

GTK+/GNOME applications with gtkmm

Dodji Seketeli

<http://www.gtkmm.org>

Agenda

- History
- gtkmm at a glance through examples
- Advantages of gtkmm
- gtkmm specifics
- What's new

History

- Started in 1998 (gtk--)
 - by Tero Pulkkinen and Havoc Pennington
 - Karl Nelson became the maintainer
- Major rewrite for gtkmm 2.0 (October 2002)
 - follows gtk+ 2.0 rewrite
 - uses defs file format from pygtk
 - a C++ friendlier face

Part of the official GNOME bindings suite

- all major pieces of GTK+/GNOME are wrapped
 - cairomm, glibmm, gdkmm, gtkmm, libglademmm etc ...
- complies with API/ABI rules
- follows the developer platform numbering scheme and release schedule

Example: Hey Hey world !

```
#include <gtkmm.h>

using namespace Glib ;
using namespace Gtk ;

int
main (int argc, char **argv)
{
    Main init (argc, argv) ;

    Window window;

    Label label ("Hey hey world !") ;

    window.add (label) ;

    window.show_all () ;

    Main::run (window) ;

    return 0 ;
}
```

Hey hey world in GTK+

```
#include <gtk/gtk.h>

void
on_delete_event_cb ()
{
    gtk_main_quit ();
}

int
main (int argc, char **argv)
{
    GtkWidget *window, *label ;

    gtk_init (&argc, &argv) ;

    window = gtk_window_new (GTK_WINDOW_TOPLEVEL) ;

    label = gtk_label_new ("Hey hey world !") ;

    gtk_container_add (GTK_CONTAINER (window), label) ;

    g_signal_connect (G_OBJECT (window), "delete-event", G_CALLBACK (on_delete_event_cb), NULL) ;

    gtk_widget_show_all (window);

    gtk_main () ;

    return 0 ;
}
```

Typesafe signals example

```
#include <gtkmm.h>

using namespace Glib ;
using namespace Gtk ;
using namespace sigc ;

void
on_button_clicked_signal ()
{
    MessageDialog dialog ("Woot ! you clicked !");
    dialog.run () ;
}

int
main (int argc, char **argv)
{
    Main init (argc, argv) ;

    Button my_button ("Click me!") ;

    Window my_window ;

    my_window.add (my_button) ;

    my_button.signal_clicked ().connect (&on_button_clicked_signal) ;

    my_window.show_all () ;

    Main::run (my_window) ;
}
```

Advantages of gtkmm (compared to GTK+)

- Type-safe: find more errors at compile time
- Clearer API
- Inheritance without macro soup and boiler plate
- Simpler memory management
- Less code
- Can be mixed with GObject code

Mixing with GObject/C

```
void
show_message ()
{
    MessageDialog dialog ("Wooot ! you clicked !");
    dialog.run ();
}

void
on_button_clicked_signal ()
{
    show_message ();
}

int
main (int argc, char **argv)
{
    Main init (argc, argv);
    Button my_button ("Click me!");

    Window my_window ;
    my_window.add (my_button);

    GtkButton *b = my_button.gobj ();

    g_signal_connect (G_OBJECT (b), "clicked", G_CALLBACK (on_button_clicked_signal), NULL);
    /*my_button.signal_clicked ().connect (&on_button_clicked) ;*/

    my_window.show_all ();
    Main::run (my_window);
}
```

gtkmm-specific techniques

- **Glib::ustring:** Like `std::string`, but UTF8
- **Glib::RefPtr<>:** smart pointer, like `boost::intrusive_ptr`
- **Signals**
`m_button.signal_clicked().connect(sigc::(on_button_clicked));`
- **Integrates with STL and BOOST**

Example: Using smart pointers

```
#include <gtkmm.h>
#include <boost/shared_ptr.hpp>

using namespace Glib ;
using namespace Gtk ;
using namespace sigc ;

/*for those who don't want to type*/
typedef boost::shared_ptr<Window> WindowPtr ;

WindowPtr
create_window (const ustring &a_title)
{
    // We can use whatever memory management we like:
    // For instance, we cqn use boost::shared_ptr<> :
    WindowPtr result (new Window) ;
    result->set_title (a_title) ;
    return result ;
}

int
main (int argc, char **argv)
{
    Main init (argc, argv) ;
    WindowPtr win = create_window ("Hooza, I won't leak this window !") ;
    win->show_all () ;
    Gtk::Main::run (*win) ;
    return 0 ;
}
```

using libglademmm

Design, edit, your UI visually, with Glade.

Implement functionality in code, with C++.

```
using namespace Glib;
```

```
using namespace Gtk ;
```

```
using Gnome::Glade::Xml ;
```

```
RefPtr<Xml> glade_xml = Xml::create ("basic.glade");
```

```
Dialog *dialog = 0;
```

```
refXml->get_widget("DialogBasic", dialog);
```

What's new

- gtkmm 2.10 stable.
 - Cairomm integration, Printing, RecentFiles
 - update a lots of widgets methods here and there
- Even more documentation.
- Here to stay

More information

- <http://www.gtkmm.org>
- Helpful people on gtkmm-list@gnome.org
- Many examples and demos, including C++ versions of all of the C examples and demos.
- Details worked out on [#c++](http://irc.gnome.org)