ASPECT INSTALLATION - UBUNTU 14.04

Phil Heron, Durham University
March 2017
email: philip.j.heron@durham.ac.uk

1. SETUP

ASPECT requires a number of packages and other libraries for installation.

A command to obtain the packages for Ubuntu 14.04 would be:

```
sudo apt-get install build-essential wget automake \
                   autoconf gfortran \
                   openmpi-bin openmpi-common \
                   libopenmpi-dev cmake subversion \
                   git libblas-dev liblapack-dev \
                   libblas3gf liblapack3gf \
                   libsuitesparse-dev libtool \
                   libboost-all-dev splint tcl \
                   tcl-dev environment-modules qt4-dev-tools
```

Libraries that would also be required are:

Trilinos
- performs linear algebra calculations

p4est
- builds distributed adaptive meshes in parallel

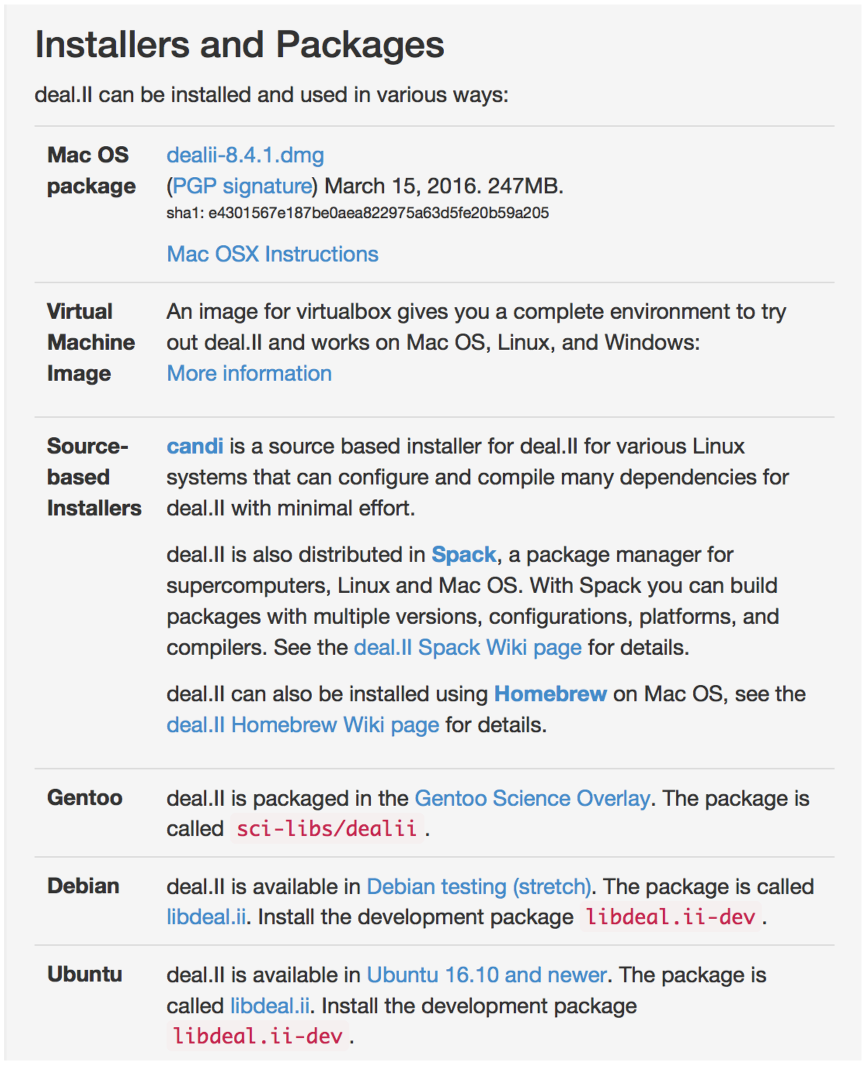
deal.ii
- a general purpose finite element library that communicates meshes to geometry, etc.

