

Makefiles on Xcode

José Coelho - jose.alberto.coelho@gmail.com

Although Xcode keeps track of build settings in its own preferences files for information beyond what could normally be maintained in a makefile, it can also work closely with your project's makefiles. If you want to use Xcode for development in Mac OS X, you can include a makefile in a Xcode project as follows:

1. Launch Xcode.
2. Choose New Project from the File menu.

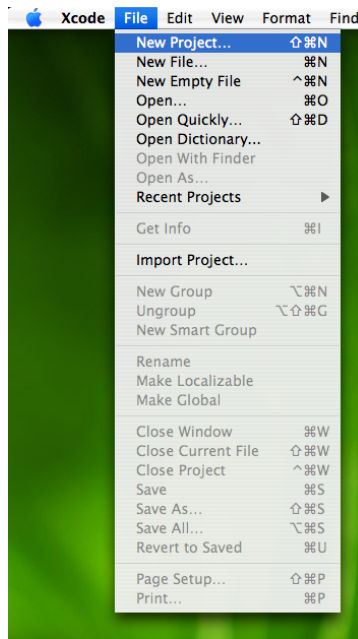


Figure 1: New Project

3. Select whatever project type you are targeting. If you ultimately want an application, select something like Cocoa Application. If you are just trying to build a command-line utility, select one of the tools—for example, Standard Tool.

