

# Building Portaudio for Mac OS X

## Requirements

OS X 10.4 or later. PortAudio v19 currently only compiles and runs on OS X version 10.4 or later. Because of its heavy reliance on memory barriers, it's not clear how easy it would be to back-port PortAudio to OS X version 10.3. Leopard support requires the 2007 snapshot or later.

Apple's Xcode and its related tools installed in the default location. There is no Xcode project for PortAudio.

Mac 10.4 SDK. Look for `/Developer/SDKs/MacOSX10.4u.sdk` folder on your system. It may be installed with XCode. If not then you can download it from Apple Developer Connection. <http://connect.apple.com/>

## Building

To build PortAudio, simply use the Unix-style `./configure && make`:

```
./configure && make
```

You do **not** need to do "make install", and we don't recommend it; however, you may be using software that instructs you to do so, in which case you should follow those instructions. (Note from Phil: I had to do "sudo make install" after the command above, otherwise XCode complained that it could not find `/usr/local/lib/libportaudio.dylib` when I compiled an example.)

The result of these steps will be a file named `libportaudio.dylib` in the directory `usr/local/lib/`.

By default, this will create universal binaries and therefore requires the Universal SDK from Apple, included with XCode 2.1 and higher.

## .4 Building With Debug Options

There are a variety of other options for building PortAudio. The default described above is recommended as it is the most supported and tested; however, your needs may differ and require other options, which are described below.

By default, PortAudio is built as a universal binary. This includes 64-bit versions if you are compiling on 10.5, Leopard. If you want a "thin", or single architecture library, you have two options:

build a non-universal library using configure options. use `lipo(1)` on whatever part of the library you plan to use.

Note that the first option may require an extremely recent version of PortAudio (February 5th '08 at least).

To build a non-universal library for the host architecture, simply use the `--disable-mac-universal` option with configure.

```
./configure --disable-mac-universal && make
```

The `--disable-mac-universal` option may also be used in conjunction with environment variables to give you more control over the universal binary build process. For example, to build a universal binary for the i386 and ppc architectures using the 10.4u sdk (which is the default on 10.4, but not 10.5), you might specify this configure command line:

```
CFLAGS="-O2 -g -Wall -arch i386 -arch ppc -isysroot /Developer/SDKs/MacOSX10.4u.sdk -mmacosx-version-min=10.3" \  
LDFLAGS="-arch i386 -arch ppc -isysroot /Developer/SDKs/MacOSX10.4u.sdk -mmacosx-version-min=10.3" \  
./configure --disable-mac-universal --disable-dependency-tracking
```

For more info, see Apple's documentation on the matter:

<http://developer.apple.com/technotes/tn2005/tn2137.html>

[http://developer.apple.com/documentation/Porting/Conceptual/PortingUnix/intro/chapter\\_1\\_section\\_1.html](http://developer.apple.com/documentation/Porting/Conceptual/PortingUnix/intro/chapter_1_section_1.html)

The second option is to build normally, and use `lipo` (1) to extract the architectures you want. For example, if you want a "thin", i386 library only:

```
lipo lib/.libs/libportaudio.a -thin i386 -output libportaudio.a
```

or if you want to extract a single architecture fat file:

```
lipo lib/.libs/libportaudio.a -extract i386 -output libportaudio.a
```

By default, PortAudio on the mac is built without any debugging options. This is because asserts are generally inappropriate for a production environment and debugging information has been suspected, though not proven, to cause trouble with some interfaces. If you would like to compile with debugging, you must run configure with the appropriate flags. For example:

```
./configure --enable-mac-debug && make
```

This will enable `-g` and disable `-DNDEBUG` which will effectively enable asserts.

## Using the Library in XCode Projects

If you are planning to follow the rest of the tutorial, several project types will work. You can create a "Standard Tool" under "Command Line Utility". If you are not following the rest of the tutorial, any type of project should work with PortAudio, but these instructions may not work perfectly.

Once you've compiled PortAudio, the easiest and recommended way to use PortAudio in your XCode project is to add "`<portaudio>/include/portaudio.h`" and "`<portaudio>/lib/.libs/libportaudio.a`" to your project. Because "`<portaudio>/lib/.libs/`" is a hidden directory, you won't be able to navigate to it using the finder or the standard Mac OS file dialogs by clicking on files and folders. You can use command-shift-G in the finder to specify the exact path, or, from the shell, if you are in the portaudio directory, you can enter this command:

```
open lib/.libs
```

Then drag the "libportaudio.a" file into your XCode project and place it in the "External Frameworks and Libraries" group, if the project type has it. If not you can simply add it to the top level folder of the project.

You will need to add the following frameworks to your XCode project:

- CoreAudio.framework
- AudioToolbox.framework
- AudioUnit.framework
- CoreServices.framework
- Carbon.framework

## Using the Library in Other Projects

For gcc/Make style projects, include "include/portaudio.h" and link "libportaudio.a", and use the frameworks listed in the previous section. How you do so depends on your build.

## Using Mac-only Extensions to PortAudio

For additional, Mac-only extensions to the PortAudio interface, you may also want to grab "include/pa\_mac\_core.h". This file contains some special, mac-only features relating to sample-rate conversion, channel mapping, performance and device hogging. See "src/hostapi/coreaudio/notes.txt" for more details on these features.

## What Happened to Makefile.darwin?

Note, there used to be a special makefile just for darwin. This is no longer supported because you can build universal binaries from the standard configure routine. If you find this file in your directory structure it means you have an outdated version of PortAudio.

```
make -f Makefile.darwin
```

Back to the Tutorial: [PortAudio Tutorials](#)