

# MANGO: AN ONLINE GUI DEVELOPMENT TOOL FOR THE TANGO CONTROL SYSTEM\*

G. Strangolino, C. Scafuri, Sincrotrone Trieste, Trieste, Italy

## Abstract

Mango is an online tool based on *QTango* [1] that allows easy development of graphical panels ready to run without need to be compiled. Developing with Mango is easy and fast because widgets are dragged from a widget catalogue and dropped into the Mango container. Widgets are then connected to the control system variables by choosing them from a *Tango* [2] device list or by dragging them from any other running application built with the *QTango* library. Mango has also been successfully used during the FERMI@Elettra commissioning both by machine physicists and technicians.

## WHAT MANGO IS

Mango is a simple tool to generate online *QTango* panels, that are ready to run without need to be compiled. In this sense, Mango is an interpreter. The interfaces generated with Mango are made up of the simple *QTango* and *Qt4* [3] widgets, and all the logic of the components and their interaction with each other relies on the intrinsic logic of the widgets themselves.

### Whom Mango is Addressed to

- The *Tango* device server programmer, who wants to design a *Tango* device fully compatible with the *QTango* library and wants to immediately test his work in progress. He might even outline a GUI choosing the most suitable widgets to represent the *Tango* attributes and commands that are modelled by the device server. The sketch can be saved and submitted to the attention of the GUI developer, who can complete it, improve some human computer interaction aspects and even import the sketch into the *Qt4 designer* to create an executable.
- The GUI developer who wants to rapidly create an interface to a *Tango* device server whose commands and attributes fully support the *QTango* widgets, without needing any additional logic.
- The *end user* of the graphical interface, who can outline a draw of the desired panel and discuss it with GUI developer.
- The control room operator. He might use Mango to create *on the fly* panels, in order to create a summarizing graphical interface gathering *QTango* objects from other real *QTango* executable applications running on the desktop. Actually, the powerful *drag and drop* system exported by the *QTango* library allows dragging a *QTango* widget from an application and dropping it into any Mango container.

- The *QTango* library tester: Mango widgets interacting with the *Tango* framework are *derived* from the corresponding *QTango* elements and represent valid test objects, both from the stability point of view and from the good design of the elements (the better the widgets are designed, the better possibilities are offered to the Mango tool to manage interoperating *QTango* elements).
- The hardware referent, who commissions the requests to the GUI developer. He may draw a draft of the application containing the relevant controls and widgets and discuss every aspect with the application programmer.

## WHAT MANGO IS NOT

Mango is not a tool to design complex control panels, with any other logic than the simple one offered by each *QTango* and *Qt4* widget through the exported *Qt properties* and the *signal/slot* connections. Nevertheless, GUI generation is easy and the work done with Mango is scalable and in no case is wasted: future improvements of the interface and the introduction of complex logic can be started from the *xml* generated with Mango by importing it into a *Qt4 designer* project.

## GUI COMPONENTS

The Mango graphical user interface is made up of the following components (see Fig.1).

- An *Object Factory*: a list of *QTango* and *Qt* widgets ready to be used in the *main Mango widget area* by dragging and dropping them from the *Object Factory* widget.
- A *Tango DB Browser*: a tree representing the *Tango* devices exported to the *Tango* database. Attributes and commands of each exported device can be dragged from the *Tango DB Browser* and dropped into the *QTango* widgets populating the *main Mango widget area* component.
- The *main Mango widget area*: a container where *Qt* and *QTango* widgets can be dropped either from the *Object Factory* component or from other *QTango* panels on the desktop. Once the widgets are dropped into this area, they can be *selected, moved and resized* with the mouse, and their properties can be personalized through the *Object Properties* component on the left and by right clicking on them. Multiple selection is possible by pressing the *Control* key while clicking with the mouse.

\* This work was supported in part by the Italian Ministry of University and Research under grants FIRB-RBAP045JF2 and FIRB-RBAP06AWK3.

Currently, setting widget properties on multiple selections is not supported, but a bunch of actions is available by right clicking over a widget belonging to the multiple selection: move, resize, alignment to grid... Main Mango widget resize events are propagated to children, that resize proportionally, unless the option *Ignore main window resizes* is checked on the *Design Options* component. The *grid* drawn over each Mango container serves as a maths squared exercise book to ease placing the collection of widgets into the main Mango widget area.

- The *Tango Point Information* component: *QTango* widgets use this object to display useful information about their state.
- The *Object Properties* component: allows to click on a property and change its value by either directly editing the property on the tree widget item or by clicking on the custom button that might appear according to the property type. Custom editing dialogues are provided for complex properties like fonts and colours.

