Graphic User Interface FAQ

Contents

General GUI Questions

What platform-independent GUI toolkits exist for Python?

Depending on what platform(s) you are aiming at, there are several. Some of them haven't been ported to Python 3 yet. At least Tkinter and Qt are known to be Python 3-compatible.

Tkinter

Standard builds of Python include an object-oriented interface to the Tcl/Tk widget set, called tkinter. This is probably the easiest to install (since it comes included with most binary distributions of Python) and use. For more info about Tk, including pointers to the source, see the Tcl/Tk home page. Tcl/Tk is fully portable to the Mac OS X, Windows, and Unix platforms.

wxWidgets

wxWidgets (https://www.wxwidgets.org) is a free, portable GUI class library written in C++ that provides a native look and feel on a number of platforms, with Windows,

Mac OS X, GTK, X11, all listed as current stable targets. Language bindings are available for a number of languages including Python, Perl, Ruby, etc.

wxPython (http://www.wxpython.org) is the Python binding for wxwidgets. While it often lags slightly behind the official wxWidgets releases, it also offers a number of features via pure Python extensions that are not available in other language bindings. There is an active wxPython user and developer community.

Both wxWidgets and wxPython are free, open source, software with permissive licences that allow their use in commercial products as well as in freeware or shareware.

Qt

There are bindings available for the Qt toolkit (using either PyQt or PySide) and for KDE (PyKDE4). PyQt is currently more mature than PySide, but you must buy a PyQt license from Riverbank Computing if you want to write proprietary applications. PySide is free for all applications.

Qt 4.5 upwards is licensed under the LGPL license; also, commercial licenses are available from The Qt Company.

Gtk+

The GObject introspection bindings for Python allow you to write GTK+ 3 applications. There is also a Python GTK+ 3 Tutorial.

The older PyGtk bindings for the Gtk+ 2 toolkit have been implemented by James Henstridge; see <http://www.pygtk.org>.

Kivy

Kivy is a cross-platform GUI library supporting both desktop operating systems (Windows, macOS, Linux) and mobile devices (Android, iOS). It is written in Python and Cython, and can use a range of windowing backends.

Kivy is free and open source software distributed under the MIT license.

FLTK

Python bindings for the FLTK toolkit, a simple yet powerful and mature cross-platform windowing system, are available from the PyFLTK project.

OpenGL

For OpenGL bindings, see PyOpenGL.

What platform-specific GUI toolkits exist for Python?

By installing the PyObjc Objective-C bridge, Python programs can use Mac OS X's Cocoa libraries.

Pythonwin by Mark Hammond includes an interface to the Microsoft Foundation Classes and a Python programming environment that's written mostly in Python using the MFC classes.

Tkinter questions

How do I freeze Tkinter applications?

Freeze is a tool to create stand-alone applications. When freezing Tkinter applications, the applications will not be truly stand-alone, as the application will still need the Tcl and Tk libraries.

One solution is to ship the application with the Tcl and Tk libraries, and point to them at run-time using the `TCL_LIBRARY` and `TK_LIBRARY` environment variables.

To get truly stand-alone applications, the Tcl scripts that form the library have to be integrated into the application as well. One tool supporting that is SAM (stand-alone modules), which is part of the Tix distribution (http://tix.sourceforge.net/).

Build Tix with SAM enabled, perform the appropriate call to `Tclsam_init()`, etc. inside Python's `Modules/tkappinit.c`, and link with libtclsam and libtksam (you might include the Tix libraries as well).

Can I have Tk events handled while waiting for I/O?

On platforms other than Windows, yes, and you don't even need threads! But you'll have to restructure your I/O code a bit. Tk has the equivalent of Xt's `XtAddInput()` call, which allows you to register a callback function which will be called from the Tk mainloop when I/O is possible on a file descriptor. See File Handlers.

I can’t get key bindings to work in Tkinter: why?

An often-heard complaint is that event handlers bound to events with the `bind()` method don’t get handled even when the appropriate key is pressed.

The most common cause is that the widget to which the binding applies doesn’t have “keyboard focus”. Check out the Tk documentation for the focus command. Usually a widget is given the keyboard focus by clicking in it (but not for labels; see the takefocus option).