

14th April 2009

GtkComboBox widget - part 1

I'm back with first meaningful post. We'll be looking at GtkComboBox widget today. Why? Because [GTK+'s official tutorial \[http://library.gnome.org/devel/gtk-tutorial/stable/x1065.html\]](http://library.gnome.org/devel/gtk-tutorial/stable/x1065.html) only talks about deprecated GtkCombo. Let's start with short introduction. [GtkComboBox \[http://library.gnome.org/devel/gtk/stable/GtkComboBox.html\]](http://library.gnome.org/devel/gtk/stable/GtkComboBox.html) is a widget whose main task is to let users select from a list or a tree of valid choices. It is quite powerful widget and is able to display everything from simple text strings to very complex data trees with images, progress bars, etc. All this is reflected in it's API, which can be separated in "simple" (text) API and "complex" API, which is closely related to GTK+'s GtkCellLayout interface. We'll be looking at the simple API functions today.

Contents of this tutorial:

1. [GtkComboBox widget - part 1 \(simple API\) \[http://tadeboro.blogspot.com/2009/04/gtkcombobox-widget-part-1.html\]](http://tadeboro.blogspot.com/2009/04/gtkcombobox-widget-part-1.html)
2. [GtkComboBox widget - part 2 \(complex API\) \[http://tadeboro.blogspot.com/2009/04/as-promised-im-back-with-second-part-of.html\]](http://tadeboro.blogspot.com/2009/04/as-promised-im-back-with-second-part-of.html)
3. [GtkComboBox widget - part 3 \(test your knowledge\) \[http://tadeboro.blogspot.com/2009/04/gtkcombobox-widget-part-3.html\]](http://tadeboro.blogspot.com/2009/04/gtkcombobox-widget-part-3.html)

Simple API

Simple API consists of 7 functions:

- **gtk_combo_box_new_text** for creating combo box
- **gtk_combo_box_append_text**, **gtk_combo_box_prepend_text** and **gtk_combo_box_insert_text** for adding items to combo box
- **gtk_combo_box_get_active_text** for obtaining currently selected item
- **gtk_combo_box_get_active** for obtaining active item's index number (this function is part of both, simple and complex, APIs)
- **gtk_combo_box_remove_text** from removing items from combo box

But enough talking, let's see some code now!;

This is a "Hello World! on steroids" application that shows how to create simple text-only combo box using simple API. It demonstrates using all of the functions from simple API. Yes, that is all there is to it.

```
/* Test me with:
* gcc -o combo combo.c $(pkg-config --cflags --libs gtk+-2.0) && ./combo
*/

#include <gtk/gtk.h>

/* This function gets called when currently selected item changes */
static void
cb_changed( GtkWidget *combo,
            gpointer    data )
{
    /* Obtain currently selected string from combo box */
    gchar *string = gtk_combo_box_get_active_text( combo );

    /* Print it to the console - if nothing is selected, print NULL */
    g_print( "Selected (simple): >> %s <<\n", ( string ? string : "NULL" ) );
}
```

```
    /* Free string */
    g_free( string );
}

/* This function deletes currently selected item from combo box */
static void
cb_delete( GtkWidget *button,
           GtkWidget *combo )
{
    gint index;

    /* Get currently selected item's index */
    index = gtk_combo_box_get_active( combo );

    /* Remove this item from combo */
    gtk_combo_box_remove_text( combo, index );
}

int
main( int   argc,
      char **argv )
{
    /*Common variables */
    GtkWidget *window;
    GtkWidget *vbox;
    GtkWidget *frame;
    GtkWidget *combo;
    GtkWidget *button;

    /* Initialization */
    gtk_init( &argc, &argv );

    /* Create main window */
    window = gtk_window_new( GTK_WINDOW_TOPLEVEL );
    g_signal_connect( G_OBJECT( window ), "destroy",
                     G_CALLBACK( gtk_main_quit ), NULL );
    gtk_container_set_border_width( GTK_CONTAINER( window ), 10 );

    /* Create vbox */
    vbox = gtk_vbox_new( FALSE, 6 );
    gtk_container_add( GTK_CONTAINER( window ), vbox );

    /* Create frame */
    frame = gtk_frame_new( "Text API" );
    gtk_box_pack_start( GTK_BOX( vbox ), frame, FALSE, FALSE, 0 );

    /* Create combo box using text API and add some data to it */
    combo = gtk_combo_box_new_text();
    gtk_container_add( GTK_CONTAINER( frame ), combo );
    gtk_combo_box_append_text( GTK_COMBO_BOX( combo ), "Hello World once" );
}
```

```
gtk_combo_box_prepend_text( GTK_COMBO_BOX( combo ), "Hello World twice" );
gtk_combo_box_insert_text( GTK_COMBO_BOX( combo ), 1, "Hello World last time" );

/* Connect signal to tour handler function */
g_signal_connect( G_OBJECT( combo ), "changed",
                  G_CALLBACK( cb_changed ), NULL );

/* Add button that, when clicked, deletes currently selected entry */
button = gtk_button_new_with_mnemonic( "_Delete selected item" );
gtk_box_pack_start( GTK_BOX( vbox ), button, FALSE, FALSE, 0 );
g_signal_connect( G_OBJECT( button ), "clicked",
                  G_CALLBACK( cb_delete ), GTK_COMBO_BOX( combo ) );

/* Show our application and start main loop */
gtk_widget_show_all( window );
gtk_main();

return( 0 );
}
```

See you in part 2 of this tutorial, where we'll tackle advanced API.