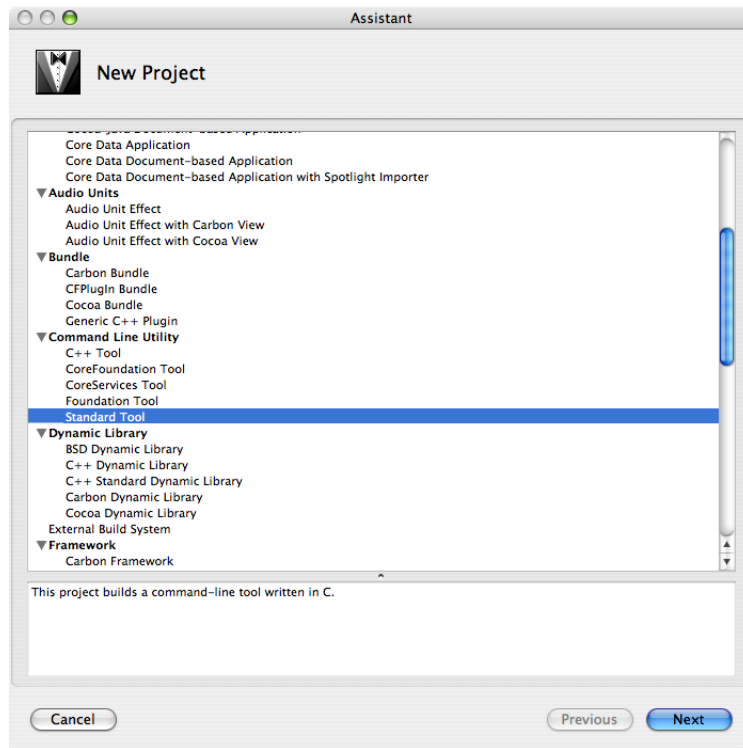


Figure 2: Standard Tool

4. Follow the prompts to name and save your project. A new default project of that type is opened.

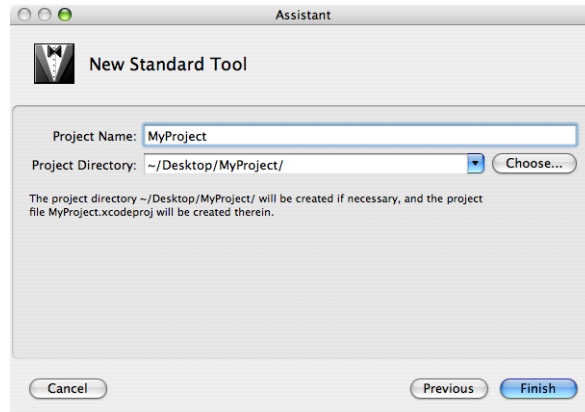


Figure 3: Example name

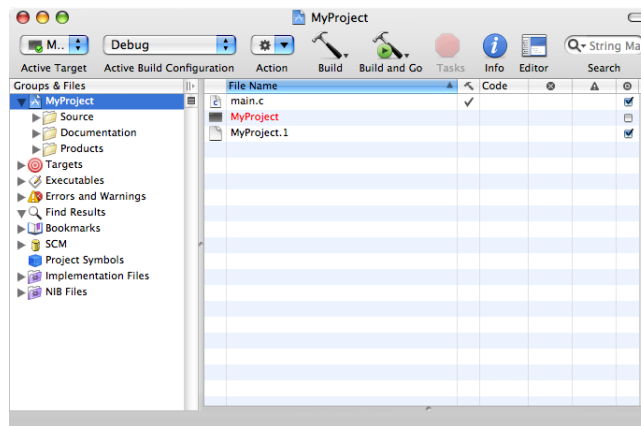


Figure 4: Default project

5. From the Project menu, Choose New Target.

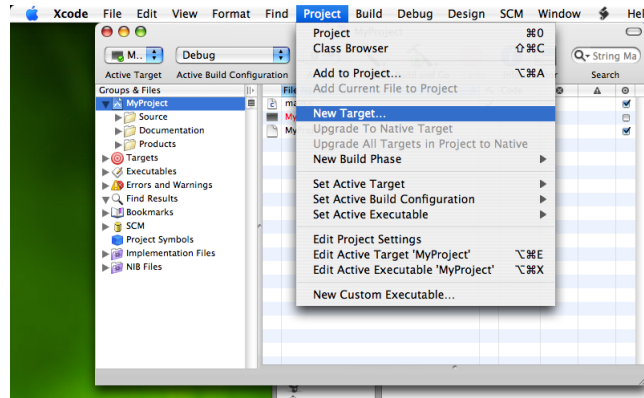


Figure 5: New target

6. Select “External Target” from the list. If this is not shown in the “Special Targets” list, you are not running the latest version of Xcode. Upgrade first.

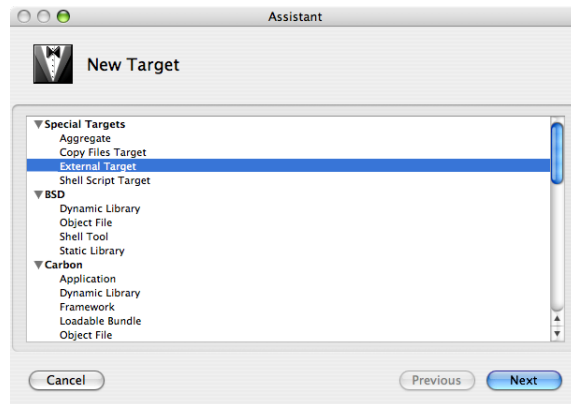


Figure 6: External Target

7. Follow the prompts to name that target. When you have done this, a target icon with the name you just gave it appears in the Targets pane of the open Xcode window.

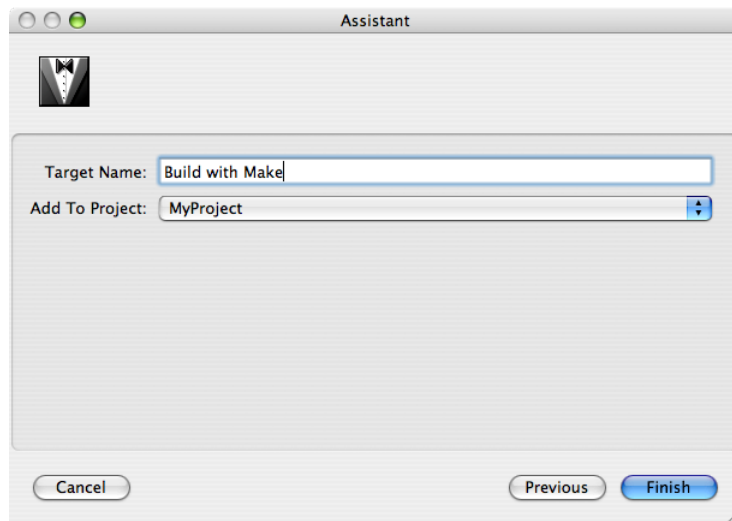


Figure 7: Example name

8. Double-click that new target. You should now see a new window with the build information for this target. This is not the same thing as clicking info. You must double-click the target itself.

