

# Howto: Rapid GUI creation with FreeBasic and glade/Gtk

by Klaus Siebke, [www.siebke.com](http://www.siebke.com)

## Preface

*For a longer time I was looking for a programming environment that offers both: easy to use and platform independent. Some weeks ago I found FreeBasic to fulfill these requirements perfectly. Just code your program once and run it in native mode under MS Windows and Linux (and if you like you can still run it under DOS). Additionally FreeBasic in combination with glade/Gtk offers a simple framework to create identical GUI applications for Windows and for Linux.*

*The only lack so far is a tool to simplify the GUI program creation with glade/Gtk. So I wrote a small utility named “**glade2bas**” which will help you to realize your own GUI application with FreeBasic and glade/Gtk.*

## What you need

1. First of all you need the FreeBasic compiler for your preferred OS which can be downloaded from: <http://www.freebasic.net/>.
2. Then you need to install Gtk. If you are using Linux it's already included in your distribution (at least in Ubuntu/Kubuntu or openSuSE). Just go to your package manager (apt or yast) and select it. If you are using MS Windows, download the latest binary from the official server: <ftp://ftp.gtk.org/pub/gtk/> and install it.
3. The next component you need is glade, the user interface builder. For Linux users it's the same as for Gtk: install the package with your package manager – that's it. Users working with MS Windows should go to <http://gladewin32.sourceforge.net/>, download the latest version of glade and install it.
4. An editor to code your application. I would prefer Geany which supports FreeBasic in a very comfortable way. You can compile and run your application “out of the box”. Geany is available for MS Windows and Linux and like all the above software components it's free. You can download Geany from <http://geany.uvena.de/>. However, in some Linux distributions Geany is already included. In this case make sure that the distribution's version is 0.12 or higher (otherwise the FreeBasic support is not implemented).
5. The glade2bas utility to create a shell program out of the glade file. You can download the source of this utility from my page [www.siebke.com/freebasic/glade2bas.zip](http://www.siebke.com/freebasic/glade2bas.zip). Unzip the file in your FreeBasic source directory and compile it with "fbc -s gui glade2bas.bas".

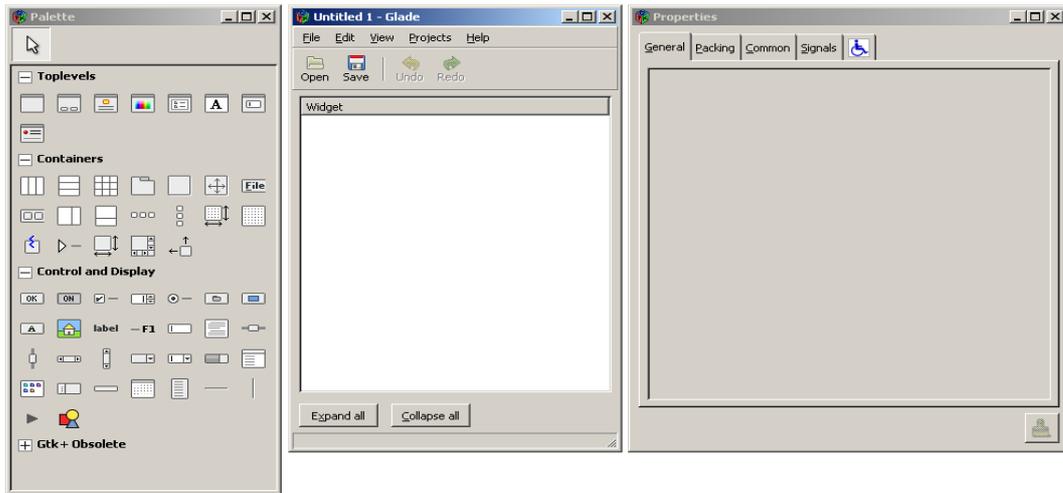
## A first application

Let's start with a simple example. Like you probably know it from other programming languages, it's a small “Hello World” application.

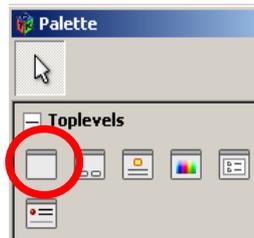
## Designing the interface with glade

To design the window(s) you can either use glade v2 or glade v3 whatever you prefer. Because it's the newer version, I use glade3 in the following example.

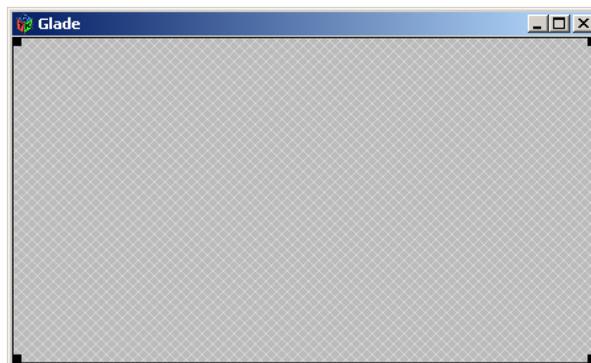
After starting glade-3 you will get this screen:



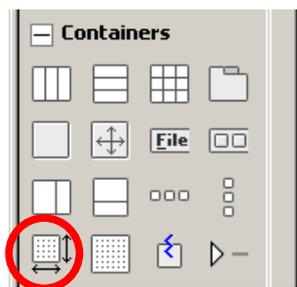
To create a new window for your application, direct your mouse pointer to the “Palette” and choose the first icon under “Toplevels”:



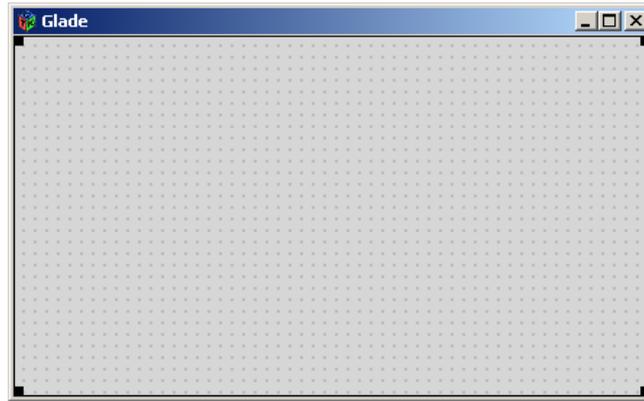
Click on it and your first glade/Gtk window will be created:



