

systemove programovanie
gtk programovanie

29 / 03 / 2011

gtkbuilder

- * vyvoj formularov v kode je uzitocny
- * ale v pripade zlozitejsich layoutov je neprehladny
- * alternativa: GtkBuilder
- * UI je deklarovane v XML
 - * rovnake ako vo WPF
 - * a mnohych inych technologiach

gtkbuilder hello world

```
<?xml version="1.0"?>
<interface>
  <object class="GtkWindow" id="window">
    <child>
      <object class="GtkButton" id="button1">
        <property name="label">
          Hello World!
        </property>
      </object>
    </child>
  </object>
</interface>
```

gtk+ hello world

```
int main(int argc, char * argv[]) {
    GtkWidget * window;
    GtkBuilder * builder;

    gtk_init(&argc, &argv);

    builder = gtk_builder_new();
    if(!gtk_builder_add_from_file(builder, "ui.xml", NULL)) {
        printf("Cannot load UI definition. %s");
        return EXIT_FAILURE;
    }

    window = GTK_WIDGET(gtk_builder_get_object(builder,
                                                "window"));

    gtk_widget_show_all(window);
    gtk_main();
    return EXIT_SUCCESS;
}
```

XML subor s
definiciami

ID komponentu

zakladne funkcie gtkbuildera

```
gtk_builder_new();
```

- vytvori novu instanciu buildera

```
gtk_builder_add_from_file(builder, "ui.xml", NULL)
```

- nacita definiciu zo suboru

```
GTK_WIDGET(gtk_builder_get_object(builder, "window"))
```

- vrati komponent podľa id

gtk+ hello world

datovy typ pre chyby
(simulacia vynimiek)

```
GtkWidget * window;  
GtkBuilder * builder;  
GError * error = NULL;
```

```
...  
builder = gtk_builder_new();  
  
if(!gtk_builder_add_from_file(builder, "ui.xml",  
                               &error))  
{  
    g_printerr("Cannot load UI definition: %s",  
              error->message);  
    return EXIT_FAILURE;  
}
```

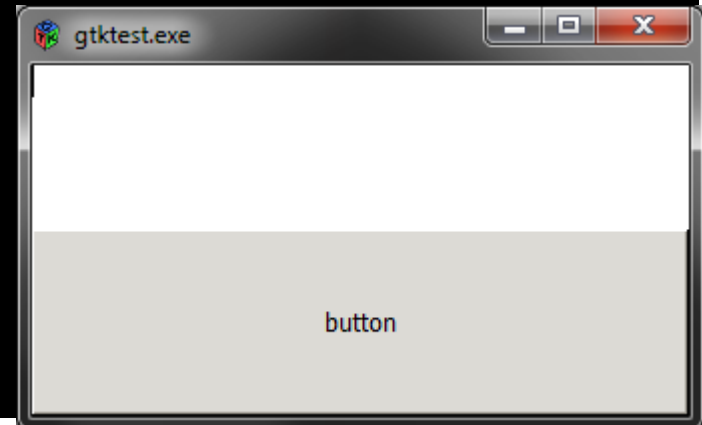
vypis chyby

```
GtkWidget * widget = GTK_WIDGET(gtk_builder_get_object(builder,
```

UI s vboxom

```
<?xml version="1.0"?>
<interface>
  <object class="GtkWindow" id="window">
    <child>
      <object class="GtkVBox" id="vbox">
        <child>
          <object class="GtkTextView" id="textview" />
        </child>
        <child>
          <object class="GtkButton" id="button1">
            <property name="label">button</property>
          </object>
        </child>
      </object>
    </child>
  </object>
</interface>
```

povolene
properties su v
dokumentaciji



UI s vboxom a packing

```
<object class="GtkVBox" id="vbox1">
  <property name="visible">True</property>
  <property name="orientation">vertical</property>
  <child>
    <object class="GtkButton" id="button1">
      <property name="label">button</property>
      <property name="visible">True</property>
    </object>
    <packing>
      <property name="expand">False</property>
      <property name="fill">False</property>
      <property name="position">1</property>
    </packing>
  </child>
</object>
```

packing vlastnosti
(vid' dokumentácia)

signals

```
<interface>  
  <object class="GtkWindow" id="window">  
    <signal name="delete_event"  
            handler="gtk_main_quit" />  
  </signal>  
</object>  
</interface>
```