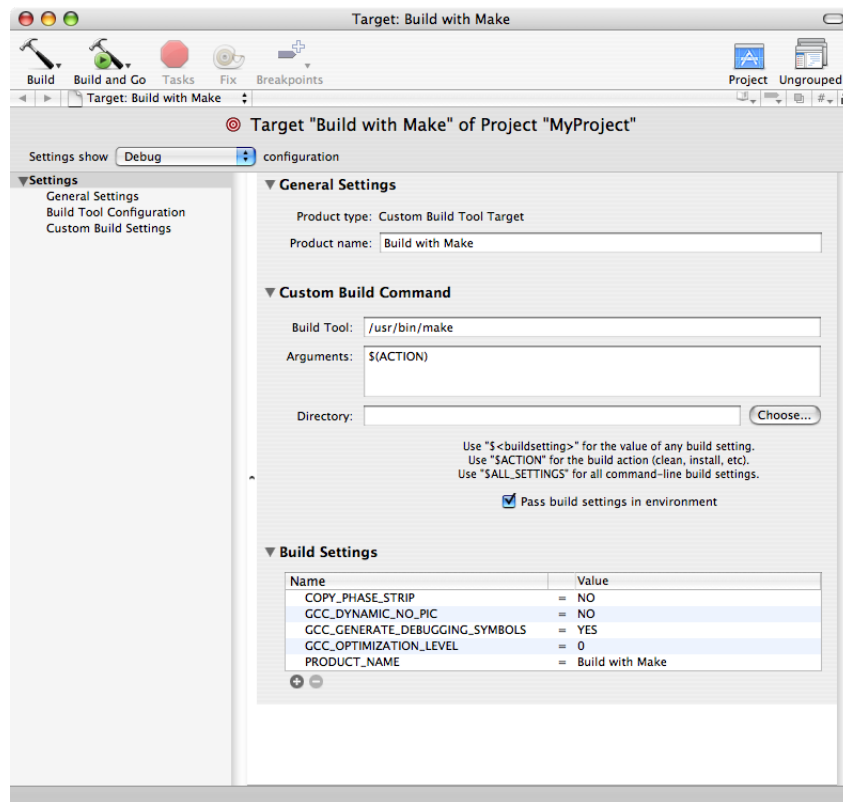


Figure 8: Target settings

9. Change the directory to point to the directory containing your makefile, and change any other settings as needed. For example, in the Custom Build Settings pane, you could change Build Tool from `/usr/bin/gnumake` to `/usr/bin/bsdmake`. More information on the fields is available in Xcode Help.

10. Change the active target to your new target by choosing “Set Active Target” from the Project menu. You may delete the old target.