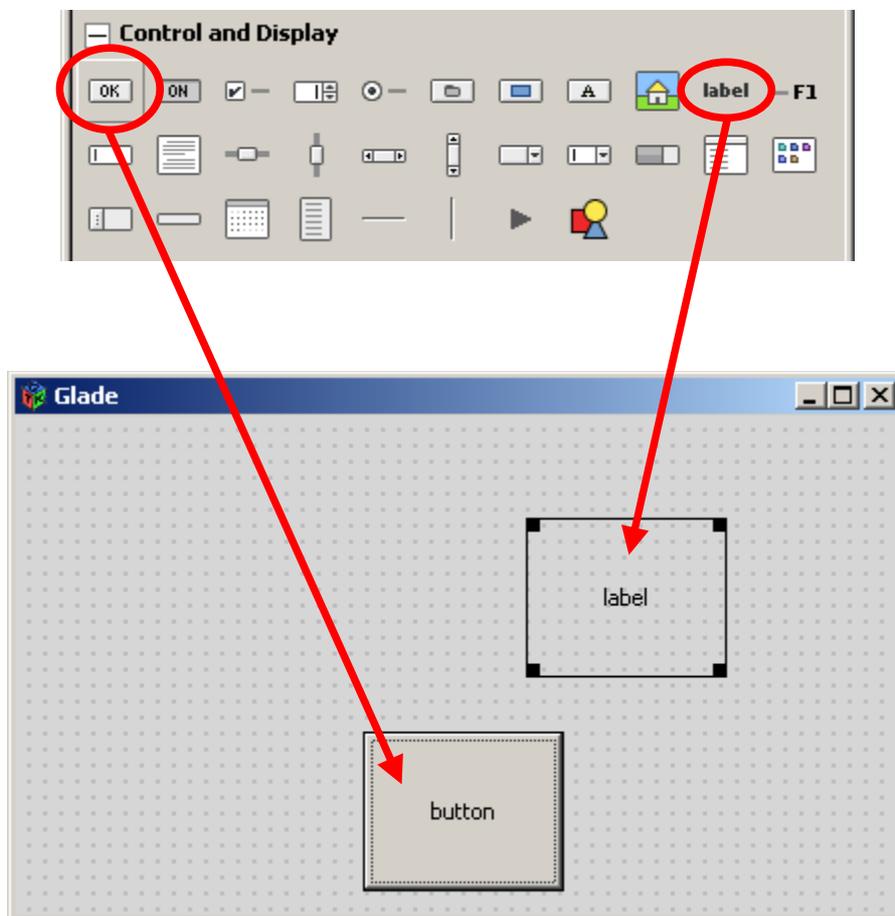
Now you have to add a layout to the window. You will find it in the “Palette” under “Containers”:



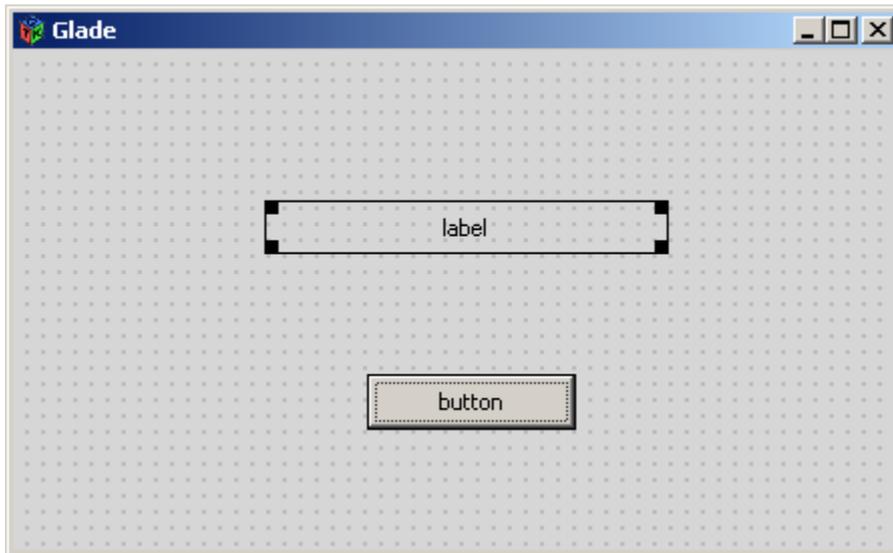
Again, click on the icon and then into your window. The window appearance will change as follows:



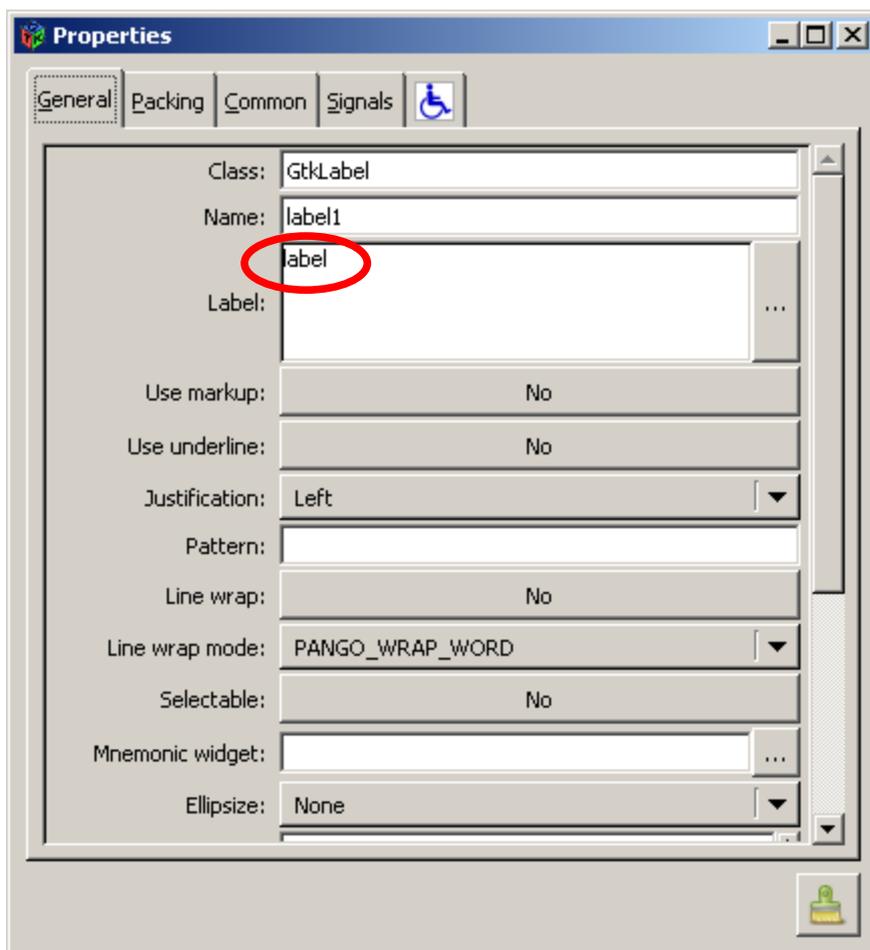
Now you are able to place some elements on the window. In the "Palette" you will find a lot of widgets under "Control and Display". Select the "OK button" and the "label" widget by clicking first on the icon and then on your screen:



If you like, you can now resize and arrange the elements a little bit:



To change the text and the action of the elements click on the widget to be changed and look at the "Properties":