Other optional libraries include:

HDF5 - an additional output format

PETSC - an alternative solver (however, this installation caused a lot of problems during configuration, so I would avoid this for an ubuntu setup)

deal.ii is required to be built and linked with p4est and trilinos. deal.ii can be configured and built across a number of different platforms, as shown from the deal.ii website: http://dealii.org/download.html



Installers and Packages

deal.II can be installed and used in various ways:

Mac OS package	dealii-8.4.1.dmg (PGP signature) March 15, 2016. 247MB. sha1: e4301567e187be0aea822975a63d5fe20b59a205 Mac OSX Instructions
Virtual Machine Image	An image for virtualbox gives you a complete environment to try out deal.II and works on Mac OS, Linux, and Windows: More information
Source-based Installers	**candi** is a source based installer for deal.II for various Linux systems that can configure and compile many dependencies for deal.II with minimal effort. deal.II is also distributed in **Spack**, a package manager for supercomputers, Linux and Mac OS. With Spack you can build packages with multiple versions, configurations, platforms, and compilers. See the deal.II Spack Wiki page for details. deal.II can also be installed using **Homebrew** on Mac OS, see the deal.II Homebrew Wiki page for details.
Gentoo	deal.II is packaged in the Gentoo Science Overlay. The package is called `sci-libs/dealii`.
Debian	deal.II is available in Debian testing (stretch). The package is called libdeal.ii. Install the development package `libdeal.ii-dev`.
Ubuntu	deal.II is available in Ubuntu 16.10 and newer. The package is called libdeal.ii. Install the development package `libdeal.ii-dev`.

For Ubuntu, the Source-based installer **Candi** makes it easier to setup deal.ii (with all other dependencies). This is a great timesaver rather than manually building all the dependencies and linking together.

2. Candi

To learn about Candi, visit

https://github.com/koecher/candi

Before installation, make sure all packages are in place on your system:

```
sudo apt-get install build-essential wget automake \
                   autoconf gfortran \
                   openmpi-bin openmpi-common \
                   libopenmpi-dev cmake subversion \
                   git libblas-dev liblapack-dev \
                   libblas3gf liblapack3gf \
                   libsuitesparse-dev libtool \
                   libboost-all-dev splint tcl \
                   tcl-dev environment-modules qt4-dev-tools
```

In terminal, download Candi through git:

```
git clone https://github.com/dealii/candi
```

A number of modifications are required to the candi setup in order to configure all the libraries with deal.ii with the ultimate end point of running ASPECT.

```
cd candi
vi candi.cfg
```

In this file, the original will show:

```
#########################################################################
#Choose configuration and components of deal.II
DEAL_CONFOPTS=" \
-D DEAL_II_COMPONENT_PARAMETER_GUI=OFF \
"

PACKAGES="load:dealii-prepare"
PACKAGES="${PACKAGES} once:opencascade"
PACKAGES="${PACKAGES} once:parmetis"
#PACKAGES="${PACKAGES} once:superlu_dist"
PACKAGES="${PACKAGES} once:hdf5"
PACKAGES="${PACKAGES} once:p4est"
PACKAGES="${PACKAGES} once:trilinos"
PACKAGES="${PACKAGES} once:petsc"
PACKAGES="${PACKAGES} once:slepc"
PACKAGES="${PACKAGES} dealii"

```

This is the configuration setup which needs to be edited in order to match what is required of deal.ii for ASPECT. To do this, we need to explicitly say that deal.ii will work with:

MPI
threads
LAPACK
UMFPACK
BOOST
ZLIB
FUNCTIONPARSER
COMPONENT MESH CONVERTER

We also need to state that deal.ii will not be compiled with:

