Guide to DVD Chapter 24 Examples: *Jaeho Chang*

Converting Command-line Applications into Xcode Projects

In this folder, you will find two examples in support of my DVD Chapter for my DVD *Converting Command-line Applications into Xcode Projects.* These examples show you how to import a source file or files into an Xcode project, and to edit, compile and run your code using Xcode.

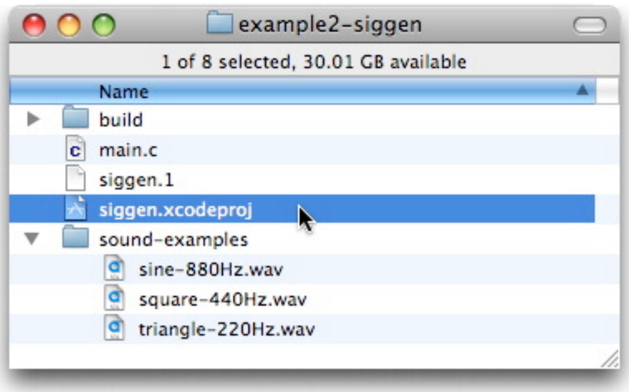
Installing Xcode

Xcode is a full-featured integrated development environment (IDE) that is a part of Mac OS X. You can install the Xcode that is included on your Mac OS X Install DVD; or you can download and install the latest version of Xcode from the Apple Developer Connection site at:

http://developer.apple.com/tools/xcode/

Opening a Project

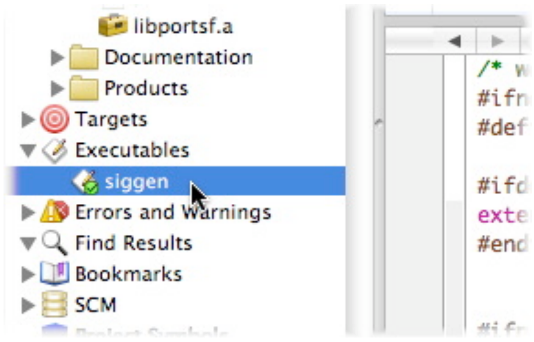
An Xcode project is not a single file; rather, it consists of a number of files. To open a project, you should locate a file with the *.xcodeproj* extension in the project folder and double-click on it.



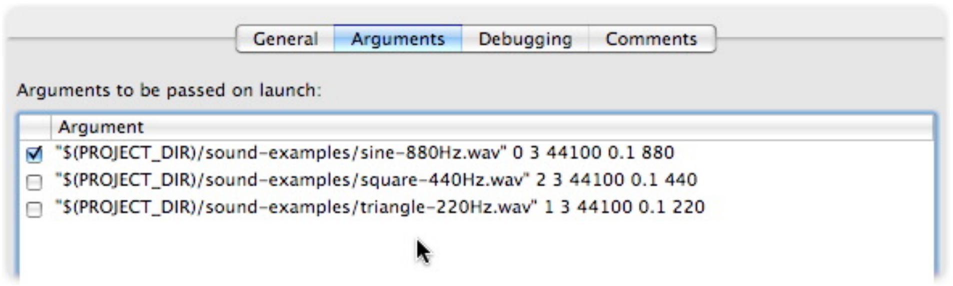
Supplying Arguments

Before running a compiled project, you can still supply command-line arguments to the executable of the project (except for the included *wonder.c* project because it does not take any arguments). This can be done with a few clicks as follows:

1. Click the triangle button beside the *Executables* group in the *Groups & Files* list, and then double-click the executable file.



2. In the *Executables* inspector, click the *Arguments* tab. There are example argument settings below that you can use to test this application. You can choose one by using the checkbox on the left side. After reading the text, you may want to add your own argument settings in this window to explore the example. Note: in some cases your program's arguments might require that you specify the complete path for an input or output soundfile, and that pathname could be quite long. In Xcode, the string "`$(PROJECT_DIR)/`" is an abbreviation (or shortcut) meaning "the project directory".

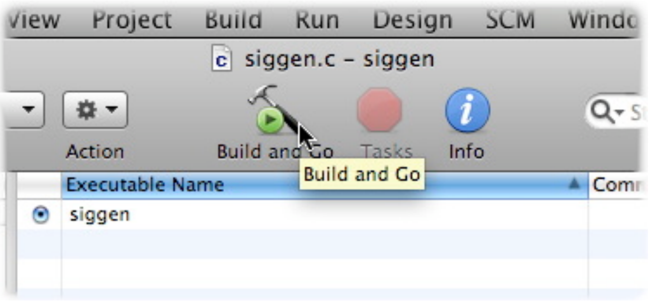


Arguments to be passed on launch:

	Argument
✓	"$(PROJECT_DIR)/sound-examples/sine-880Hz.wav" 0 3 44100 0.1 880
☐	"$(PROJECT_DIR)/sound-examples/square-440Hz.wav" 2 3 44100 0.1 440
☐	"$(PROJECT_DIR)/sound-examples/triangle-220Hz.wav" 1 3 44100 0.1 220

Running a Project

To compile and run your program, either click the *Build and Go* button on the toolbar, or choose *Build and Go* or *Build and Run* from the Build menu.



