

# **JVSIP Environment Fundamentals**

**Version 0.81**

**March 4, 2015**

© 2015 Randall Judd, all rights reserved.

A non-exclusive, non-royalty bearing license is hereby granted to all persons to copy, modify, distribute and produce derivative works for any purpose, provided that this copyright notice and following disclaimer appear on all copies:

THIS LICENSE INCLUDES NO WARRANTIES, EXPRESSED OR IMPLIED, WHETHER ORAL OR WRITTEN, WITH RESPECT TO THE SOFTWARE OR OTHER MATERIAL INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF PERFORMANCE OR DEALING, OR FROM USAGE OR TRADE, OR OF NON-INFRINGEMENT OF ANY PATENTS OF THIRD PARTIES. THE INFORMATION IN THIS DOCUMENT SHOULD NOT BE CONSTRUED AS A COMMITMENT OF DEVELOPMENT BY ANY PARTY.

## Introduction

This document is included with a VSIPL distribution I have called Judd VSIP (JVSIP) to differentiate this VSIPL implementation from other implementations.

In this document the environment needed to make and use the various JVSIP tools are introduced. Some pointers on installing the environment are here but the user needs to find complete instructions from other sources. There are too many ways to do things and I don't know (or understand) them all.

My primary environment resides on an Apple computer with OS X (10.8.4 as I write this) operating system. The fundamentals should be similar for other platforms although the install methods for the environment will differ.

What do I mean by environment? Well for instance to run python code you will need a python interpreter installed and to compile a C VSIPL library you will need a C compiler.

## Tools

I spent a fair amount of time installing or trying to install, using various methods, the tools I describe below. Installation locations, paths, version, files, libraries, etc. can get messy on your system. From time to time you need to step back and clean up old files and libraries.

## Compiler

You need a C compiler. The C code is generally written to conform to ANSI C 89 so any C Compiler should work. On OS X you should get Xcode from the Apple app store and install it. After installing Xcode you need to install the terminal tools. Instructions for doing this are readily available on the internet.

For the linux platform usually you can load compilers using **apt-get**

The default build mechanism in JVSIP for the C Library is **make** using the gnu make implementation. The library is archived using **ar** as a static library. If you happen to be on a system that does not support these tools then you may need to roll your own method. Note that you can compile (assuming your c compiler is fairly normal) all the source files to object files in the **c\_VSIP\_src** directory with the simple incantation

```
cc -c *.c -I./
```

so unless your system is very esoteric you should be able to figure out how to compile the code for a C VSIPL library suitable for your use.

## Python

You need python (unless you only want the C library in which case you can ignore the python stuff). Python can be troublesome although once you get things working everything will be fine. I currently am developing using python 2.7. There is also a python 3 but I have not tested with it.

Note that OS X comes with a python. You probably don't want to use that one so install something newer.

Make sure your PATH variable (at a terminal prompt > **echo \$PATH**) is set up so that the proper python is called.

## SWIG

The C code I supply must be wrapped with the python C API in order to be made into a python module. Wrapping all the code I have in JVSIP would be an enormous effort. Luckily a tool called Software Wrapper and Interface Generator exists. You need this. Find information at

<http://www.swig.org>

## Python Modules

There are several python modules that are handy to have. None are needed to run pyJvsip but to use the notebooks (files with extension **.ipynb**) scattered throughout the JVSIP distribution you will need ipython (see <http://ipython.org> ) and various supporting modules like numpy, scipy, and matplotlib.

## How to Install Tools

Note there are various ways to install things. I only cover the OS X platform here since that is the one I am familiar with.

To the degree possible (on a mac at least) you want to use an install method that is consistent across all the tools. Using `easy_install` for some stuff, `brew` for other stuff, `fink` and/or `macports` for other stuff, download and build directly for some stuff, etc. is a bad idea. I am still confused by why there is both a **pip** and an **easy\_install**.

The install tools I currently prefer are **brew** <http://mxcl.github.io/homebrew/> and **pip** (comes with python installation when installed with brew). The brew tool lets one install things like python, SWIG, gfortran, freetype, etc. Once python is installed you use pip to install python modules like ipython, numpy, scipy, etc. Currently on my OS X platform the only tools not installed by brew and pip that I use with JVSIP are text editors and Xcode.

I recommended the web site

<http://www.lowindata.com/2013/installing-scientific-python-on-mac-os-x/>

for instructions on what and how to install all the python tools I use. I won't reproduce their work here.

Note that you need to install brew first. I recommend checking by doing

**\$ brew update**

to see if you already have it. If you don't search on the web for instructions on how to install it. I think (from the site above)

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

should install brew.

Note brew puts everything in /usr/local . I think you can also have brew put things in other spots but I am not expert enough to discuss them. You can, for instance, download brew directly from the github site.

My last advice is to do the following

**Install brew first.** Then do

**\$ brew doctor**

and fix any problems. When all the problems are fixed, including getting rid of any warnings about your PATH variable, then do

**\$ brew install python**

Once you have the freshest python (make sure it is the first one in your path) follow along with the instructions available in the [Lowin Data](#) site.

After everything is built and installed you should be able to go into the JVSIP python directory and start the installation procedures there. If you have problems feel free to contact me.

## Additional Caveats

If your system is clean things should work pretty much as described above. If you like me and have installed some things previously then the previous installation might get in the way of brew or pip. Brew will suggest override mechanisms to fix problems; but pip is not so helpful.

If pip gives you problems installing something then first uninstall the package (pip uninstall *package*). Once I needed to uninstall twice since I had a package in two places and pip would not reinstall until I uninstalled the package at both places. Then do a pip install *package*. If it installs fine your good to go. If it fails to install generally there is an old file or directory in the way that pip can not overwrite. Remove these files (pip will tell you what is in the way). AGAIN do a pip uninstall. THEN do a pip install. Repeat until successful.

Note. To see what brew has installed use **brew list**. To see what pip has installed use **pip list**.

One of the reason I have gone to these two packages is to help with updating the packages as they are changed. For brew you do

**brew upgrade *package***

and for pip you do

**pip install --upgrade *package***.

Unfortunately pyJvsip is not supported by brew or pip. Maybe someday. Read document jvsip\_HowToMake.pdf which should be in the same directory this document is in.

## **Additional Links**

<http://ipython.org/install.html>

<http://mxcl.github.io/homebrew/>

[Installing Homebrew on OS X](#)