

Get started with PS Move Api on MacOS X

The PS Move API is a cross-platform C library for the Sony Playstation Move Motion Controller. It includes bindings for Python, Java, Qt, Processing and others.

The library is available for Mac OS X, Windows and Linux (USB and Bluetooth).

This tutorial will guide you through the installation process on MacOS X (10.6 and up) and explain how to set up a PS Move API project in Xcode. I tried to keep it really straightforward and simple to follow. If you're new to this, don't be put off by all the command line stuff. Just follow along and you'll be fine.

Updates

News about the API are posted there: <http://thp.io/2010/psmove/>

Sources are on Github: <https://github.com/thp/psmoveapi>

The latest stable revision of the present tutorial can be downloaded there: <http://goo.gl/B6FaL>

Or read the working copy on Google docs: <http://goo.gl/cahbj> (if you see me editing, say hi!)

I need help!

Subscribe to the PS Move API mailing list: <https://lists.ims.tuwien.ac.at/mailman/listinfo/psmove>

I want to help!

Subscribe to the PS Move API mailing list: <https://lists.ims.tuwien.ac.at/mailman/listinfo/psmove>

I want to help but I cannot code!

No problem! Offer Thomas a beer :) <http://flattr.com/thing/147084/PS-Move-API>

There's an error in the tutorial!

OMG! Please write about it in the mailing list (copy the title of this tutorial in the title of your mail). I'll correct the mistake as soon as I can.

1) Installation and pairing

You will need...

... a PS Move controller (obviously)

... a mini-usb cable

... to create a root password: <http://youtu.be/10sKJQuXq6A>

... to install Homebrew: <http://mxcl.github.com/homebrew/> (see step 0.1)

... to install cmake, and git from Homebrew (see step 0.2)

Step 0.1

Open a Terminal window (the Terminal.app is located in `/Applications/Utilities/`), type in:

```
ruby -e "$(curl -fsSkL raw.githubusercontent.com/mxcl/homebrew/go)"
```

and press [enter].

Step 0.2

Likewise...

```
brew install cmake git
```

Optional

- Download and install [Doxygen](#) for mac (to generate documentation)

Installing and pairing instructions

Step 1

Connect the controller via USB.
Enable Bluetooth in MacOS.

The following steps will be done in the Terminal.

Step 2

We open the folder where we want the API installed. For the purpose of this tutorial:

```
cd /Developer/Library/
```

Note : the API will install in a sub-folder named `/psmoveapi/`.

Step 3

We download the source code for the PS Move API and external libraries:

```
git clone git://github.com/thp/psmoveapi.git
cd psmoveapi
git submodule init
git submodule update
```

Step 4

Build external libraries included in the source:

```
bash -e -x contrib/build-osx-snapshot
```

Step 5

Wait for the script to finish. We then create the “`build`” folder and open it:

```
mkdir build
cd build
```