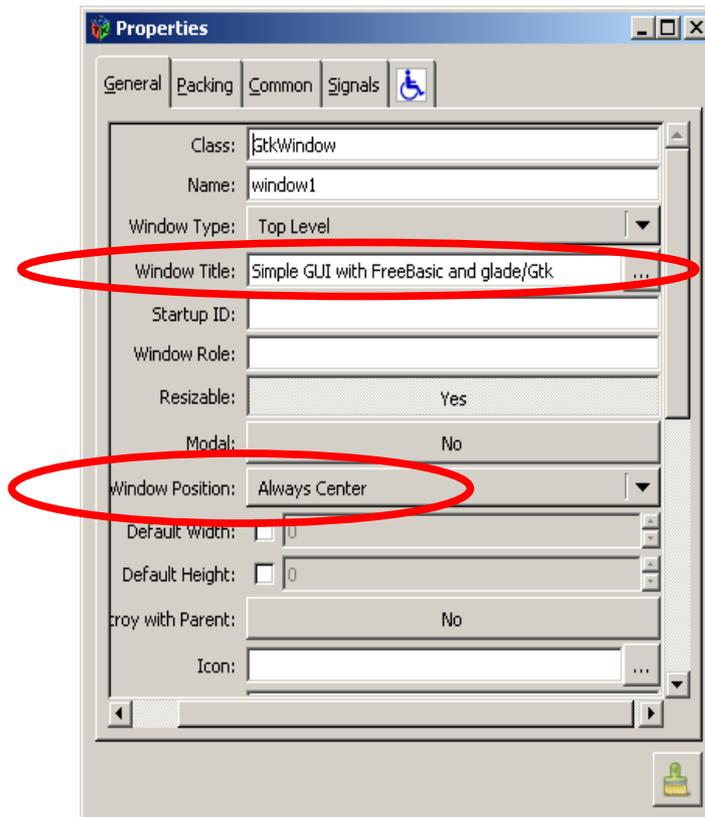


Change the text of the label from "label" to "Hello World!" and the text of the OK button from "button" to "OK".

Your window should now look like this one:

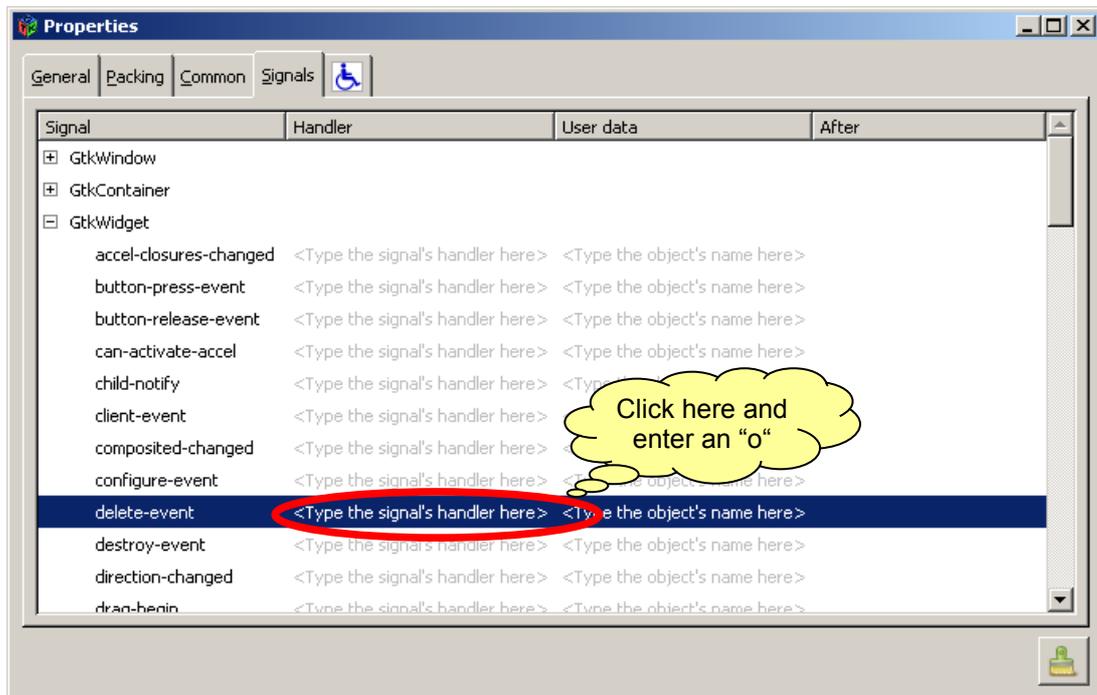


Let's change some properties (title, startup position and size) of the window itself:

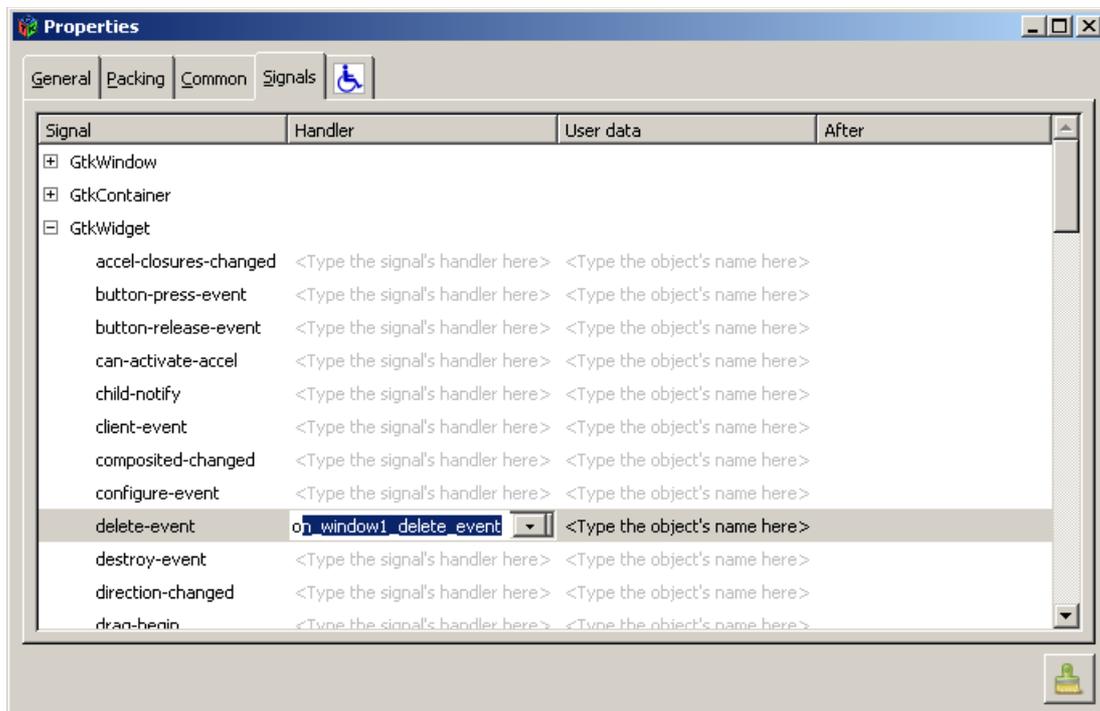


Finally you should define some actions (=signals) to react on in the FreeBasic program:

1. Define a signal to terminate the program when the window is deleted. Go to the properties and click on the "Signals" tab strip. Then go to line "delete-event" and click in the "Handler" column:

