

# Porgy build instructions

## ***Dependencies list***

### **Required**

CMake

Qt4

Freetype

Zlib

Glew

libjpeg

libpng

libxml2

Boost

### **Optional**

Python

## ***Basic compilation and installation***

These are generic installation instructions. Tulip uses CMake as its main build system. The simplest way to compile this package is:

1. ``cd`` to the directory containing the package's source code, and create a ``build`` directory. Enter this ``build`` folder, and type ``cmake ..``. This will check for all of Tulip's dependencies, and warn if some are missing. If you want to build with debug information, or with optimisations, you need to tell CMake to do so. You can either use ``cmake -i``, which will ask you what kind of build you want (Debug/Release), or invoke cmake like this:  
``cmake .. -DCMAKE_BUILD_TYPE=Release`` (or Debug).
2. Type ``make`` to compile the package. (if you have more than one core, use ``-jX`` where X is your number of cores + 1.
3. Type ``make install`` to install the programs and any data files.
4. You can remove the program binaries and object files from the source code directory by typing ``make clean``.

## ***Compilers and Options***

You can pass options to cmake, either by using ``cmake .. -i`` or ``cmake-gui ..``, which will provide you with some control over the build.

You can change the build type (Debug Vs. Release) and compilation options.

By default, ``make install`` will install the package's files in ``/usr/local/bin``, ``/usr/local/man``, etc. You can specify an installation prefix other than ``/usr/local`` by giving cmake' the option ``-DCMAKE_INSTALL_PREFIX=PATH``.