NASA - PDS Cassini Propeller Hunter C++ and Java Code Refactoring Assembly Deployment Guide

Revision History

Author	Revision Number	Date
yedtoss	1.0	03/29/2015

Deployment Instructions

1.Organization of Submission

2.Application Setup

3.Third Party Libraries

4.Compilation Instructions

- MAC OS X
- LINUX

5.Usage Instructions

- Run any of the binary with -h as option to see how to use it.
- Note that all options are optional, if you don't provide it default value will be used

6.Verification

- Porting all code to C++
- Separate in three programs
- Accuracy
- Radius and Longitude
- Command Line options and threading
- Logs and Error Messages
- Splitting code in Multiple files
- Speed Comparison

7.Resource Contact List

Deployment Instructions

1. Organization of Submission

docs/ Contains this deployment guides
src/ Contains the source code
test_files/ Contains test files

2. Application Setup

MAC OS X 10.10 64 bits
Linux/Ubuntu 14.04 64 bits
g++ 4.9
Cmake >= 2.8

g++ lower than that won't work

3. Third Party Libraries

Easyloggingcpp version 9.80 https://github.com/easylogging/easyloggingpp/releases
Tclap version 1.2.1 http://sourceforge.net/projects/tclap/files/
Cereal version 1.1.1 http://uscilab.github.io/cereal/
CSV commit aa9a4cc44b3e786a4e13a04c054a95c669df88ab https://github.com/jay/CSV

4. Compilation Instructions

MAC OS X

First we need to install g++ 4.9 then cmake and after that we can compile and get the binaries.

Here is how to do it

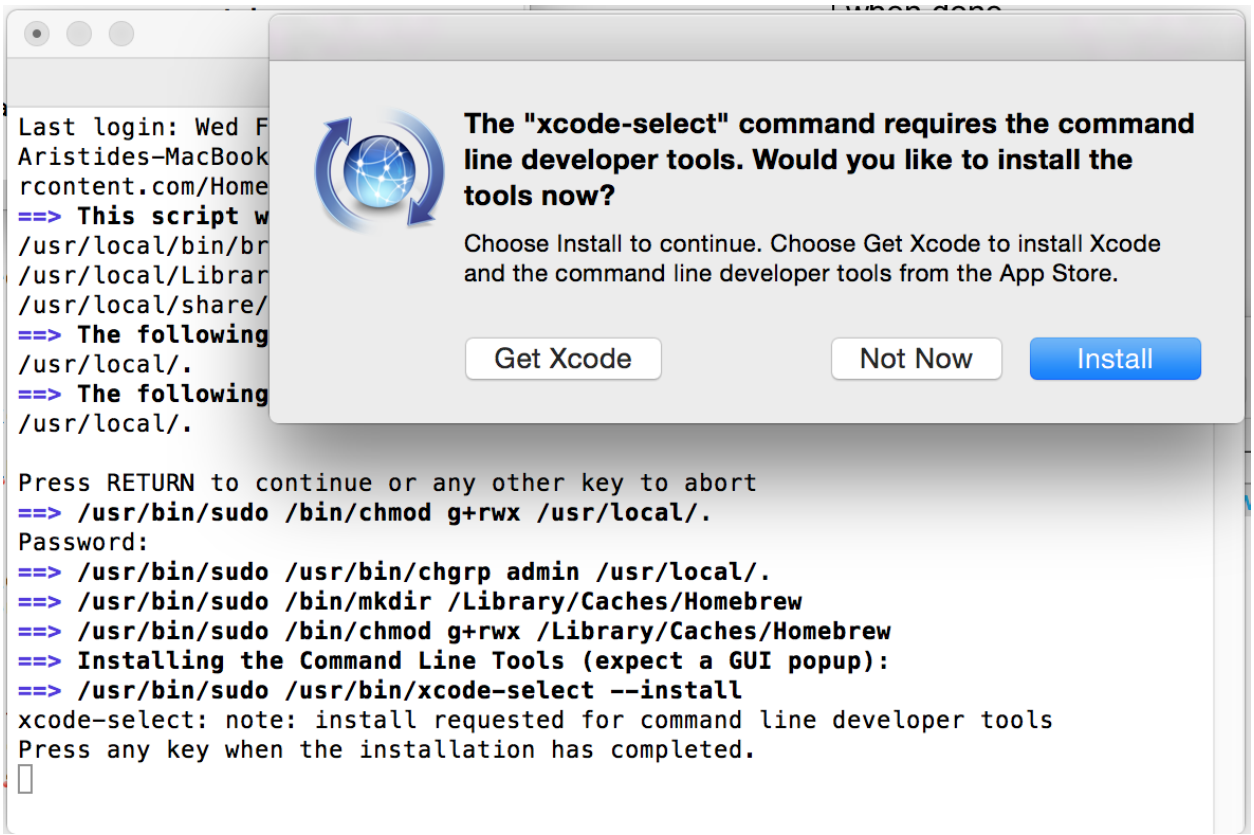
To deploy the application in OS X, perform the following step:

- Install brew: ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)".