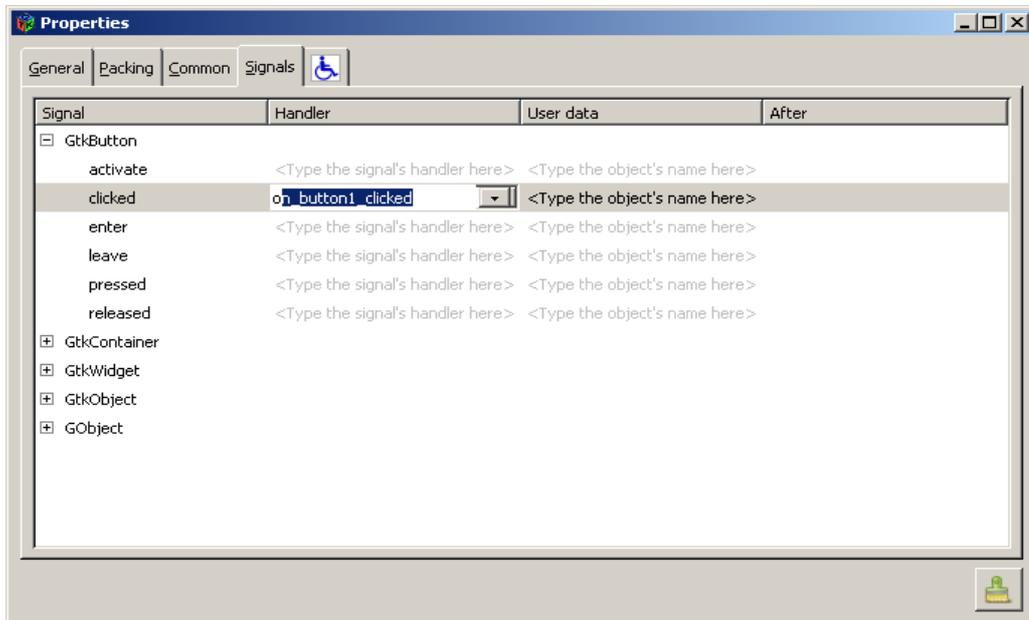


The name of the handler will automatically be completed:

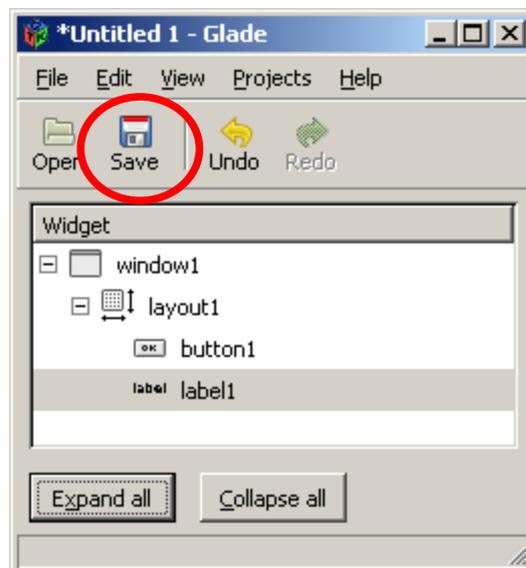


Press Enter to assign the handler name to the event.

2. Define a signal when the OK button is pressed ⇒ Click on the OK button widget on your screen. Go to the properties and click on the “Signals” tab strip (same procedure as with the signal above – the handler name will be automatically completed when you key in “o” and press Enter afterwards):



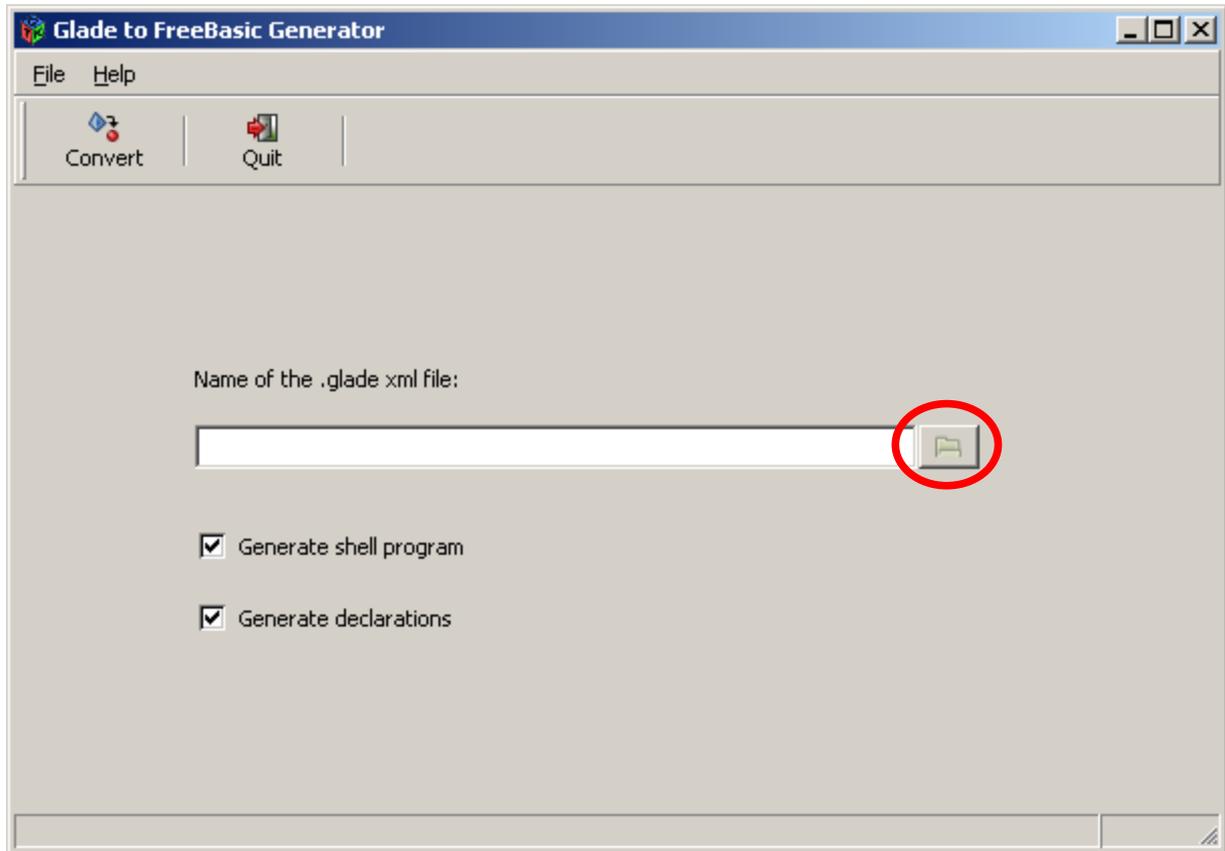
Please save your work now as file “helloworld.glade” by pressing the save button in the glade main window:



