

A brief CMake readme for CS 148

Stu Black

Wed Sep 09 16:56:53 2009

You have probably struggled or will struggle with `make`, a build system associated with Unix. `Make` does its job well, but it can be quite a trial to get accustomed to its way of declaring how things get done. (Amongst other charming idiosyncrasies, it's whitespace-sensitive.) Additionally, there are many different ways to get something set up quickly and easily in `make`, only to have your project outgrow the way you set it up back when it was only three files in one directory.

CMake tries to get around some of `make`'s shortcomings by providing an easy way of declaring how your project is compiled. From the `cmake` manpage:

CMake is a cross-platform build system generator. Projects specify their build process with platform-independent CMake listfiles included in each directory of a source tree with the name `CMakeLists.txt`. Users build a project by using CMake to generate a build system for a native tool on their platform.

On the CS machines, CMake builds makefiles by default. A sample session with CMake might look something like this:

1. `emacs CMakeLists.txt`
2. `cmake .`
3. `make`

In step 1, you edit your project's `CMakeLists.txt`. Basic `CMakeLists.txt` are provided for you in your group's Subversion repositories. They are liberally commented to show how to state which files should be built as part of your project.

In step 2, you run `cmake`, which reads your `CMakeLists.txt` and builds your makefiles. In addition to `Makefile`, the actual makefile, CMake generates `CMakeCache.txt`, `cmake_install.cmake`, and the directory `CMakeFiles`. You can ignore these.

In step 3, you run `make`, which reads the makefiles written by `cmake` and builds your project. Once you have done steps 1 and 2, you don't need to do them again unless you add or remove files from your project or decide to change how it is built in some other way (such as by changing your compiler options).

CMake's home on the Web is <http://www.cmake.org/>
Extensive documentation can be found at one of:

- `man cmake`
- <http://www.cmake.org/cmake/help/cmake2.6docs.html>

Special help may be found by searching the CMake mailing list archive at:

- <http://www.cmake.org/pipermail/cmake/>

You can also ask your TAs about problems you encounter.

1 What if I don't want to use CMake?

If you don't want to use CMake, you can use some other build system. We strongly recommend using one of CMake or GNU make. Do not use a build system that isn't installed on the CS machines, as your instructors will have to compile your code.

If you want to use plain-vanilla make, we have provided a basic Makefile for you in `/course/cs148/asgn`. You will need `Makefile`, `Makefile.rules`, and `Makefile.def`.