

Multiplatformní GUI toolkity

GTK+ a Qt

Jan Outrata



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLMOUCI



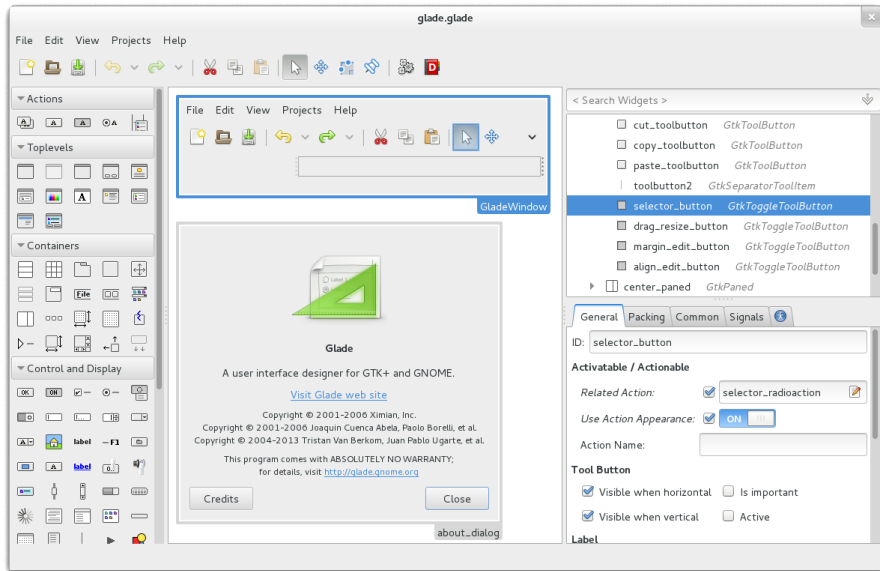
- = programová knihovna (nebo kolekce knihoven) implementující prvky GUI = **widgety** (tlačítka, seznamy, menu, posuvník, bary, dialog, okno atd.) a umožňující tvorbu GUI (grafického uživatelského rozhraní) aplikace
- vlastní jednotný nebo nativní (pro platformu/systém) vzhled widgetů, možnost stylování
- nízkourovňové (Xt a Xlib v X Windows System a libwayland ve Waylandu na unixových systémech, GDI Windows API, Quartz a Carbon v Apple Mac OS) a vysokoúrovňové (MFC, WTL, WPF a Windows Forms v MS Windows, Cocoa v Apple Mac OS X, Motif/Lesstif, Xaw a XForms na unixových systémech)
- multiplatformní = pro více platforem (MS Windows, GNU/Linux, Apple Mac OS X, mobilní) nebo platformově nezávislé (Java) – aplikace může být také
- (většinou) **událostmi řízené programování (event-driven programming)** – toolkit v hlavní smyčce zachytává události (uživatelské od myši nebo klávesnice, od časovače, systému, aplikace samotné atd.) a umožňuje implementaci vlastních obsluh (even handler, callback function), objektově orientované programování (objekty = widgety aj.) – nevyžaduje OO programovací jazyk!



- **language binding** = API (aplikační programové rozhraní) toolkitu v jiném prog. jazyce než původní API a toolkit samotný
- **GUI designer/builder** = WYSIWYG nástroj pro tvorbu GUI s využitím toolkitu, hierarchicky skládáním prvků, z uloženého XML pak generuje kód nebo GUI vytvoří za běhu aplikace
- nekomerční (GNU (L)GPL, MIT, open source) i komerční licence
- např. GTK+ (C), Qt (C++), wxWidgets (C++), FLTK (C++), CEGUI (C++), Swing/JFC (Java), SWT (Java), JavaFX (Java), Tcl/Tk (Tcl), XUL (XML) aj. (http://en.wikipedia.org/wiki/List_of_widget_toolkits)



- primárně pro X Window System a Wayland na unixových systémech (vč. GNU/Linuxu), portovaný na MS Windows, Apple Mac OS X a mobilní a embedded platformy (Openmoko)
- v **C**, language bindings pro C++ (gtkmm), C#/.NET (Gtk#, projekt Mono, jen verze 2), Javu (java-gnome), Python, Ruby, PHP (jen verze 2) a další
- části GTK+ (widgety), Pango (text), GDK (wrapper nad nízkoúrovňovými toolkity), ATK (přístupnost), GIO (souborové operace), využívá knihovny Glib (datové struktury, smyčka událostí, vlákna, objektový systém apod.) a Cairo (2D grafika)
- GUI designery **Glade** – i součást vývojového prostředí (IDE) Anjuta, **Stetic** – pro Gtk#, součást IDE MonoDevelop
- licence GNU LGPL, **free software**, součást projektů GNU a GNOME
- použití v linuxových desktopových prostředích **GNOME**, Unity, Xfce, LXDE, ROX aj. a mnoha aplikacích (GNOME, GIMP, Evolution, GNU Emacs, Pidgin, Wireshark aj.)