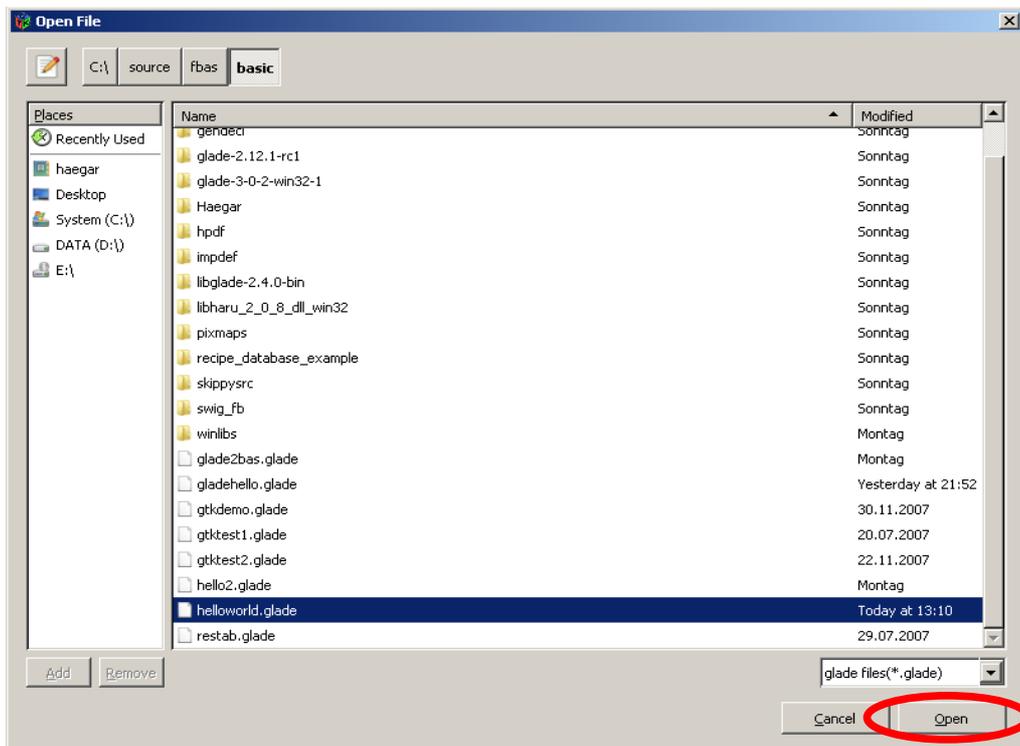
For this simple example that's all on the glade design side. You can now close the application and run the utility glade2bas.

## Convert glade to FreeBasic with glade2bas

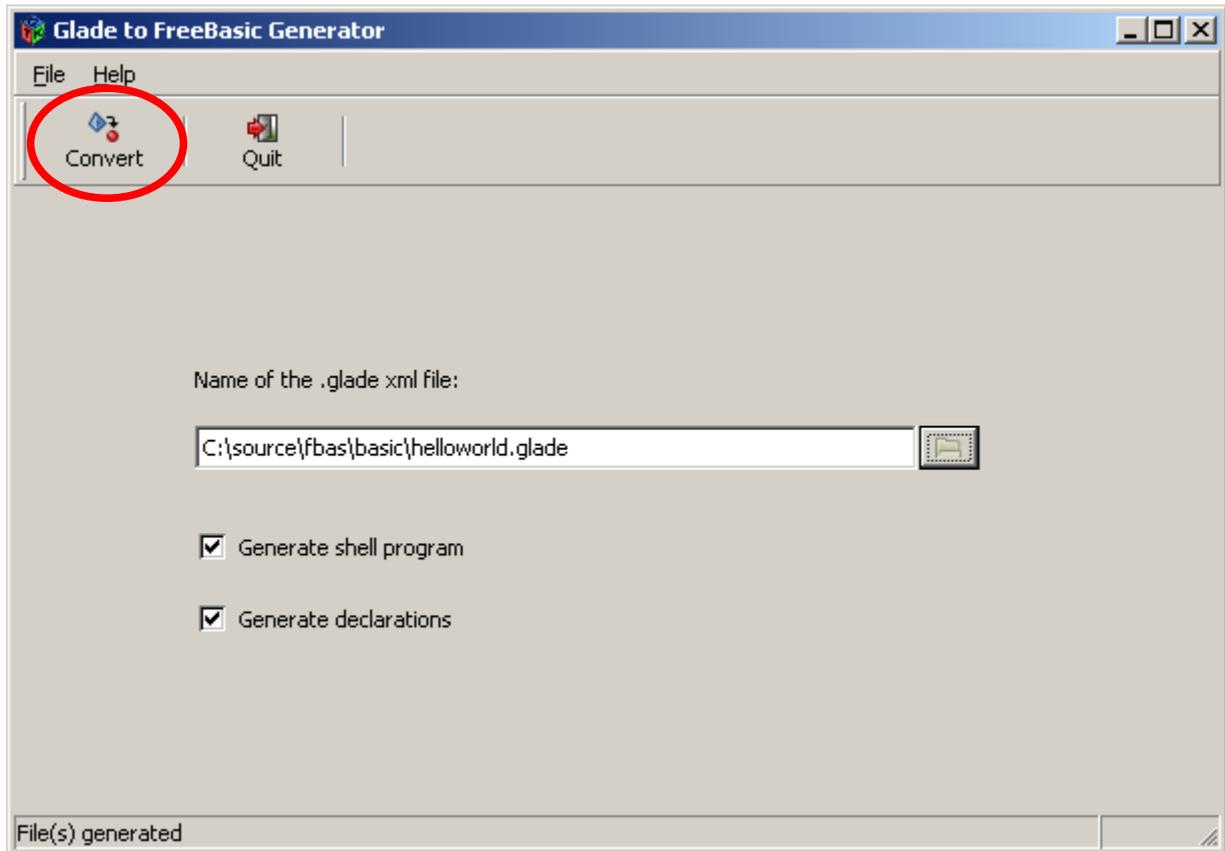
After starting glade2bas the following screen appears:



Click on the open button right to the text field and select the file "helloworld.glade".



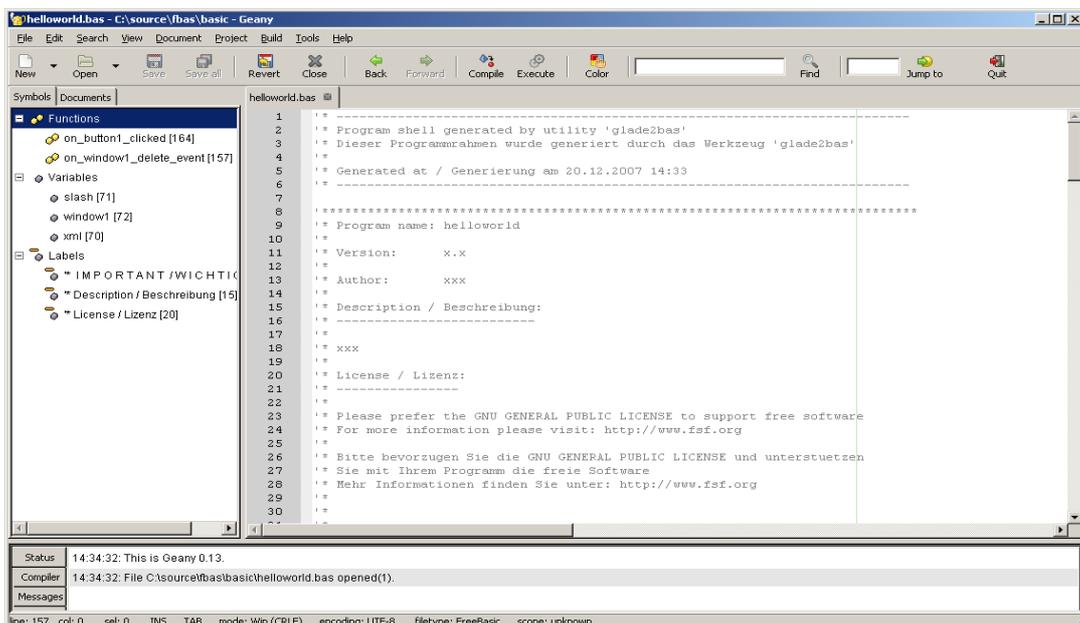
Now you can generate the FreeBasic shell program by clicking on the “Convert” button:



The utility has created 2 files: helloworld.decl and helloworld.bas.