Nothing happens? Please remember that the program you just built is not really a *Cocoa Application* with a nice GUI; it is still a command-line program that has a text-based interface. To control and monitor a text-based or command-line application, you have to use *Console*. Choose *Console* from the *Run* menu and you will now see the result of Build and Go you just did.

In Xcode, typically, you will be using graphical buttons and menu items to control something, and so it is worth noting that *Console* is not fully featured, but just for "monitoring" what the system or application is doing.

About the '*common*' and '*extra*' Folders

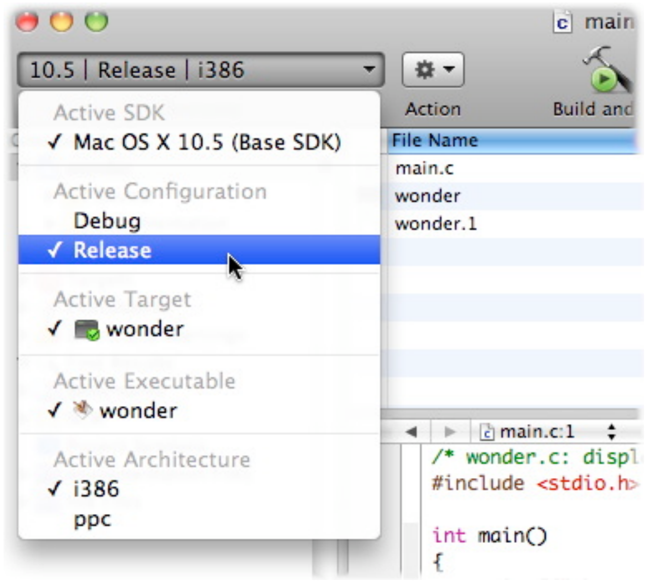
The *common* folder contains headers, sources, and libraries that are common to all applications. The *extra* folder contains the project for building the *portsf* library under Mac OS X. To work with these examples, you do not need to deal with these, but please read the DVD chapter text to understand what these are.

Where is the Executable File?

You may be particularly interested in this chapter because you would prefer to work in *Xcode* rather than in the *Terminal*. However, when the "build" is done in Xcode, by may be left wondering: "Where is the executable file?"

The answer is that it depends on the *build configuration* you chose. A "build configuration" is a named collection of build settings that you want to apply to your project. Without modification, a new Xcode project comes with two configurations: *Debug* and *Release*. One of the primary differences between these two is that for debug builds, you want to compile things as quickly as possible, while for release builds, you want to get the product optimized for performance.

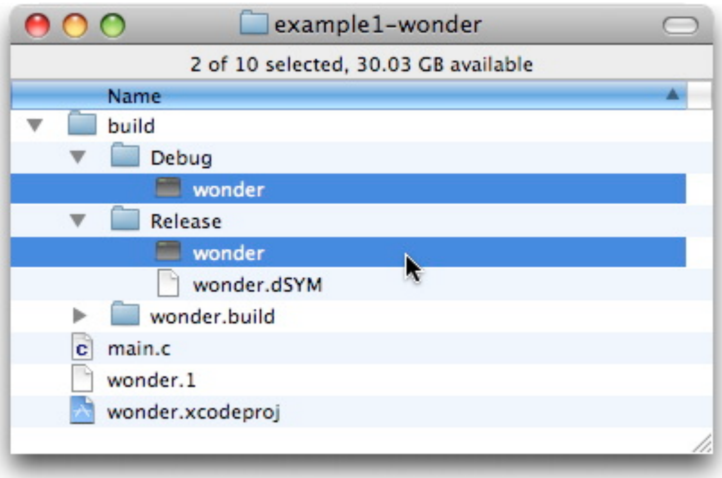
The default build configuration is Debug, but you can change this as shown below:



```
/* wonder.c: displ
#include <stdio.h>

int main()
{
```

If you executed *Build* or *Build and Go*, you can find the executable files in the *build* folder of your project. For example, when I built the project using both configurations, I get the executable as shown below:



Now you might want to try to run the executable in the Terminal. This is sometimes better, especially given that you do not need to launch Xcode every time you want to run your program, and so, once “built” or “compiled”, you may copy and move the executable to anywhere you might want – on this hard drive or any other!