Step 6

We configure the build using the `ccmake` command (careful: two c's):

```
ccmake ..
```

```
Page 1 of 1
APPKIT /System/Library/Frameworks/AppKit.framework
AVFOUNDATION /System/Library/Frameworks/AVFoundation.framework
CMAKE_BUILD_TYPE
CMAKE_INSTALL_PREFIX /opt/psmoveapi_2.0.0~2012-10-04+bb814dbd
CMAKE_OSX_ARCHITECTURES
CMAKE_OSX_DEPLOYMENT_TARGET
CMAKE_OSX_SYSROOT /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/
COREFOUNDATION /System/Library/Frameworks/CoreFoundation.framework
FOUNDATION /System/Library/Frameworks/Foundation.framework
IOBLUETOOTH /System/Library/Frameworks/IOBluetooth.framework
IOKIT /System/Library/Frameworks/IOKit.framework
OpenCV_DIR OpenCV_DIR-NOTFOUND
PSMOVE_BUILD_EXAMPLES ON
PSMOVE_BUILD_JAVA_BINDINGS ON
PSMOVE_BUILD_PROCESSING_BINDIN ON
PSMOVE_BUILD_PYTHON_BINDINGS ON
PSMOVE_BUILD_QT_BINDINGS OFF
PSMOVE_BUILD_TESTS ON
PSMOVE_BUILD_TRACKER ON
PSMOVE_BUILD_TUIO_SERVER ON
PSMOVE_USE_CL_EYE_SDK OFF
PSMOVE_USE_DEBUG OFF
PSMOVE_USE_LOCAL_OPENCV ON
PSMOVE_USE_PSEYE ON
PSMOVE_USE_TRACKER_TRACE ON
QTKIT /System/Library/Frameworks/QtKit.framework
QUARTZCORE /System/Library/Frameworks/QuartzCore.framework
SWIG_DIR /usr/local/Cellar/swig/2.0.8/share/swig/2.0.8
SWIG_EXECUTABLE /usr/local/bin/swig
SWIG_VERSION 2.0.8

APPKIT: Path to a library.
Press [enter] to edit option CMake Version 2.8.9
Press [c] to configure Press [g] to generate and exit
Press [h] for help Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

To modify the building configuration, navigate with the up and down arrows of your keyboard and press [enter] to edit an option.

If you want to use the API with [Processing](#), activate `PSMOVE_BUILD_JAVA_BINDINGS` and `PSMOVE_BUILD_PROCESSING_BINDIN`

To get camera tracking, activate:

`PSMOVE_BUILD_TRACKER` and `PSMOVE_USE_TRACKER_TRACE`

Then press [c] to configure, then [e] to come back to the configuration screen and finally [g] to generate the configuration settings (if the option does not show up on the first try, retry [c] then [e]).

Step 7

We are now back to the regular shell command. Type in:

```
cmake ..
make -j4
./example
```

The controller vibrates and the sphere blinks showing that the installation was successful.

Step 8

We now proceed to the pairing of the controller so that the system recognizes it as a bluetooth device.

```
./psmovepair
```

Note: Some MacOS 10.7 users have trouble with the automated pairing. If this is your case, see the “Manual pairing” instructions below and come back here when you’re done.

Note 2: Mountain Lion (Mac OS X 10.8) introduced changes related to bluetooth. A fix is underway. You can get more info in the comments of this [blog post](http://gutefabrik.com/blog/?p=1843) by Doug Wilson (<http://gutefabrik.com/blog/?p=1843>) or asking on the mailing list.

Step 9

Once the pairing is done, press the PS button on the controller and wait until the LED stops blinking.

Click on the Bluetooth icon and you should now see a line reading “**Motion Controller**”.

Open the “**example**” file in the /build folder and enjoy the flow of data!

Updating

The API gets frequent updates. Be sure to use the latest version. This is how to update:

```
cd /Developer/Library/psmoveapi
git pull
cd build
cmake ..
make -j4
```

Manual Pairing

On some versions of MacOS 10.7, automated pairing may not work, but fear not! I’ll show you how you can pair your controller manually.

Step 8.1

Right after you run `./psmovepair` write down the adress you find after “`controller address:`” in the form “`aa:bb:cc:dd:ee:ff`”

Step 8.2

Check that the controller can't connect. Press the PS button, you get a PIN entry:



Don't type in anything and close the pin entry (click on "Reject").

Step 8.3

Disable Bluetooth (or the modifications that follow won't work).

Step 8.4

In the terminal, run :

```
pgrep blued
```

again and again until it does not print anything anymore (e.g. the process "blued" has quit) - this can take a few seconds up to a minute.

Step 8.5

Type in the following (this is just one line). Replace "aa-bb-cc-dd-ee-ff" by the controller address you wrote down earlier (note: lowercase and hyphen "-" as a separator).