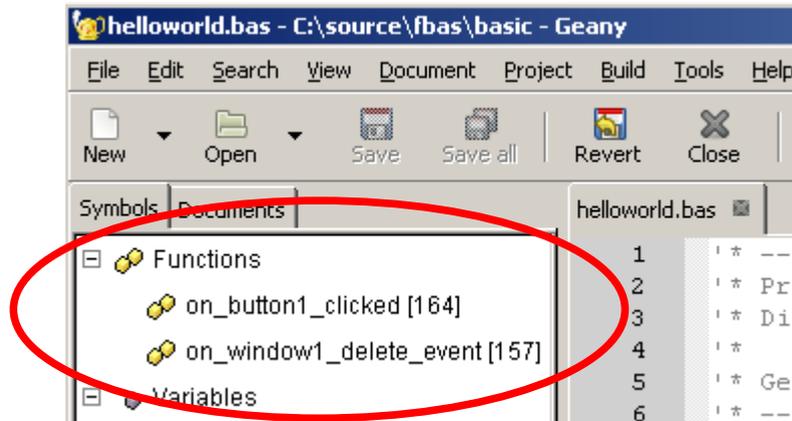
## Add your logic to the application

Lets have a look at the generated source. Start Geany and open the file helloworld.bas:

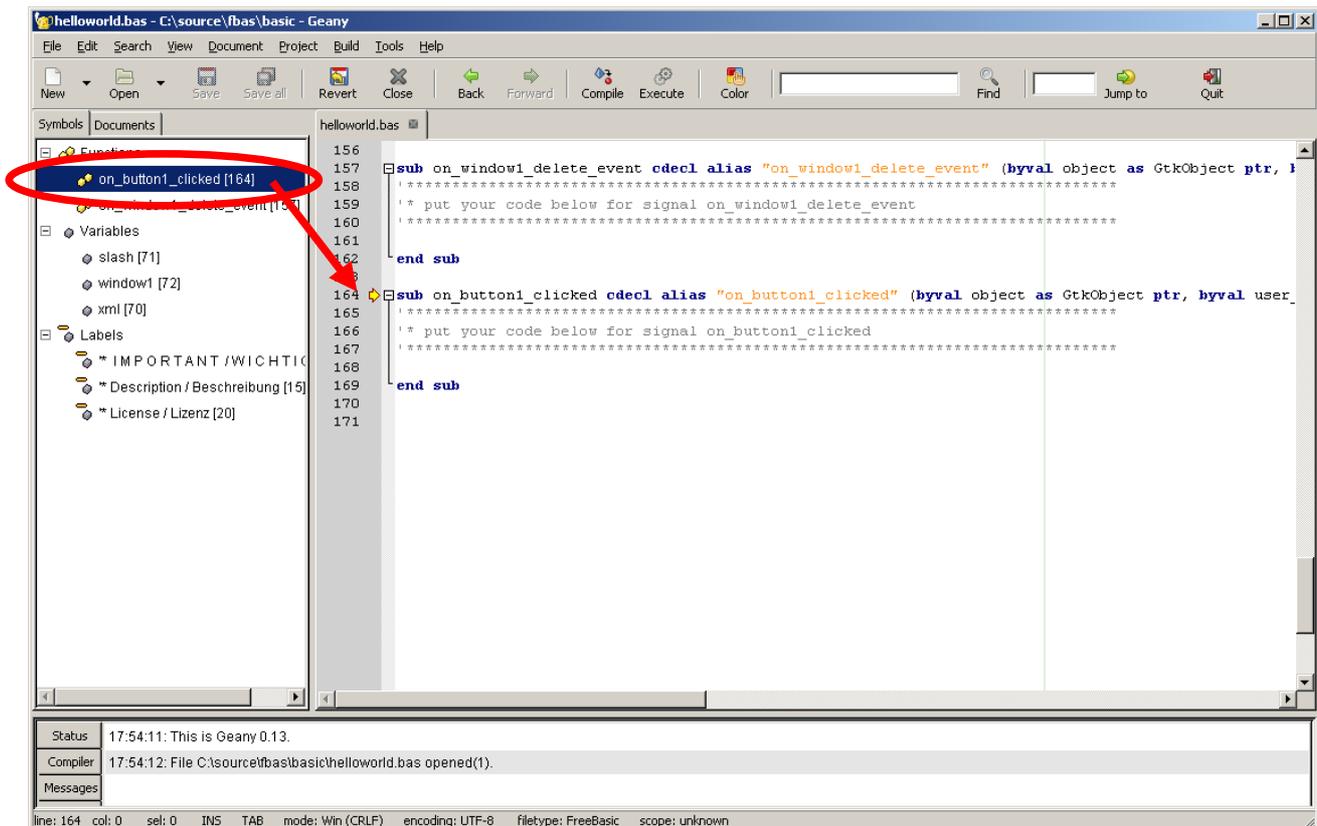


The generated program is already runnable. You can compile and run it. However it will not react on the defined signals. So it makes sense to add some additional statements.

On the top of the left side you can see the signals we defined in the glade design listed by Geany as "functions":



Click on the first function "on\_button1\_clicked" and the editor on the right will directly position to that function (or SUB in FreeBasic notation):



Now you can add a statement to react on the event that is initiated when the OK button is pressed. Just key in:

`print "OK button pressed"`

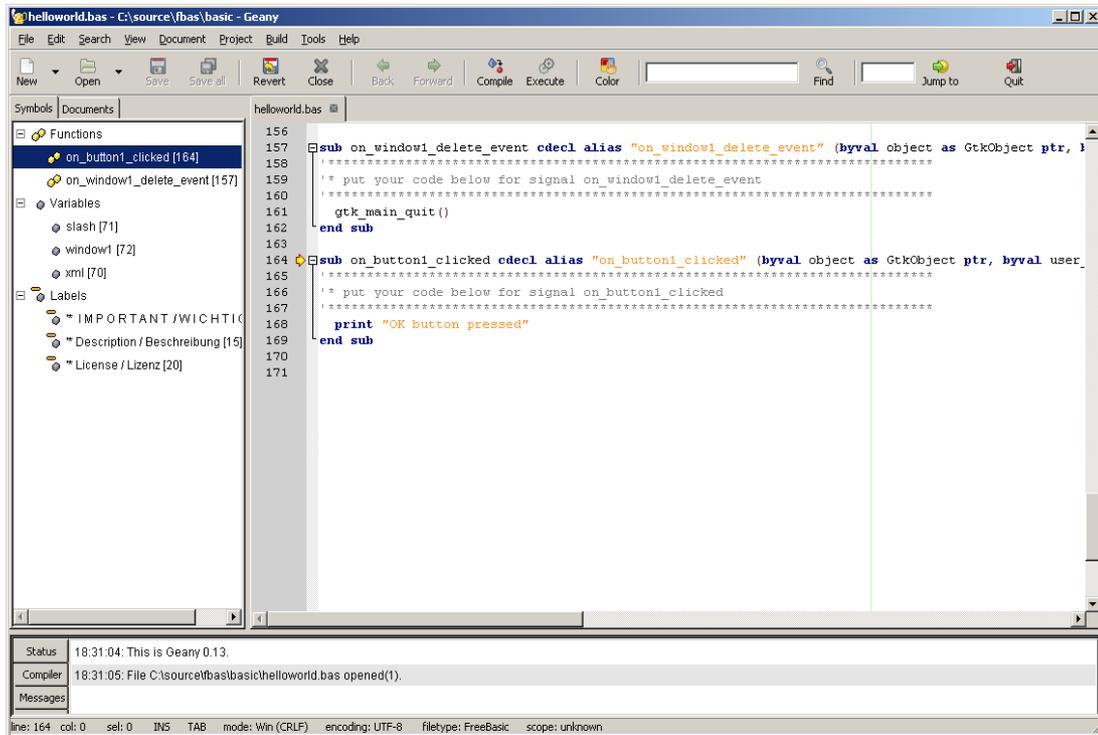
below the comment box.

