

## Lab 8: Introduction to Visual Ruby

### Learning Objectives:

1. Introduction to Visual Ruby
2. Gtk::Builder
3. Module: GladeGUI
4. References

### Supporting files:

1. Glade\_Examples.zip

---

## 1. Introduction to Visual Ruby

Visualruby was designed for ruby programmers who want to add a GUI to their ruby scripts. VR simplifies the process of adding GTK+ windows to your applications. Visualruby is integrated with the glade interface designer, such that only a click on a ruby file is necessary to edit its widgets using glade. The editor has a "Run" button that allows you to execute your script. In case of an error, the editor automatically jumps to the spot in your code that caused the error.

The GTK is a library for creating graphical user interfaces. The library is created in C programming language. The GTK library is also called the GIMP Toolkit. Originally, the library was created while developing the GIMP image manipulation program. Since then, the GTK became one of the most popular toolkits under Linux and BSD Unix. Today, most of the GUI software in the open source world is created in Qt or in GTK. Language bindings exist for C++, Ruby, Python, Perl, Java, C#, JavaScript, PHP and other programming languages.

Glade Interface Designer is a graphical user interface builder for GTK+, with additional components for GNOME. The user interfaces designed in Glade are saved as XML. By using the GtkBuilder GTK+ object, these can be loaded by applications dynamically as needed. In its third version, Glade is programming language-independent, and does not produce code for events, but rather the XML file is then used with an appropriate binding.

To try Visualruby, try running VR on terminal and click on "Open Project" and play around with all the examples. Those examples cover vast majority of everything you can do with VR. You can also download the visualruby project itself and try executing the code.

---

## 2. Gtk::Builder

GtkBuilder is the XML format that the Glade Interface Designer uses to save its forms. These documents can then be used in conjunction with the GtkBuilder object to instantiate the form using GTK+. GladeXML is the XML format that was used with conjunction with libglade, which is now deprecated

Glade Interface Designer automatically generates all the source code for a graphical control element. The "Gtk.Builder class" allows user interfaces to be designed without writing code. The class describes the interface in an Extensible Markup Language (XML) file and then loads the XML description at runtime and creating the objects automatically. The Glade Interface Designer allows creation of the user interface in a manner, such that the description of the user interface is independent from the programming language being used.

More information on GtkBuilder can be found [here](#).

An example of creation of GUI with Glade and Gtk::Builder is as follows:

```
require 'gtk2'

builder = Gtk::Builder.new
builder.add_from_file(file)
builder.connect_signals { |handler| method(handler) }
```

The first line in the above code creates the Builder object, which is responsible of creating the Glib::Objects from the xml definition and also stores them for later use (you can call `get_object(<object name>)` on builder, it will return the widget defined with object name).

The second line loads your interface definition, where file is the path to your gtkbuilder file.

The third line connects signals calls the block provided, once for every signal you have defined in your interface. The 'handler' is just a string defining the name of the signal.

---

### 3. Module: GladeGUI

GladeGUI connects your class to a glade form. It will load a .glade file into memory, enabling your ruby programs to have a GUI interface. GladeGUI works by adding an instance variable named "builder" to your class. The "builder" variable is an instance of Gtk::Builder It holds references to all your windows and widgets. While it is running, a thread runs the code in the block associated with the call to Thread.new and then it stops running.

To use the GladeGUI interface, include this line in your code:

```
class MyClass
  include GladeGUI
end
```

GladeGUI will load a corresponding glade file for this class. It knows which glade file to load by using a naming convention:

```
/folder/MyClass.rb
```

Will load this glade file:

```
/folder/glade/MyClass.glade
```

The class, the class file name and the glade file name must match. You should always name your class, script, and glade file the same name (case sensitive).

An example of using GladeGUI is as follows:

```
class MyClass

  include GladeGUI
  def initialize()
    puts @builder.to_s # => nil
  end
end
```

```
def before_show()  
  puts @builder.to_s # => Gtk::Builder  
end  
  
end
```

GladeGUI adds an instance variable, `builder` to your class. It loads all the windows and widgets from your glade file into `@builder`. So, you use `@builder` to manipulate everything in your class's GUI. `@builder` is set when you call the `show_glade()` method, as the above code shows.

---

#### 4. References:

- [1] <http://visualruby.net/GladeGUI.html>
  - [2] <http://visualruby.net/>
  - [3] [https://en.wikipedia.org/wiki/Glade\\_Interface\\_Designer](https://en.wikipedia.org/wiki/Glade_Interface_Designer)
  - [4] <https://stackoverflow.com/questions/1017908/ruby-glade-gtkbuilder-example>
  - [5] <https://glade.gnome.org/>
-