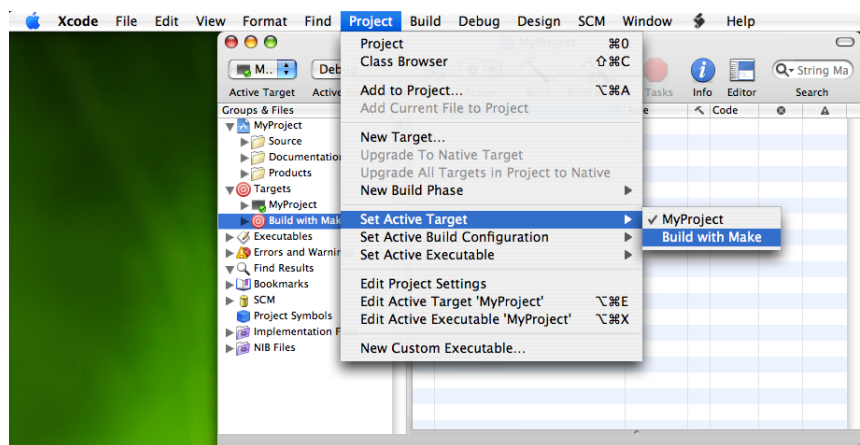


Figure 9: Set active target

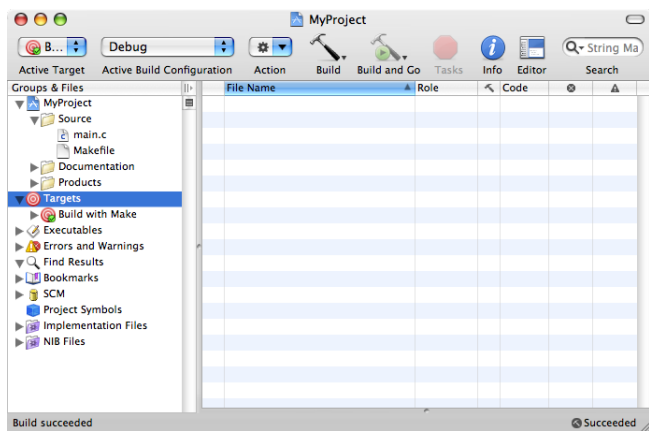


Figure 10: Old target deleted

11. Add the source files to the project. To do this, first open the disclosure triangle beside the “Source” folder in the left side of the project window. Next, drag the folder containing the sources from the Finder into that “Source” folder in Xcode. Tell Xcode not to copy files. Xcode will recursively find all of the files in that folder. Delete anything you don’t want listed.