The other SUB (on\_window1\_delete\_event) shall terminate the program. So key in:

`gtk_main_quit()`

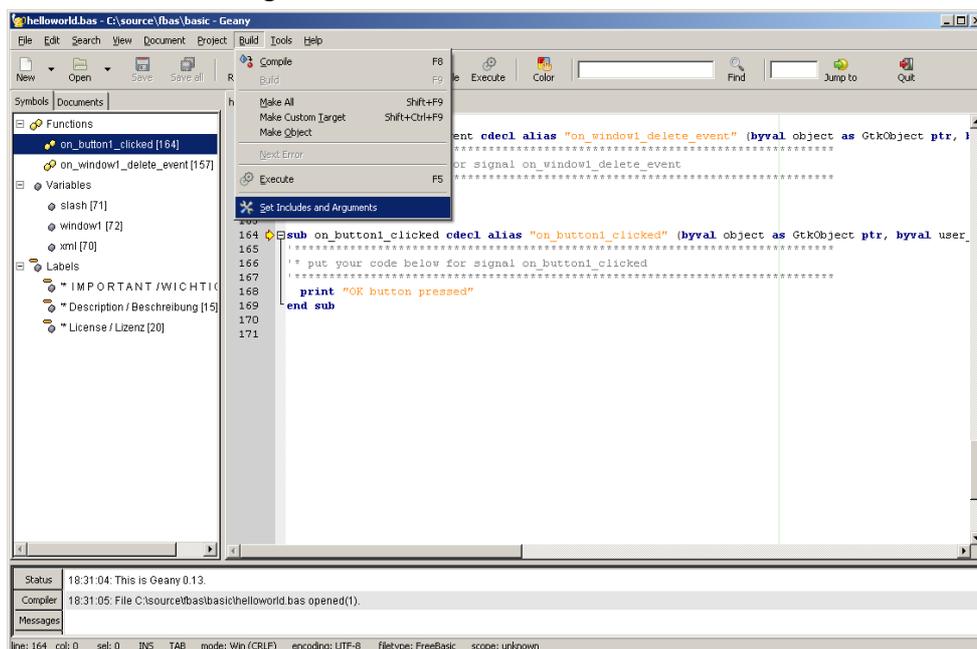
below that comment box.

Now your code should look this way:

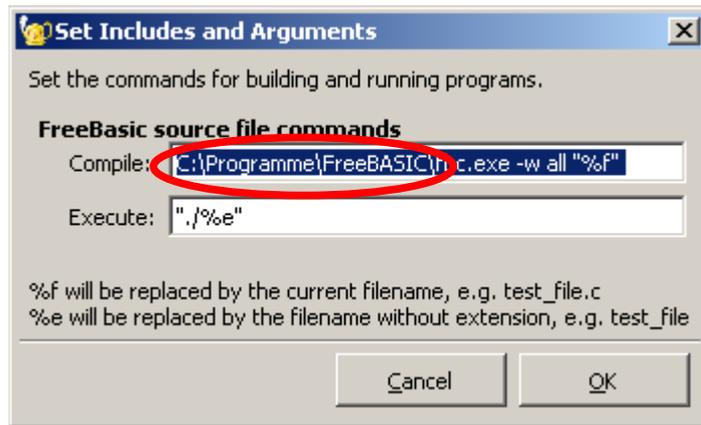


## Compile the application

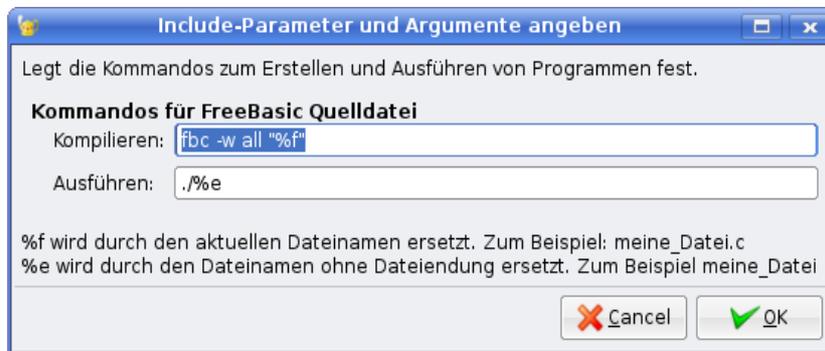
To compile the program directly out of Geany please check the settings. Go to “Build” and then to “Set Includes and Arguments” in the menu:



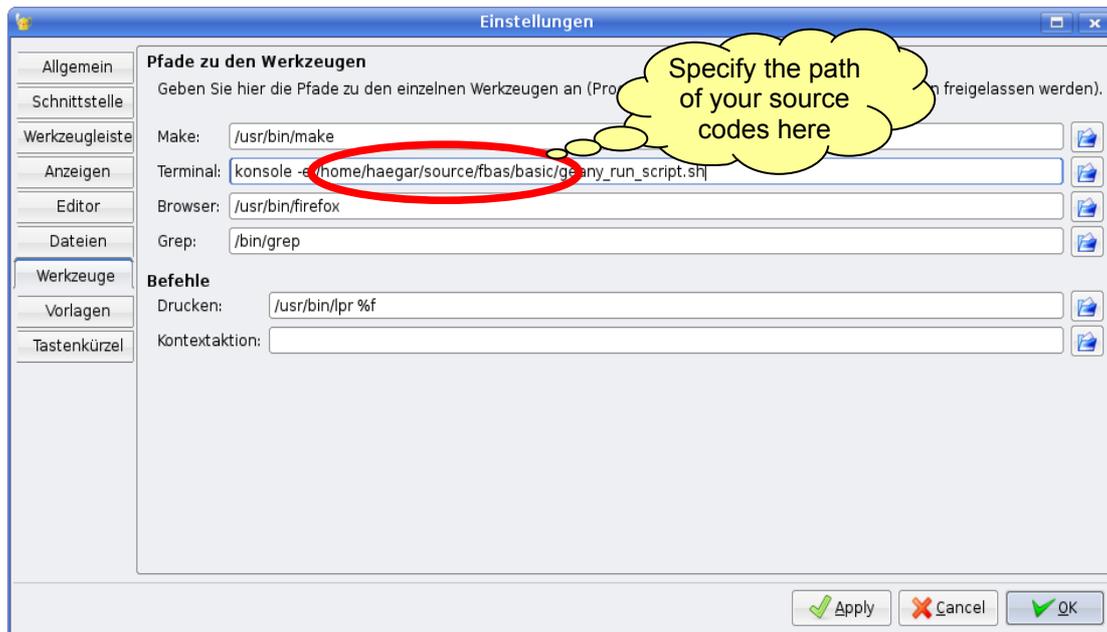
Make sure that the correct path to the FBC executable is specified – for MS Windows I had to change it on my machine as follows:



For Linux you can leave the Geany default values unchanged ...