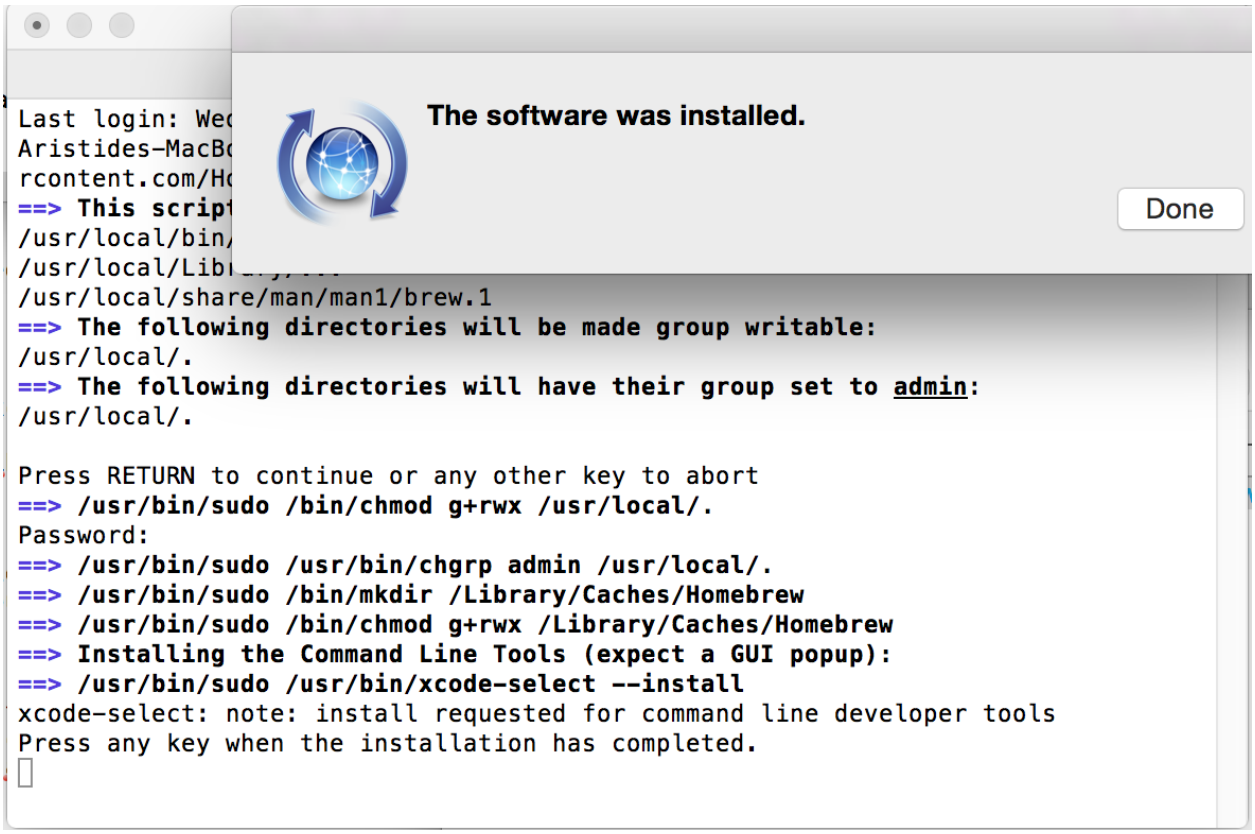
```
Aristides-MacBook-Pro:build yedtoss$ ruby -e "$(curl -fsSL https://raw.githubuse
rcontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/Library/...
/usr/local/share/man/man1/brew.1
==> The following directories will be made group writable:
/usr/local/.
==> The following directories will have their group set to admin:
/usr/local/.

Press RETURN to continue or any other key to abort
```

If you do not have the Xcode command-line tools, you will be prompted to install them as shown below.



Install the software and make sure the installation is successful as shown below.