```
sudo defaults write /Library/Preferences/com.apple.Bluetooth HIDDevices -  
array-add "aa-bb-cc-dd-ee-ff"
```

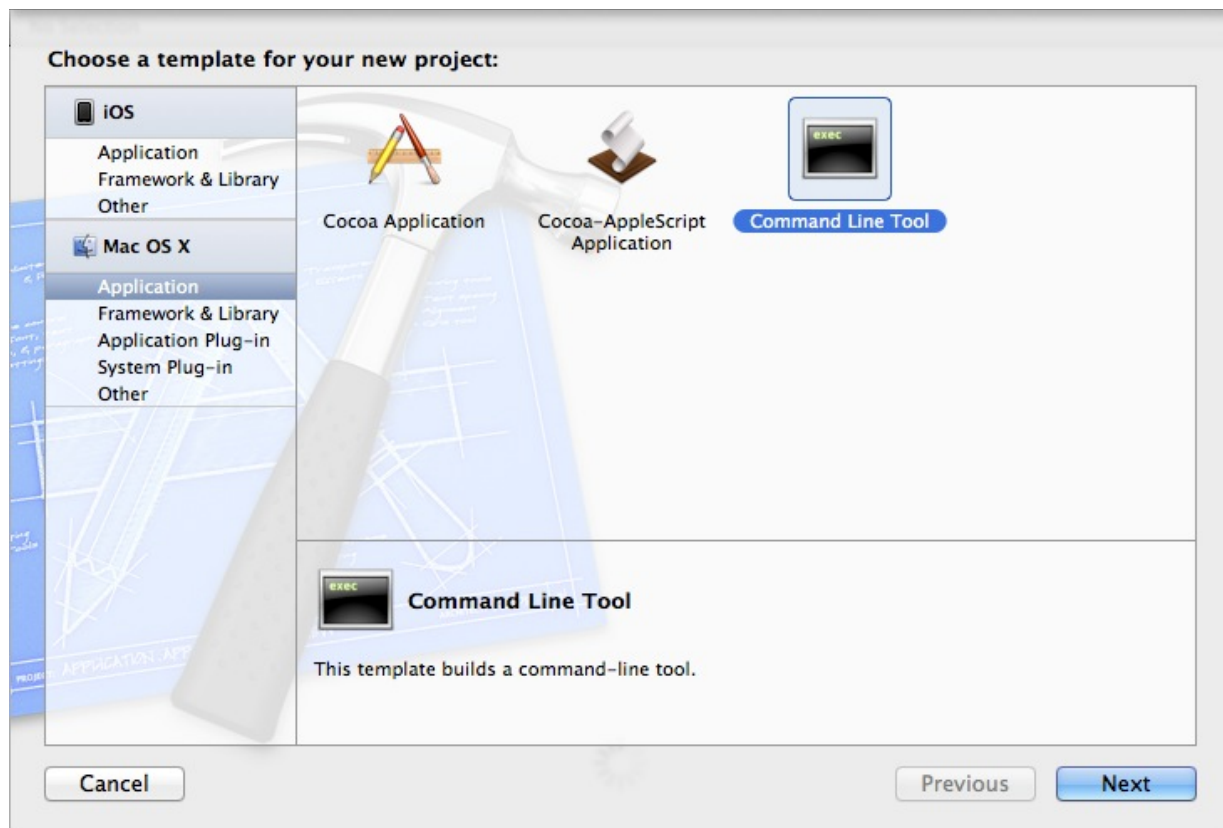
Step 8.6

Enable Bluetooth again.

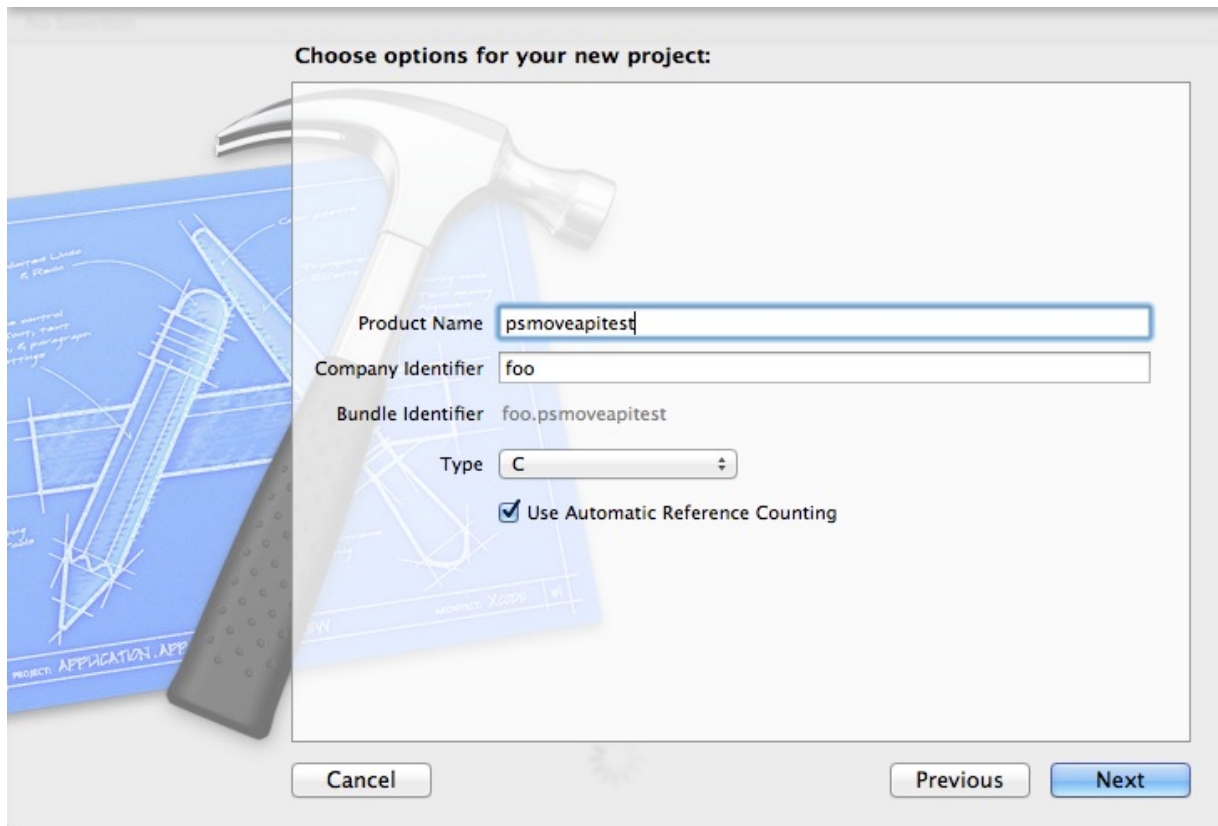
2) Creating a C project linked to the API in Xcode

Open Xcode.

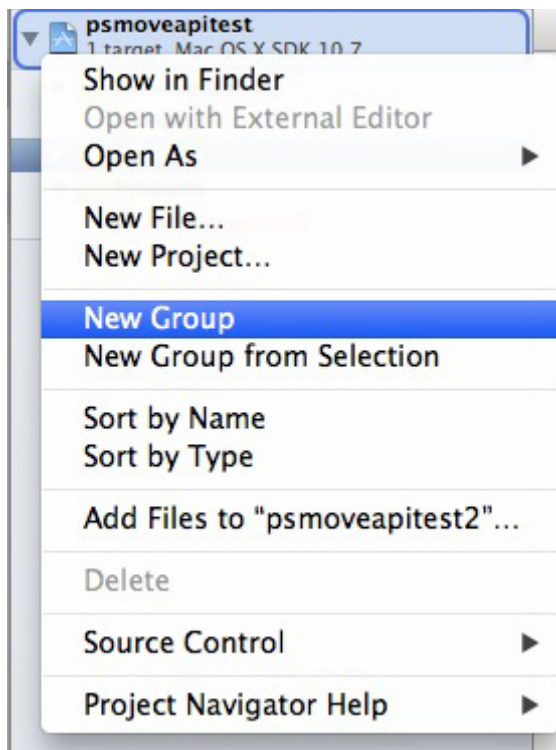
Go to [File>New>New Project...](#)



Under Mac OS X, click on “[Application](#)”.
From the list of templates, choose “[Command Line Tool](#)” and click “[Next](#)”.

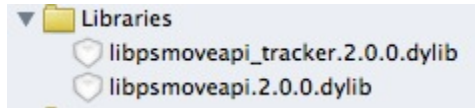


Give your project any name you want. For example “**psmoveapitest**”.