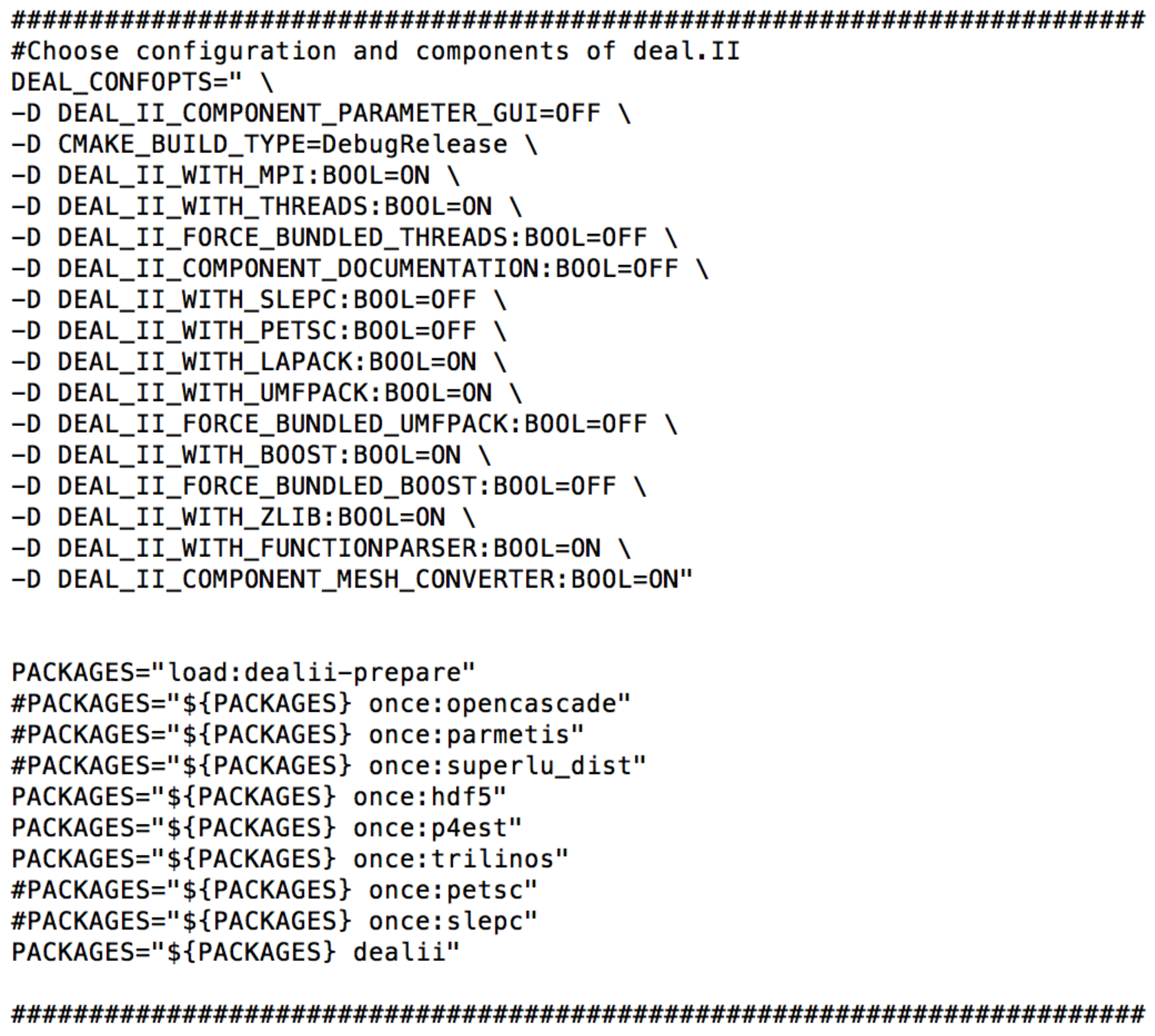
SLEPC
PETSC
BUNDLED THREADS
BUNDLED BOOST

Furthermore, as candi can download a great number of packages that deal.ii can work with, we only need to download the ones that are directly relevant to ASPECT. As a result, we can can comment out the packages that we do not need (which are open cascade, PETSC, SLEPC).

```
#Choose configuration and components of deal.II
DEAL_CONFOPTS=" \
-D DEAL_II_COMPONENT_PARAMETER_GUI=OFF \
-D CMAKE_BUILD_TYPE=DebugRelease \
-D DEAL_II_WITH_MPI:BOOL=ON \
-D DEAL_II_WITH_THREADS:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_THREADS:BOOL=OFF \
-D DEAL_II_COMPONENT_DOCUMENTATION:BOOL=OFF \
-D DEAL_II_WITH_SLEPC:BOOL=OFF \
-D DEAL_II_WITH_PETSC:BOOL=OFF \
-D DEAL_II_WITH_LAPACK:BOOL=ON \
-D DEAL_II_WITH_UMFPACK:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_UMFPACK:BOOL=OFF \
-D DEAL_II_WITH_BOOST:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_BOOST:BOOL=OFF \
-D DEAL_II_WITH_ZLIB:BOOL=ON \
-D DEAL_II_WITH_FUNCTIONPARSER:BOOL=ON \
-D DEAL_II_COMPONENT_MESH_CONVERTER:BOOL=ON"


PACKAGES="load:dealii-prepare"
#PACKAGES="${PACKAGES} once:opencascade"
```

```
#PACKAGES="${PACKAGES} once:parmetis"
#PACKAGES="${PACKAGES} once:superlu_dist"
PACKAGES="${PACKAGES} once:hdf5"
PACKAGES="${PACKAGES} once:p4est"
PACKAGES="${PACKAGES} once:trilinos"
#PACKAGES="${PACKAGES} once:petsc"
