

[About](#)[Users](#)[Administrators](#)[Developers](#)

Signals

Programming with *gtkmm* 3

gtkmm, like most GUI toolkits, is *event-driven*. When an event occurs, such as the press of a mouse button, an appropriate signal will be *emitted* by the Widget that was pressed. Each Widget has a different set of signals it can emit. To make a button click result in an action, we set up a *signal handler* to catch the button's "clicked" signal.

gtkmm uses the libsigc++ library to implement signals. Here is an example line of code that connects a `Gtk::Button`'s "clicked" signal with a signal handler called "on_button_clicked":

```
m_button1.signal_clicked().connect( sigc::mem_fun(*this,
    &HelloWorld::on_button_clicked) );
```

For more detailed information about signals, see the [appendix](#).

For information about implementing your own signals rather than just connecting to the existing *gtkmm* signals, see the [appendix](#).

[◀ Widgets](#)[Glib::ust](#)[About](#)

[Printing](#)

[Recently Used Documents](#)

[Plugs and Sockets](#)

[Keyboard Events](#)

[Timeouts, I/O and Idle Functions](#)

[Memory management](#)

[Glade and Gtk::Builder](#)

[Internationalization and Localization](#)

[Custom Widgets](#)

[Multi-threaded programs](#)

[Recommended Techniques](#)

[Contributing](#)

[The RefPtr smartpointer](#)

[Signals](#)

[Creating your own signals](#)

[Comparison with other signalling systems](#)

[gtkmm and Win32](#)

[Working with gtkmm's Source Code](#)

[Wrapping C Libraries with gmmproc](#)

The GNOME Project

[About Us](#)
[Get Involved](#)

Resources

[Documentation](#)
[Wiki](#)

News

[Latest Release](#)
[Planet GNOME](#)

Teams
The GNOME Foundation
Support GNOME
Contact

Mailing Lists
IRC Channels
Bug Tracker
Development Code
Build Tool

Development News
Identi.ca
Twitter

Copyright © 2005–2014 **The GNOME Project**

Optimised for standards. Hosted by Red Hat.