### DESIGNING A CONTROL PANEL

Composing a graphical control panel with Mango is as easy as dragging an object from the *Object Factory*

component and dropping it into the *main Mango widget area*. Once a widget becomes part of the *main Mango widget area*, it can be configured through its properties. For instance, a *Tango source or target* can be associated to a *QTango* widget. To change object properties, one has to *first select* the object with the left button of the mouse. The object is highlighted with a light blue colour and its cyan border is sensible to mouse resize actions. Multiple selection is possible through the *control* key pressed in conjunction with the mouse button. A set of actions is available for multiple selections by right clicking with the mouse over one of the widgets.

### Simple Widgets and Containers

Mango widgets can be divided into *simple widgets* and *containers*. Containers are special widgets that can contain both simple widgets and other containers. They propagate the resize events to their children and manage saving their properties. One can populate the interface both with simple widgets and containers without worrying too much about what kind of object she is using. As one can see in Fig.1, the *readers* are connected to *Tango* devices.

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)



Figure 1: Mango graphical user interface.

In Fig.1 the *tab widget* is currently *selected* and its properties are accessible through the *Object Properties* widget. The *Tango Point Information* component displays the status of the underlying Tango connection that provides data to the plot. *Writers* are partially enabled: they cannot *write* the tango quantity they are connected to when in *design* mode.

### EXECUTING A CONTROL PANEL

A Mango project saved as a profile can be loaded in two different ways:

- via the command line;
- launching the Mango application normally, then switching off *design mode* via the *View* menu and loading a profile by triggering the action *Profile Manager...* from the *File* menu.

In the first way you can integrate the Mango panel into a launcher icon or into an application browser. Fig.2 represents the same panel as in Fig.1 in execution mode.

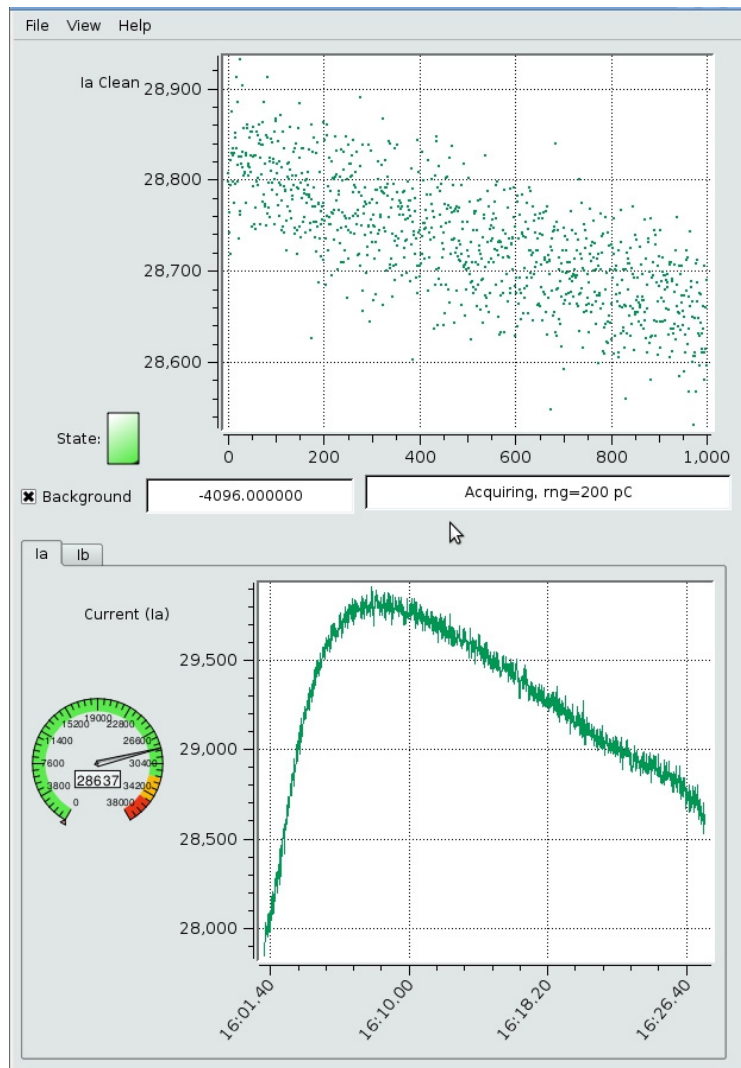


Figure 2: A Mango control panel in execution mode, currently used in Fermi@Elettra control room.

### PROFILES

The graphical interfaces realized with Mango can be easily managed via the Mango *profiles*. A profile stores the configuration of a GUI designed with Mango.

### CONCLUSIONS

Mango is currently used by GUI and Tango device server programmers to sketch graphical user interfaces and testing purposes. Up to now, Elettra and Fermi@Elettra control room operators have designed and run with Mango about ten control panels.

### REFERENCES

- [1] G.Strangolino et al., “QTango: a library for easy tango based GUIs development”, ICALEPCS2009, Kobe, October 2009, THP096.
- [2] *Tango*, <http://www.tango-controls.org/>
- [3] *Qt for Application Development*, <http://qt.nokia.com/>