... but you will have to change the Terminal string in the properties:



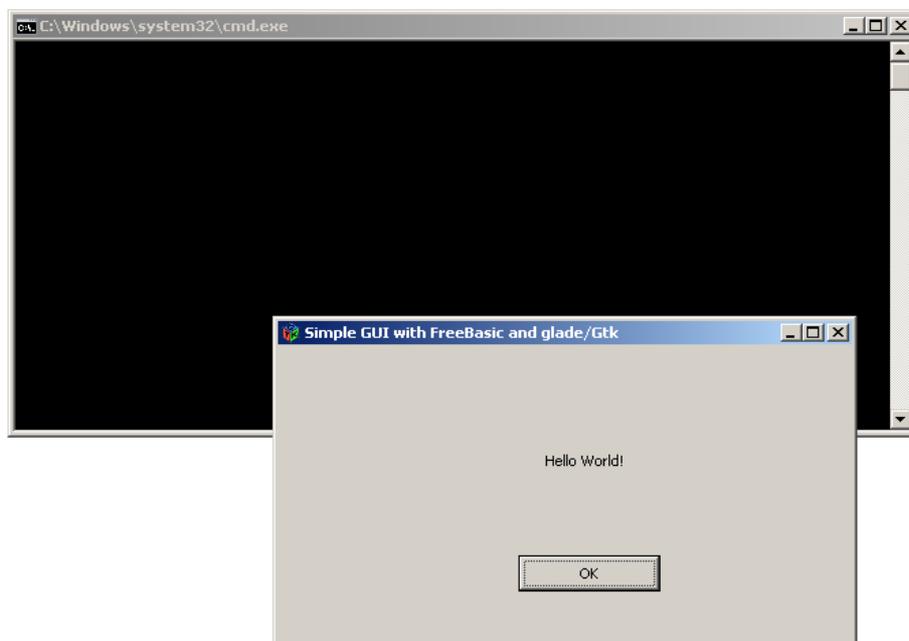
After checking the above settings press the “Compile” button on the tool bar and you will get this messages in the status window:



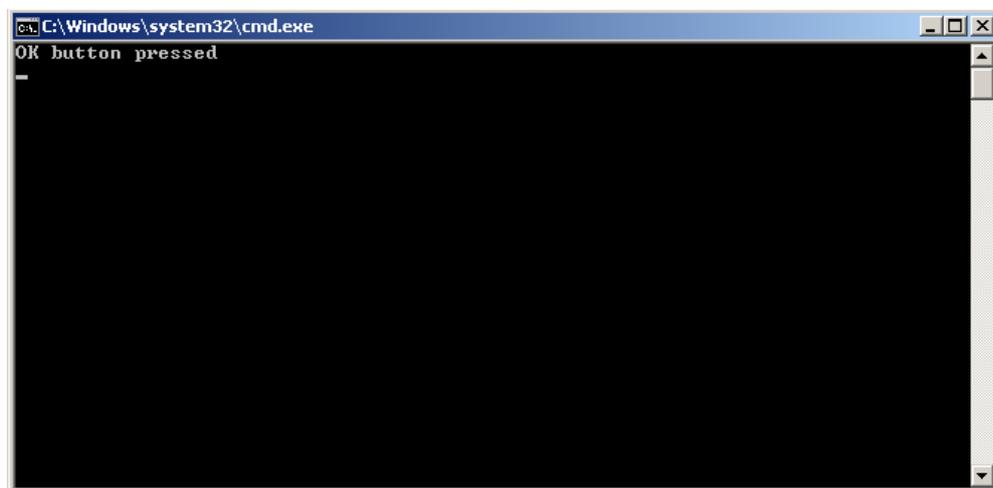
Alternatively you can open a command shell and compile the program manually.

## Run the application

Within Geany simply press the “Execute“ button and the program starts:



Try the OK button and see what happens in the command shell window:

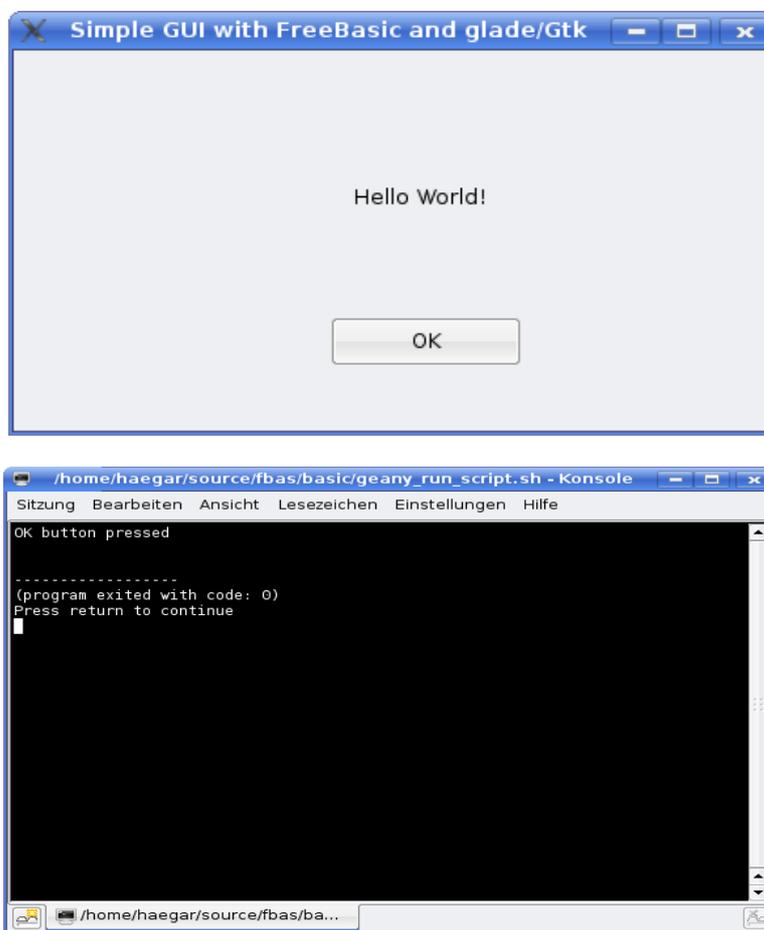


Now click on the “x“ on the right corner of the window ...



... and the application finishes.

Here's how it looks in Linux (kubuntu):



## Summary

In this lesson you learned how to write a simple glade/Gtk application with FreeBasic. For more information about Gtk visit <http://www.gtk.org/>. Hint: the generated shell program and the utility program are fully commented (in english and german) and can serve as examples for the glade/Gtk GUI programming with FreeBasic.

## Troubleshooting

If you are running into problems (especially on MS Windows) check the following:

- Do you have the latest Gtk version installed on your system?  
There's a trick: the latest Gtk version is always part of GIMP. So download GIMP from <http://www.gimp.org/> and install it. Afterwards copy (with replace) the directory C:\Program Files\GIMP-2.0\bin into your Gtk path, which should be C:\Program Files\Common Files\GTK\2.0\bin.
- Is the libglade-2.0-0.dll correctly installed (either in the application directory or in the \windows\system32 directory)?

Disclaimer:

Microsoft (MS) Windows and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

All other product and company names mentioned may be trademarks or registered trademarks of their owners.