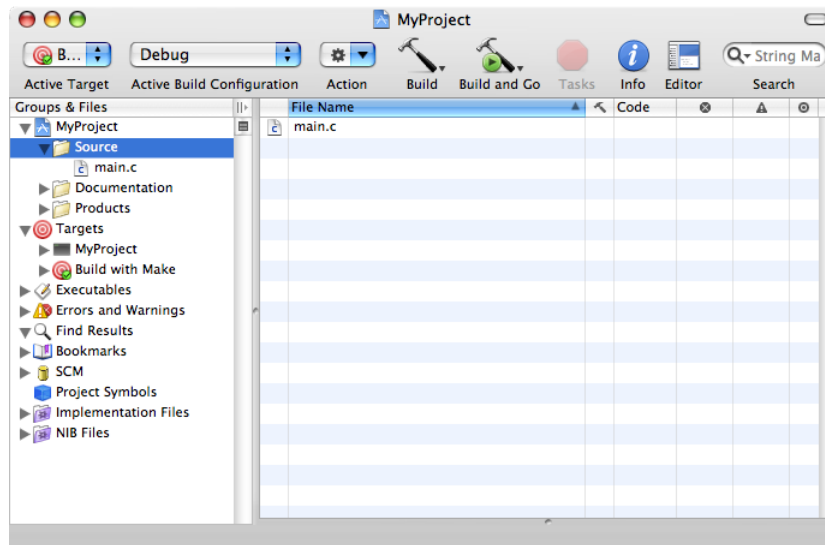


Figure 11: Add source files

12. Add a Makefile to the project. To do this, right click on “Source” folder. Add new file.

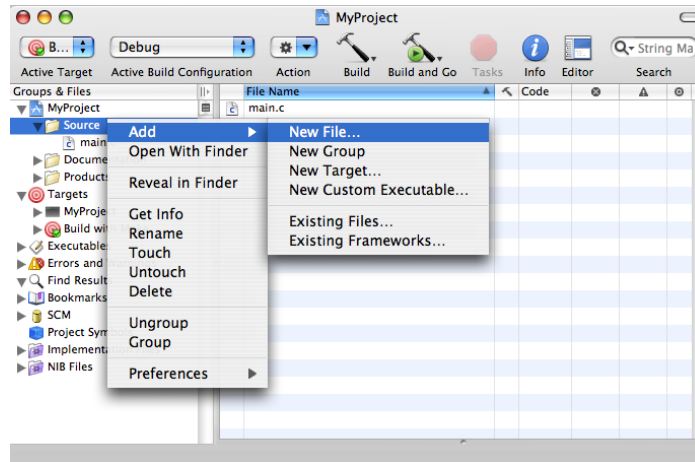


Figure 12: Add source files

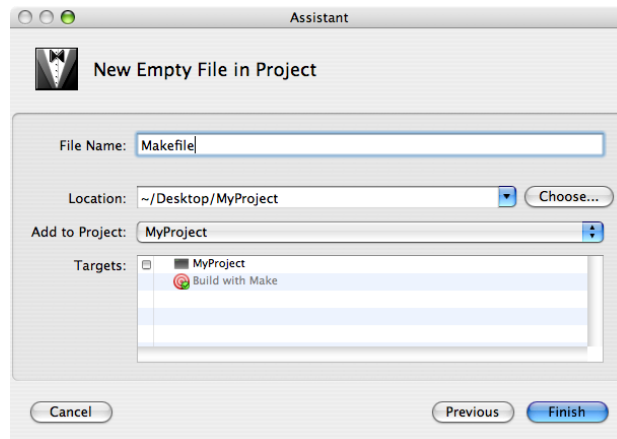


Figure 13: Creating a Make file

13. Add rules to the makefile.

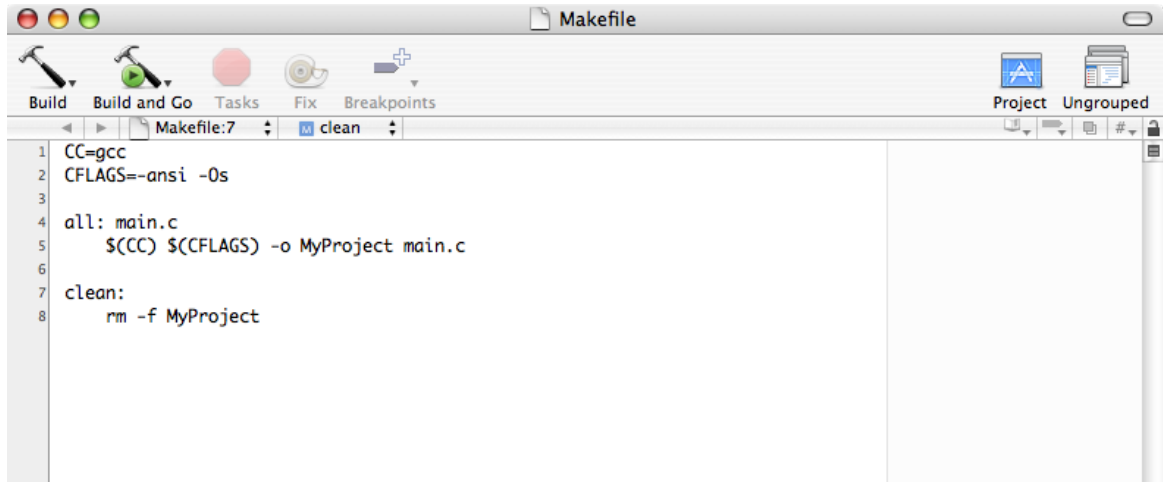


Figure 14: Example makefile

14. Tell Xcode where to find the executable by choosing “New Custom Executable” from the Project menu. Choose the path where the executable is located, then add the name of the executable.