#PACKAGES="${PACKAGES} once:slepc"
PACKAGES="${PACKAGES} dealii"
```

In the file this would look like this:

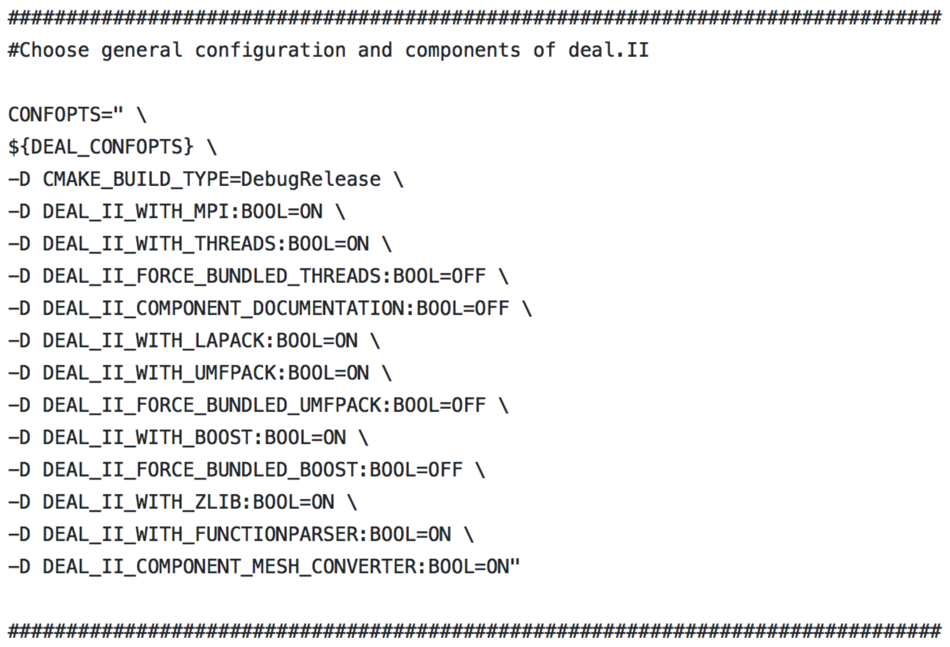


```
##########################################################################
#Choose configuration and components of deal.II
DEAL_CONFOPTS=" \
-D DEAL_II_COMPONENT_PARAMETER_GUI=OFF \
-D CMAKE_BUILD_TYPE=DebugRelease \
-D DEAL_II_WITH_MPI:BOOL=ON \
-D DEAL_II_WITH_THREADS:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_THREADS:BOOL=OFF \
-D DEAL_II_COMPONENT_DOCUMENTATION:BOOL=OFF \
-D DEAL_II_WITH_SLEPC:BOOL=OFF \
-D DEAL_II_WITH_PETSC:BOOL=OFF \
-D DEAL_II_WITH_LAPACK:BOOL=ON \
-D DEAL_II_WITH_UMFPACK:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_UMFPACK:BOOL=OFF \
-D DEAL_II_WITH_BOOST:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_BOOST:BOOL=OFF \
-D DEAL_II_WITH_ZLIB:BOOL=ON \
-D DEAL_II_WITH_FUNCTIONPARSER:BOOL=ON \
-D DEAL_II_COMPONENT_MESH_CONVERTER:BOOL=ON"


PACKAGES="load:dealii-prepare"
#PACKAGES="${PACKAGES} once:opencascade"
#PACKAGES="${PACKAGES} once:parmetis"
#PACKAGES="${PACKAGES} once:superlu_dist"
PACKAGES="${PACKAGES} once:hdf5"
PACKAGES="${PACKAGES} once:p4est"
PACKAGES="${PACKAGES} once:trilinos"
#PACKAGES="${PACKAGES} once:petsc"
#PACKAGES="${PACKAGES} once:slepc"
PACKAGES="${PACKAGES} dealii"

##########################################################################
```