Press any key. It is possible that you will see an error about the active developer path:

```
build — bash — 80×24
bash                                bash
/usr/local/.
==> The following directories will have their group set to admin:
/usr/local/.

Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/chmod g+rwx /usr/local/.
Password:
==> /usr/bin/sudo /usr/bin/chgrp admin /usr/local/.
==> /usr/bin/sudo /bin/mkdir /Library/Caches/Homebrew
==> /usr/bin/sudo /bin/chmod g+rwx /Library/Caches/Homebrew
==> Installing the Command Line Tools (expect a GUI popup):
==> /usr/bin/sudo /usr/bin/xcode-select --install
xcode-select: note: install requested for command line developer tools
Press any key when the installation has completed.
==> Downloading and installing Homebrew...
xcrun: error: active developer path ("/Volumes/Xcode/Xcode.app/Contents/Develope
r") does not exist, use `xcode-select --switch path/to/Xcode.app` to specify the
 Xcode that you wish to use for command line developer tools (or see `man xcode-
select`)
==> Installation successful!
==> Next steps
Run `brew doctor` before you install anything
Run `brew help` to get started
Aristides-MacBook-Pro:build yedtoss$
```

If you do get that error, you can correct it by specifying the version of Xcode to use:
sudo xcode-select --switch /Applications/Xcode.app

```
Aristides-MacBook-Pro:build yedtoss$ sudo xcode-select --switch /Applications/Xc
ode.app
Password:
Aristides-MacBook-Pro:build yedtoss$
```

Now run this command:
brew doctor

```
build — bash — 80×24
bash    bash    less
Aristides-MacBook-Pro:build yedtoss$
Aristides-MacBook-Pro:build yedtoss$
Aristides-MacBook-Pro:build yedtoss$
Aristides-MacBook-Pro:build yedtoss$
Aristides-MacBook-Pro:build yedtoss$
Aristides-MacBook-Pro:build yedtoss$
Aristides-MacBook-Pro:build yedtoss$
Aristides-MacBook-Pro:build yedtoss$ brew doctor
Please note that these warnings are just used to help the Homebrew maintainers
with debugging if you file an issue. If everything you use Homebrew for is
working fine: please don't worry and just ignore them. Thanks!

Warning: You have MacPorts or Fink installed:
  /opt/local/bin/port

This can cause trouble. You don't have to uninstall them, but you may want to
temporarily move them out of the way, e.g.

  sudo mv /opt/local ~/macports

Warning: Your Xcode (6.1) is outdated
Please update to Xcode 6.1.1.
Xcode can be updated from the App Store.
Aristides-MacBook-Pro:build yedtoss$
```

Ignore any warnings that may appear.
Install GCC 4.9:
brew install gcc
You can ignore warnings about multilib.

```
build — curl — 80×24
bash    curl    less
/usr/local/Cellar/gmp/6.0.0a: 15 files, 3.2M
==> Installing gcc dependency: mpfr
==> Downloading https://homebrew.bintray.com/bottles/mpfr-3.1.2-p10.yosemite.bot
################################################################################# 100.0%
==> Pouring mpfr-3.1.2-p10.yosemite.bottle.1.tar.gz
/usr/local/Cellar/mpfr/3.1.2-p10: 24 files, 3.5M
==> Installing gcc dependency: libmpc
==> Downloading https://homebrew.bintray.com/bottles/libmpc-1.0.3.yosemite.bottl
################################################################################# 100.0%
==> Pouring libmpc-1.0.3.yosemite.bottle.tar.gz
/usr/local/Cellar/libmpc/1.0.3: 10 files, 380K
==> Installing gcc dependency: isl
==> Downloading https://homebrew.bintray.com/bottles/isl-0.12.2.yosemite.bottle.
################################################################################# 100.0%
==> Pouring isl-0.12.2.yosemite.bottle.2.tar.gz
/usr/local/Cellar/isl/0.12.2: 55 files, 3.1M
==> Installing gcc dependency: cloog
==> Downloading https://homebrew.bintray.com/bottles/cloog-0.18.1.yosemite.bottl
################################################################################# 100.0%
==> Pouring cloog-0.18.1.yosemite.bottle.2.tar.gz
/usr/local/Cellar/cloog/0.18.1: 33 files, 560K
==> Installing gcc
==> Downloading https://homebrew.bintray.com/bottles/gcc-4.9.2_1.yosemite.bottle
##############################################                                  61.3%
```

Install CMake:
brew install cmake

- Now create a directory $BUILD outside the src/ directory of the submission and enter that directory
- Run cmake src -DCMAKE_CXX_COMPILER="g++-4.9"
- Run make
- You will get three binaries in the $BUILD directory named: PropellerTrain, PropellerTest and `PropellerStats`

LINUX

First we need to install g++ 4.9 then cmake and after that we can compile and get the binaries.
Here is how to do it for Ubuntu 14.04:

Run sudo add-apt-repository ppa:ubuntu-toolchain-r/test

Run sudo apt-get update

Run sudo apt-get install g++-4.9

Run sudo apt-get install cmake

- Now create a directory $BUILD outside the src/ directory of the submission and enter that directory
- Run cmake src -DCMAKE_CXX_COMPILER="g++-4.9"
- Run make
- You will get three binaries in the $BUILD directory named: PropellerTrain, PropellerTest and PropellerStats

5. Usage Instructions

Run any of the binary with -h as option to see how to use it.