helloworld.c (1):

```
#include <gtk/gtk.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>
```

```
gchar *hellostr;
```

```
void window_destroy(GtkWidget *widget, gpointer data)
{
    gtk_main_quit ();
}
```

```
void button_clicked(GtkWidget *widget, gpointer data)
{
    g_printf ("%s\n", hellostr);
}
```



helloworld.c (2):

```
GtkWidget *helloworld_new ()
{
    GtkWidget *window, *button;

    hellostr = g_locale_to_utf8 (_ ("Hello World"), -1, NULL, NULL, NULL);

    window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
    g_signal_connect (G_OBJECT (window), "destroy",
                     G_CALLBACK (window_destroy), NULL);

    button = gtk_button_new_with_label (hellostr);
    g_signal_connect (G_OBJECT (button), "clicked",
                     G_CALLBACK (button_clicked), NULL);
    g_signal_connect_swapped (G_OBJECT (button), "clicked",
                              G_CALLBACK (window_destroy), G_OBJECT (window));
    gtk_container_add (GTK_CONTAINER (window), button);
    gtk_widget_show (button);

    return window;
}
```



helloworld.c (3):

```
int main(int argc, char *argv[])
{
    GtkWidget *helloworld;

    gtk_init (&argc, &argv);

    helloworld = helloworld_new();
    gtk_widget_show (helloworld);

    gtk_main ();

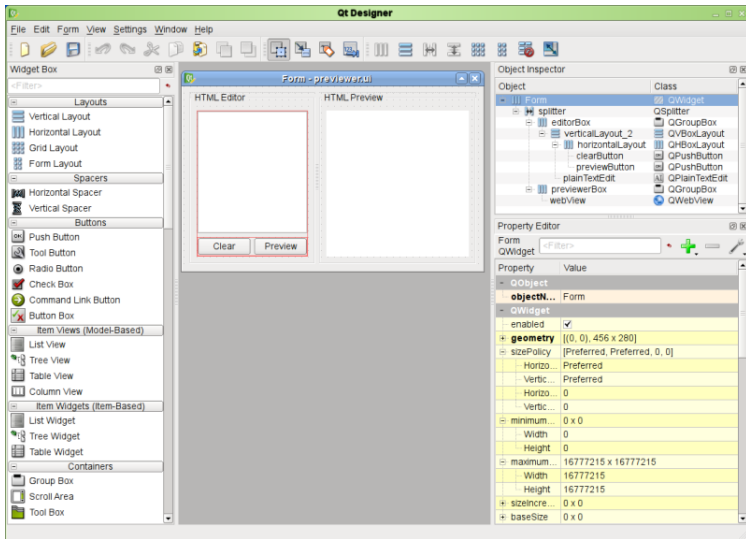
    return 0;
}
```




- 1 v rámci IDE (Anjuta)
- 2 „ručně“ překladačem C s pomocí nástrojů pro cesty k hlavičkovým souborům a knihovnám:

```
pkg-config --cflags --libs gtk+-3.0
```

- pro unixové systémy (X Window System a Wayland, vč. GNU/Linuxu), MS Windows, Apple Mac OS X a mobilní a embedded platformy (Android, iOS, Windows Phone, Embedded Linux)
- v **C++** (rozšířeném o komunikaci mezi objekty, pro události a jejich služby, překladač **Metaobject compiler, moc**, do C++), language bindings pro C#/.NET (QtSharp, Qyoto pro verzi 4), Javu (Qt Jambi, jen verze 4), Python (PyQt, PythonQt, PySide), Ruby (QtRuby, jen verze 4), PHP (PHP-Qt, jen verze 4), Common Lisp (Common Qt, jen verze 4) a další
- části essential (Core, GUI, Widgets, Network, Multimedia, SQL aj.) a add-on (OpenGL, Script, WebKit, XML, Bluetooth aj.)
- GUI designer **Qt Designer** – i součást vývojového prostředí (IDE) Qt Creator
- licence GNU LGPL/GPL (edice Open source) a Qt Commercial (10 dní vyzkoušení), **firemní** (Trolltech, Nokia, Digia/The Qt Company) a open source (Qt Project) vývoj
- použití v linuxovém desktopovém prostředí **KDE** a mnoha aplikacích (KDE, Google Earth, Skype, VirtualBox, VLC, Autodesk, Mathematica aj.)



Qt: Hello World



main.cc:

```
#include "helloworld.h"
#include <qapplication.h>

int main (int argc, char *argv[])
{
    QApplication app(argc, argv);

    HelloWorld helloworld;

    //app.setMainWidget (&helloworld);
    helloworld.show();
    app.exec();

    return 0;
}
```

Qt: Hello World



helloworld.h:

```
#ifndef HELLOWORLD_H
#define HELLOWORLD_H

#include <qpushbutton.h>
#include <qmainwindow.h>

class HelloWorld : public QMainWindow
{
    Q_OBJECT
public:
    HelloWorld();
    virtual ~HelloWorld();
protected:
    QString hellostr;
    QPushButton button;
protected slots:
    virtual void button_clicked();
};

#endif // HELLOWORLD_H
```

helloworld.cc (1):

```
#include "helloworld.h"  
#include <qlayout.h>  
#include <stdio.h>
```

```
HelloWorld::HelloWorld()  
    : hellostr (tr ("Hello_ World")),  
      button(hellostr, this)  
{  
    QObject::connect (&button, SIGNAL (clicked()), this,  
                      SLOT (button_clicked()));  
    QObject::connect (&button, SIGNAL (clicked()), this, SLOT (close()));  
  
    button.setSizePolicy (QSizePolicy::Minimum, QSizePolicy::Minimum);  
    button.setMinimumSize (button.size());  
    setCentralWidget (&button);  
    adjustSize();  
    button.show();  
}
```

Qt: Hello World



helloworld.cc (2):

```
HelloWorld::~HelloWorld()  
{  
}
```

```
void HelloWorld::button_clicked()  
{  
    printf ("%s\n", hellostr.toAscii().data());  
}
```

- 1 v rámci IDE (Qt Creator)
- 2 „ručně“ překladačem C++ (+ moc) se zadáním cest k hlavičkovým souborům a knihovnám nebo s generováním Makefile ze souboru projektu pomocí qmake:

```
qmake -project -o project.pro
qmake project.pro
moc -o moc_*.cpp *.h
-I<qt_include_dir> -L<qt_lib_dir> -lQtGui -lQtCore
```