

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

Simple Example

Programming with *gtkmm* 3

To begin our introduction to *gtkmm*, we'll start with the simplest program possible. This program will create an empty 200 x 200 pixel window.

Source Code

File: `base.cc` (For use with *gtkmm* 3, not *gtkmm* 2)

```
#include <gtkmm.h>

int main(int argc, char *argv[])
{
    auto app =
        Gtk::Application::create(argc, argv,
            "org.gtkmm.examples.base");

    Gtk::Window window;
    window.set_default_size(200, 200);

    return app->run(window);
}
```

We will now explain each line of the example

```
#include <gtkmm.h>
```

All *gtkmm* programs must include certain *gtkmm* headers; `gtkmm.h` includes the entire *gtkmm* kit. This is not a good idea, because it includes a megabyte or so of headers, but for simple programs, it suffices.

The next statement:

```
Glib::RefPtr<Gtk::Application> app = Gtk::Application::create(argc, argv,
```

creates a `Gtk::Application` object, stored in a `RefPtr` smartpointer. This is needed in all *gtkmm* applications. The `create()` method for this object initializes *gtkmm*, and checks the arguments passed application on the command line, looking for standard options such as `--display`. It takes these from the argument list, leaving anything it does not recognize for your application to parse or ignore. This ensures *gtkmm* applications accept the same set of standard arguments.

The next two lines of code create a window and set its default (initial) size:

```
Gtk::Window window;  
window.set_default_size(200, 200);
```

The last line shows the window and enters the *gtkmm* main processing loop, which will finish when the window is closed. Your `main()` function will then return with an appropriate success or error code.

```
return app->run(window);
```

After putting the source code in `simple.cc` you can compile the above program with `gcc` using:

```
g++ simple.cc -o simple `pkg-config gtkmm-3.0 --cflags --libs`
```

Note that you must surround the `pkg-config` invocation with backquotes. Backquotes cause the shell to execute the command inside them, and to use the command's output as part of the command line. Note that `simple.cc` must come before the `pkg-config` invocation on the command line.

◀ [Simple Example](#)

[Headers and Lin](#)

[About](#)

The GNOME Project

- About Us
- Get Involved
- Teams
- The GNOME Foundation
- Support GNOME
- Contact

Resources

- Documentation
- Wiki
- Mailing Lists
- IRC Channels
- Bug Tracker
- Development Code
- Build Tool

News

- Latest Release
- Planet GNOME
- Development News
- Identi.ca
- Twitter

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

Optimised for standards. Hosted by Red Hat.