Bye.

Posted 14th April 2009 by [Tadej Borovšak](#)

Labels: [GTK+](#), [GtkComboBox](#), [tutorial](#)

12

[View comments](#)



Louise Hoffman 25 October, 2009 02:27

Dear Tadej,

This was exactly what I needed to solve the problem I had with the combobox =)

What I don't understand is this:

```
g_signal_connect( G_OBJECT( button ), "clicked", G_CALLBACK( cb_delete ), GTK_COMBO_BOX(
combo ) );
```

Why is "GTK_COMBO_BOX(combo)" correct and not just "combo"?

Also, why must I give GtkButton *button as argument to cb_delete()?

I mean, I don't use "*button" for anything in cb_delete()...?

Hugs,
Louise

[Reply](#)



tadeboro 25 October, 2009 09:44

Hello.

If you have a look at API reference [here](#), you'll see that function that will be connected to "clicked" signal of GtkButton should have a prototype:

```
void callback( GtkButton *, gpointer )
```

This is why first parameter of `cb_delete()` is `GtkButton`.

As for the casting of the last parameter of `g_signal_connect` macro, things are really simple.

Each callback function has `gpointer` as last parameter. We can cast that parameter to anything we want in order to make our lives a bit easier.

I defined callback function as: **`void cb_clicked(GtkButton *, GtkComboBox *)`** and this means that last argument to `g_signal_connect` needs to be `GtkComboBox *`.

If I were to define callback as: **`void cb_clicked(GtkButton *, GtkWidget *)`** it would be fine to pass `combo` as last parameter to `g_signal_connect`.

Tadej

[Reply](#)



Louise Hoffman 25 October, 2009 23:51

Thanks for the explanations =)

[Reply](#)



Louise Hoffman 26 October, 2009 00:27

Can I ask another question, as I haven't seen anyone address this.

I will be using a `ComboBox` for selecting an audio channel. If the loaded file have 2 channels, the `ComboBox` should have 0 and 1 as options, and in case of 4 channels, it should have 0,1,2,3 and so on.

As `gtk_combo_box_append_text()` expects a string as the 2nd parameter, how do I insert an integer instead?

And as `gtk_combo_box_get_active_text` returns a string, how do I convert it back to an integer?

[Reply](#)



Ravi 10 March, 2010 11:05

hello,Tadej

for GTKpy, i want to connect `on_button_press` event with `combobox` but it's not working just see following code if you can help me

```
combo=gtk.combo_box_entry_new_text()
combo.connect("on_button_press",self.setmode)
```

```
def setmode(self)
    print 'it's not woking'
```

when i connect using "event" i get only `gtk.expose` not `gtk.button` why??

[Reply](#)

**Monika** 24 July, 2010 15:47

Hi Mr.Tadej,

Can u help me about Combo box widget, I am trying to implement same thing through xml file. But I didnt get the Output..

So How to create same application through xml.Main problem with CellRenderering..

Monika

[Reply](#)**Sumitro** 29 January, 2011 04:29

Thanks for the combobox tutorial. It really saved my day!
Just needed a sample for the syntax

[Reply](#)**Anonymous** 09 March, 2011 13:41

Hi Tadej.

I was just wondering if it is possible to type text in GtkComboBoxEntry and after return retrieve typed text. I would expect for a activate or some other signal to be emitted, but so far I have found nothing.

Thank you in advance, Slartius

[Reply](#)**Anonymous** 09 March, 2011 13:44

Hi, Tadej

I would like to type a text into a GtkcomboBoxEntry and after a return retrieve typed text and do something with it. I Was expecting a activate signal or something like that, but have so far found nothing similar. Do you maybe know if this is possible.

Thank you in advance, Slartius

[Reply](#)**Anonymous** 09 March, 2011 13:46

Hi, Tadej

I would like to type a text into a GtkcomboBoxEntry and after pressing return, to retrieve typed text and do something with it. I was expecting a activate signal or something like that, but have so far found nothing similar. Do you maybe know if this is even possible.

Thank you in advance, Slartius

[Reply](#)**Aleš** 09 March, 2011 14:01

Hi, Tadej

I would like to type a text into a GtkcomboBoxEntry and after a return retrieve typed text and do something with it. I Was expecting a activate signal or something like that, but have so far found nothing similar. Do you maybe know if this is possible.

Thank you in advance, Slartius

[Reply](#)



Aleš 09 March, 2011 14:02

Hi, Tadej

I would like to type a text into a GtkcomboBoxEntry and after a return retrieve typed text and do something with it. I Was expecting a activate signal or something like that, but have so far found nothing similar. Do you maybe know if this is possible.

Thank you in advance, Slartius

[Reply](#)

Enter your comment...

Comment as:

Google Accou ▾

[Publish](#)

[Preview](#)