Chose the language of the project from the drop-down menu : **C**



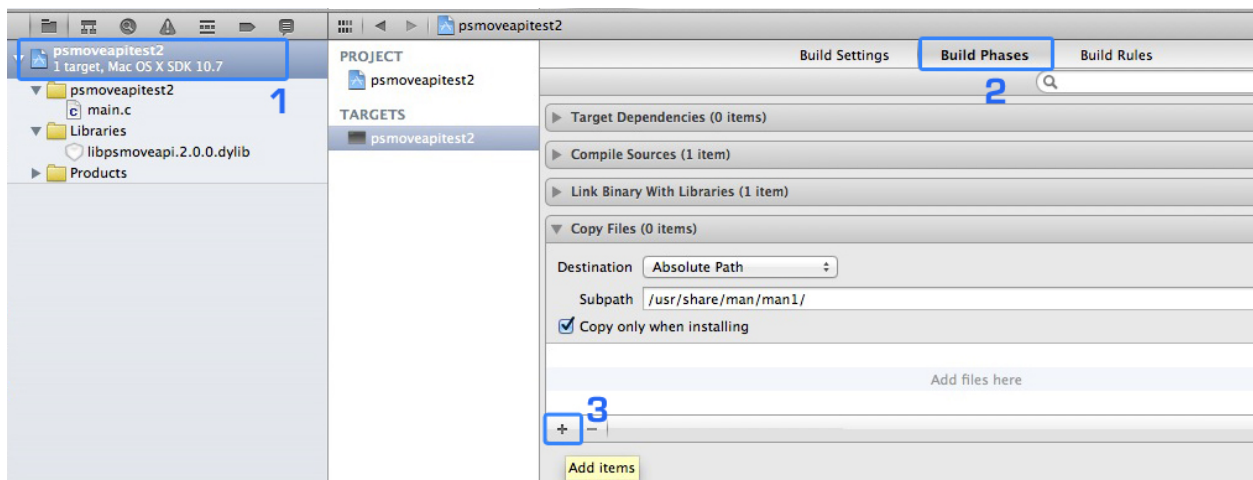
Create a new group in the left panel and name it "Libraries" for example

In the finder, open the /psmoveapi folder and look in /build for two dynamic library files named "libpsmoveapi.2.0.0.dylib" and "libpsmoveapi_tracker.2.0.0"



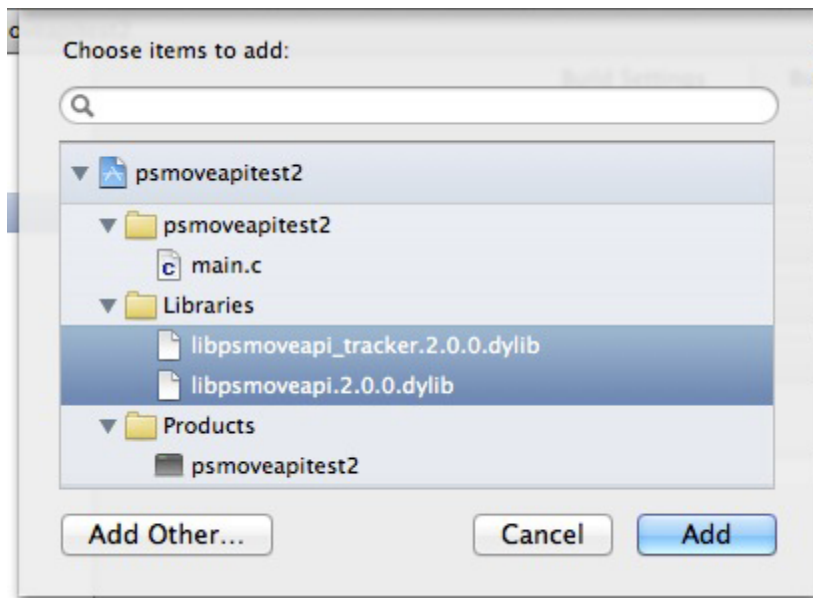
Drag the files to the "Libraries" folder you created in Xcode.

In the confirmation window, **do not** tick the "Copy items into destination's group folder" option and click "Finish"