Note that all options are optional, if you don't provide it default value will be used

It is expected to first run PropellerTrain then in second, PropellerTest and finally PropellerStats.

6. Verification

Porting all code to C++

It is easy to verify that all code have been ported to C++

Separate in three programs

It is easy to see that the code has been splited in three programs. Check their help to see how to use them

Accuracy

Run the previous code (Get it from contest specification and there is a README, Read also contest specification for where to put the testing files) like that:

java -jar PropellerDetector.jar ./PropellerDetector-1 quick_index.lbl quick_index2.lbl full_ground_truth.csv -v

```
Found 'Post' in N1717603842.
Found 'Post' in N1717603992.
Found 'Sikorsky' in N1717594542.
Found 'Post' in N1717604292.
Found 'Post' in N1717604142.
Found 'Kingsford Smith' in N1717594992.
Found 'Kingsford Smith' in N1717594692.
Linked 'Post' 4 times.
Linked 'Kingsford Smith' 2 times.
Correct detections: 7 / 10 - 1050 false positives.
positionScore: 144817.3661367136
linkingScore: 70000.0
```

Now run in order

PropellerTrain -t quick_index.lbl -g full_ground_truth.csv -m training_model.dat
PropellerTest -t quick_index2.lbl -g full_ground_truth.csv -m training_model.dat -o candidates.dat
PropellerStats -o candidates.dat

Run these commands from the test_files/ directory. And make sure to download and extract both
https://drive.google.com/file/d/0B-0bDf2WzZsgXzZVMGNJaGhIS28/view?usp=sharing

and https://drive.google.com/file/d/0B-0bDf2WzZsgUmt4NXlMTk5wdFU/view?usp=sharing

such that test_files/eval and test_files/contest directory exists.

You will notice that both previous code and current code give as result in the output:

```
2015-04-10 11:47:24,775 INFO  [default] Found 'Post' in N1717603842.

2015-04-10 11:47:24,775 INFO  [default] Found 'Post' in N1717603992.

2015-04-10 11:47:24,775 INFO  [default] Found 'Sikorsky' in N1717594542.

2015-04-10 11:47:24,776 INFO  [default] Found 'Post' in N1717604292.

2015-04-10 11:47:24,776 INFO  [default] Found 'Post' in N1717604142.

2015-04-10 11:47:24,784 INFO  [default] Found 'Kingsford Smith' in N1717594992.

2015-04-10 11:47:24,790 INFO  [default] Found 'Kingsford Smith' in N1717594692.
```

```
2015-04-10 11:48:48,225 INFO  [default] Linked 'Post' 4 times.

2015-04-10 11:48:48,225 INFO  [default] Linked 'Kingsford Smith' 2 times.

2015-04-10 11:48:48,225 INFO  [default] Correct detections: 7 / 10 - 1050 false
positives.

2015-04-10 11:48:48,225 INFO  [default] positionScore: 144817

2015-04-10 11:48:48,225 INFO  [default] linkingScore: 70000
```

All decimal are truncated in output.
You can also see those results in test_files/output_answer.txt after running all three programs.

Radius and Longitude

The radius and longitude are the two last columns of output in test_files/output_answer_continious.txt.
There are also available in test_files/output_answer.txt

Command Line options and threading

Refer to the respective help of each command and test each command line options.
Add the command line option “-c 4” to use 4 threads.
You should use the full system test to see a noticeable speed improvement over the non threaded version.

Logs and Error Messages

Check the files test_files/output_answer_continious.txt It contains propeller results that are continuously displayed

Also on the standard output, info messages, error messages and debug messages are continuously displayed. Messages in standard output are also available in logs/myeasylog.log

Splitting code in Multiple files

The code has been split in several files. There is a .h and .cpp when relevant to do so.

Speed Comparison

New code is faster than old code. You can run the command line options provided in accuracy subsection preceded by the command “time”.

In our testing we got the following for the previous code

```
real    1m11.445s
user    1m27.860s
sys     0m1.563s
```

For the new code we got respectively for PropellerTrain, PropellerTest and PropellerStats

```
real    0m24.501s
user    0m24.434s
sys     0m0.084s
```

```
real    0m23.463s
user    0m23.145s
sys     0m0.252s
```

```
real    0m0.004s
user    0m0.004s
sys     0m0.000s
```

7. Resource Contact List

Name	Resource Email
yedtoss	