funkcia, ktora sa ma zavolat

v kode potom zavolame:

```
gtk_builder_connect_signals(builder, NULL);
```

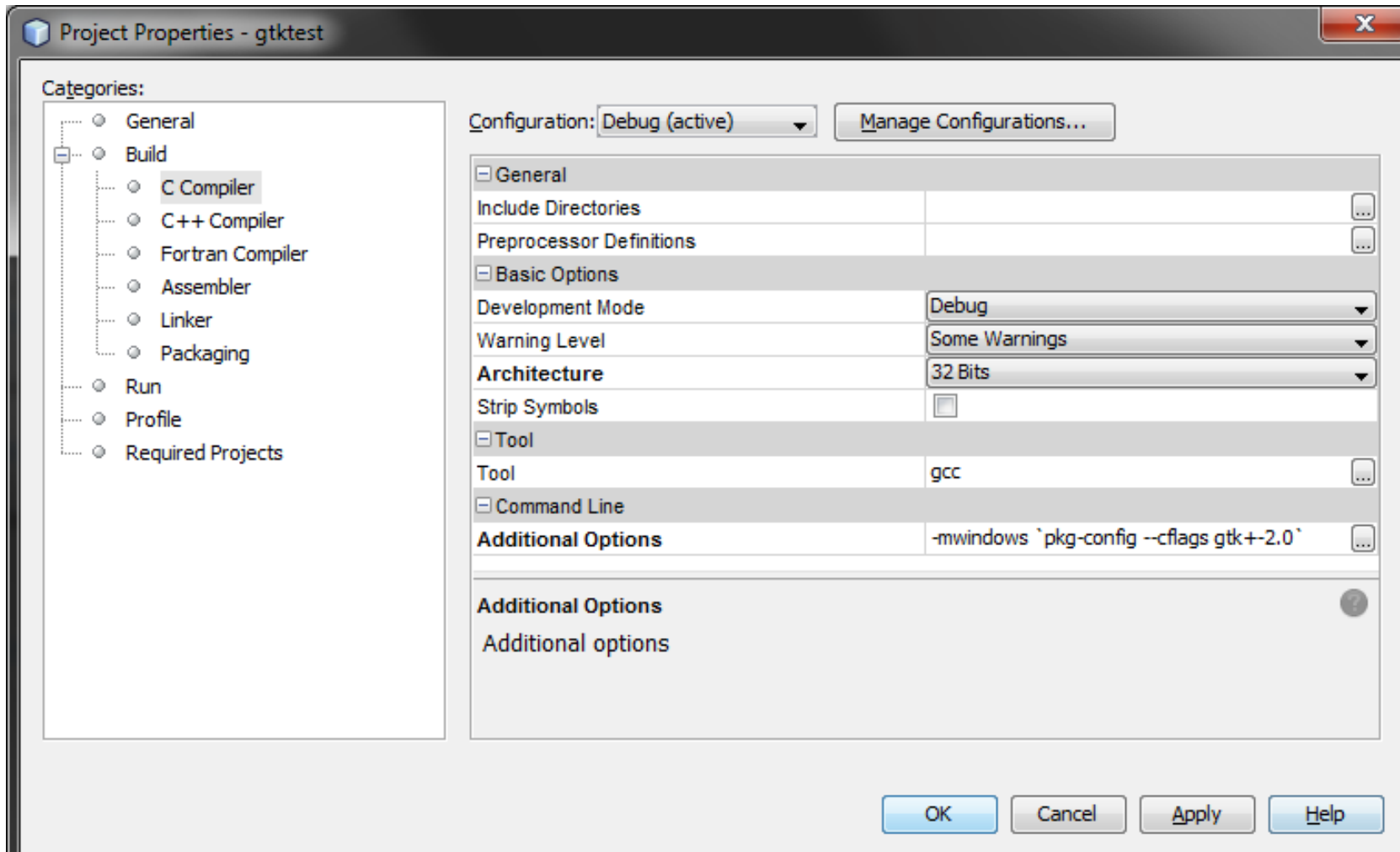
nuansy na windowse

- vsetky obsluzne funkcie musia byt oznacene makrom `G_MODULE_EXPORT`

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

nuansy na windowse

- aplikacia defaultne startuje konzolu
- ak sa chceme zbavit konzoly:
 - parameter
–mwindows pre GCC



ZOZNAMY

- v XML jednoduchšia deklaracia
- zasada: zoznam je degenerovany strom
- trieda GtkTreeView
- strom =
 - stlpce: s konkrétnym datovým typom
 - cellrenderery: formátuju vzhľad
 - data: model v MVC

ZOZNAMY: modely: GtkListStore

- zoznam ma len jeden stlpec
- specifikujeme datove typy
 - gint: int
 - gchararray: retazec
 - ...

```
<object class="GtkListStore" id="liststore">  
  <columns>  
    <column type="gchararray" />  
  </columns>  
</object>
```

Deklaracia v XML

```
<object class="GtkTreeView" id="treeview">  
  <property name="model">liststore</property>  
  <property name="headers-visible">FALSE</property>  
  <child>  
    <object class="GtkTreeViewColumn" id="col1">  
      ....  
    </object>  
  </child>  
</object>
```

referencia
na model

deklaracia
stlpcov

Deklaracia v XML: stlpce

```
<object class="GtkTreeViewColumn" id="col1">
  <property name="title">Údaje</property>
  <child>
    <object class="GtkCellRendererText"
      id="cellrenderertext1" />
    <attributes>
      <attribute name="text">0</attribute>
    </attributes>
  </child>
</object>
```

popisok
stlpca

formatujeme
stlpec s
indexom 0

renderer

zoznam ma len jeden
stlpec = staci jediny
GtkTreeViewColumn