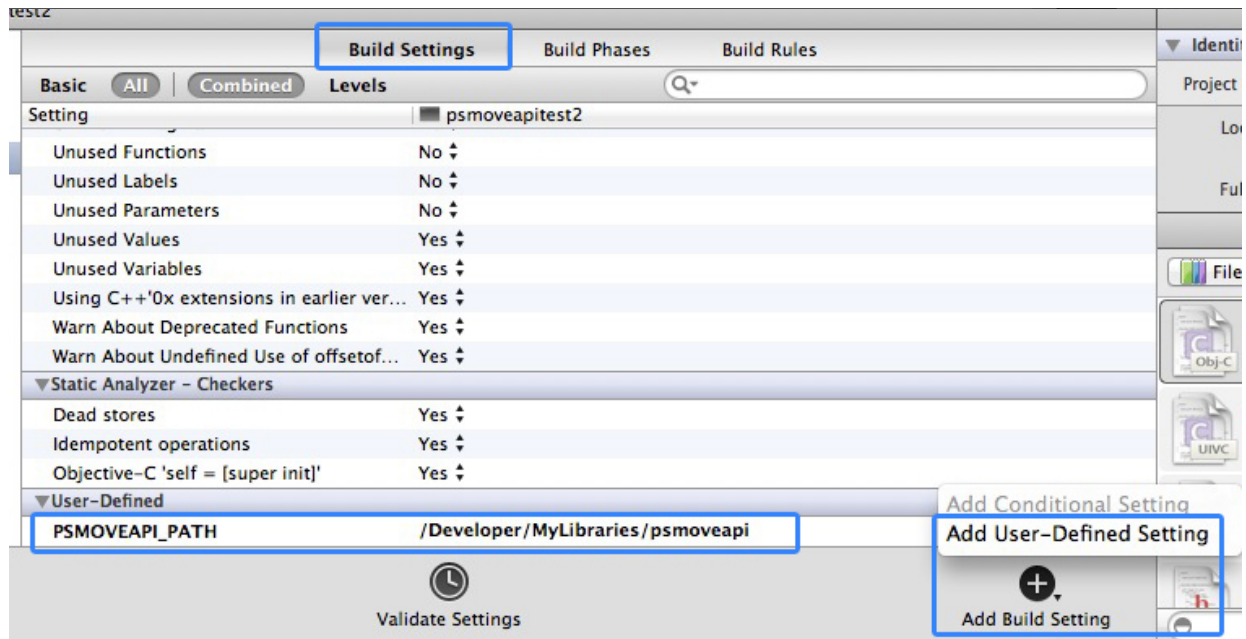


Click on the project name and find the "Build Phases" tab.

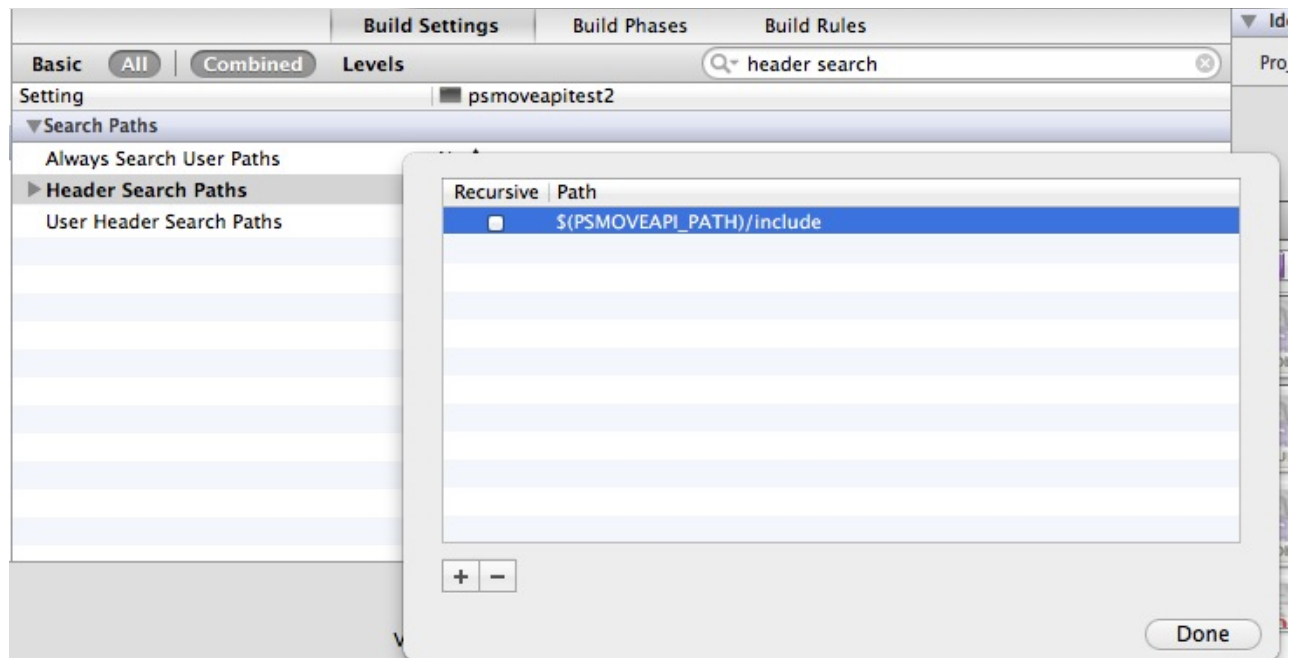
Click on the triangle next to "Copy Files" then on the + sign at the bottom left.



A window opens and asks to "Choose items to add"
Select all the dylibs in "Libraries" then click "Add".



Now click on the "Build Settings" tab.
Click on "Add Build Settings" (in earlier versions of Xcode, click on the gear icon at the bottom left)
Then click on "Add User-Defined Setting".
Type in "PSMOVEAPI_PATH" (without the quotes) for the setting's name and the path of the folder where you installed the lib. In my case : /Developer/MyLibraries/psmoveapi



In the search field, type "header search"

Add "\$(PSMOVEAPI_PATH)/include" to the "Header Search Paths" setting.
If you're going to use OpenCV, also add "\$(PSMOVEAPI_PATH)/opencv/build/install/include".
For TUIO, add "\$(PSMOVEAPI_PATH)/external/TUIO_CPP" and select "recursive".
Click "Done". (in Xcode 4.5, just click outside of the popup window)

In the "main.c", add the lines

```
#include "psmove.h"  
#include "psmove_tracker.h"
```

Build and run to check everything is ok.

You can start by pasting the contents of `example.c` or `tracker.c` over the contents of your `main.c` and compile it then play around with the code.

You're done!

Get some inspiration from the files in the `/psmoveapi/examples/c/` folder on how to use the API and read the doc for details on the methods included. Some of the examples will require a little tweaking in Xcode before they compile. I'll update this tutorial as soon as I figure it out. Feel free to share your experience on the mailing list (<https://lists.ims.tuwien.ac.at/mailman/listinfo/psmove>).