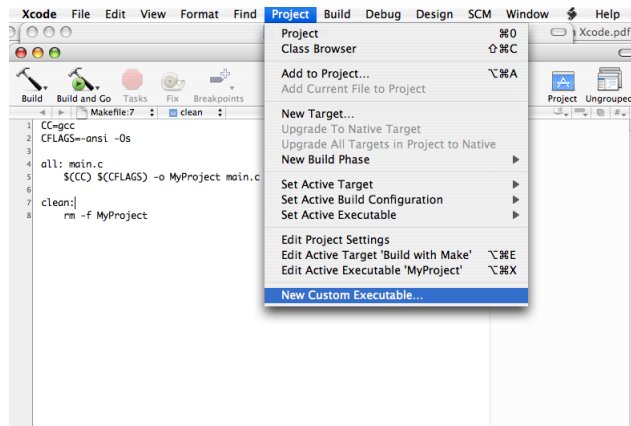


Figure 15: New Custom Executable

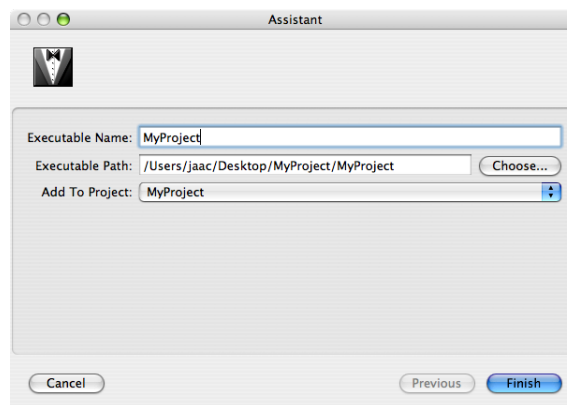


Figure 16: Executable name as on makefile

15. It's possible to customize multiple runtime options, as working directory, executable arguments.

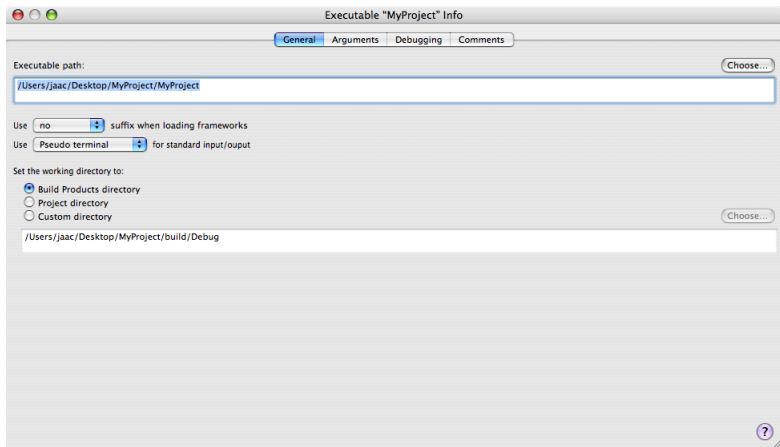


Figure 17: Miscellaneous options

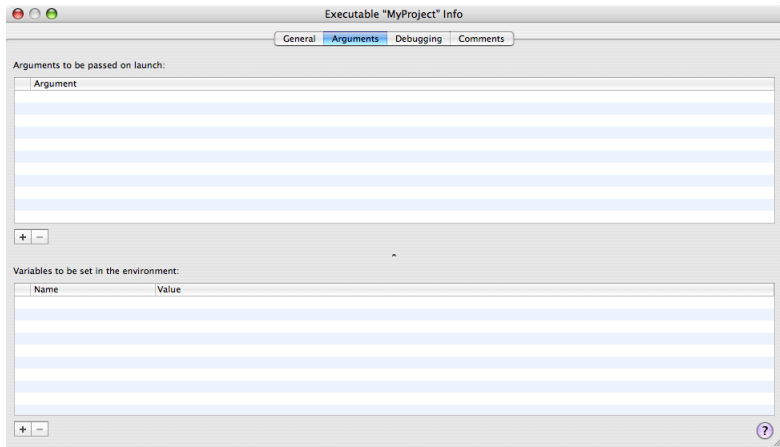


Figure 18: Run arguments

16. Now you are ready to build the project, click the Build and Run button in the toolbar, select Build from the Build menu, or just press Command-B.

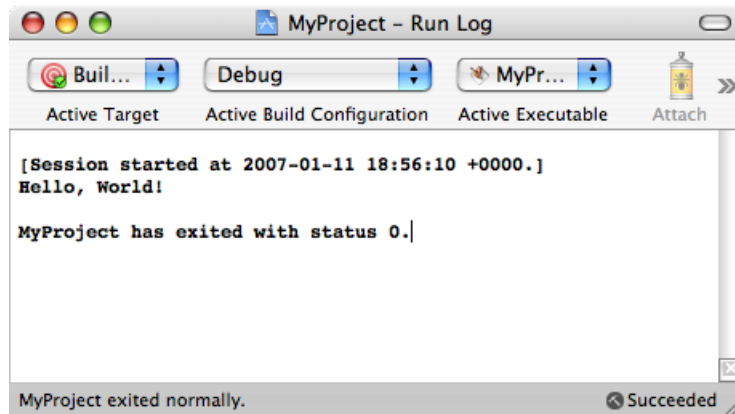


Figure 19: Run log - “Hello, World!”