrimitives implements OutputPrimitives
DriverPrimitives extends BasePrimitives
ChuiPrimitives extends DriverPrimitives

Each widget implements the Widget interface. The Widget interface define a draw() method which is called to draw the widget.
The draw method use an OutputManager instance to draw to the screen.
The current implementation for ChUI widgets provides simple drawing primitives character oriented.
The draw method for a widget is called whenever the widget has to be draw.

#4 - 05/06/2014 08:58 AM - Marius Gligor

I have some questions related to this issue:

1. Should we follow the same model for GUI like the ChUI model (see note #3) and define specific draw methods using GUI primitives on each widgets?

2. For WINDOW widget the size is always calculated based on character size (font metrics)?

#5 - 05/06/2014 09:08 AM - Greg Shah

Please put these details into the parent task. I will respond there.

#6 - 05/08/2014 11:17 AM - Marius Gligor

- File window_attributes.txt added

I attached a list with all WINDOW widget attributes more than 70!
I started to do tests using some WINDOW attributes on the customer's windev01 server.
Most attributes cannot be used on DEFAULT-WINDOW.

#7 - 05/14/2014 09:33 AM - Marius Gligor

- File mag_upd20140514a.zip added

I added a new layer of inheritance PaneEntity and PaneConfig which holds shared options for DIALOG, FRAME and WINDOW widgets.
The GUI driver classes were moved on com/goldencode/p2j/ui/client/gui/ package.

Some observations:

1. setParent and getParent are implemented in BaseEntity class but setParentHandle and getParentHandle are implemented in the super class GenericWidget.
Should We move the implementation for setParentHandle and getParentHandle to BaseEntity?

2. I found inside rules different method names for the same option. For example I found setFgColor and setFgcolor definitions which basically are for the same attribute.
Should we change the definitions inside the rules in order to have the same method name?

3. In P4G options calculated in CHARS units are of type DECIMAL. This kind of options should be kept as double value inside the configuration files.
On character interfaces this values are rounded to integers but in graphical interfaces I think this should be transformed in pixels at one moment that's why we have to keep the original values which are decimals.
I found integer values for such kind of options inside the existing configuration files.
Even more I saw in the source code statements like: (int) double as a round substitute.
This is not OK because (int) double does not round the returned integer value it returns just the integer part.
Example for a value of 0.75 it returns 0 instead 1.

#8 - 09/04/2014 10:31 AM - Greg Shah

- Status changed from WIP to Closed

- % Done changed from 0 to 100

#9 - 11/16/2016 12:12 PM - Greg Shah

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

Files

window_attributes.txt	7.37 KB	05/08/2014	Marius Gligor
mag_upd20140514a.zip	262 KB	05/14/2014	Marius Gligor