Next we need to edit the deal.ii package setup in the individual configuration file for deal.ii (rather than the global one for candi that we just edited).

```
cd deal.II-toolchain/

cd packages
vi dealii.package
```

Here we edit this:



```
################################################################################
#Choose general configuration and components of deal.II

CONFOPTS=" \
${DEAL_CONFOPTS} \
-D CMAKE_BUILD_TYPE=DebugRelease \
-D DEAL_II_WITH_MPI:BOOL=ON \
-D DEAL_II_WITH_THREADS:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_THREADS:BOOL=OFF \
-D DEAL_II_COMPONENT_DOCUMENTATION:BOOL=OFF \
-D DEAL_II_WITH_LAPACK:BOOL=ON \
-D DEAL_II_WITH_UMFPACK:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_UMFPACK:BOOL=OFF \
-D DEAL_II_WITH_BOOST:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_BOOST:BOOL=OFF \
-D DEAL_II_WITH_ZLIB:BOOL=ON \
-D DEAL_II_WITH_FUNCTIONPARSER:BOOL=ON \
-D DEAL_II_COMPONENT_MESH_CONVERTER:BOOL=ON"

################################################################################
```

To this:

```
CONFOPTS=" \
${DEAL_CONFOPTS} \
-D CMAKE_BUILD_TYPE=DebugRelease \
-D DEAL_II_WITH_MPI:BOOL=ON \
-D DEAL_II_WITH_THREADS:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_THREADS:BOOL=OFF \
-D DEAL_II_COMPONENT_DOCUMENTATION:BOOL=OFF \
-D DEAL_II_WITH_SLEPC:BOOL=OFF \
-D DEAL_II_WITH_PETSC:BOOL=OFF \
-D DEAL_II_WITH_LAPACK:BOOL=ON \
-D DEAL_II_WITH_UMFPACK:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_UMFPACK:BOOL=OFF \
-D DEAL_II_WITH_BOOST:BOOL=ON \
-D DEAL_II_FORCE_BUNDLED_BOOST:BOOL=OFF \
-D DEAL_II_WITH_ZLIB:BOOL=ON \
-D DEAL_II_WITH_FUNCTIONPARSER:BOOL=ON \
-D DEAL_II_COMPONENT_MESH_CONVERTER:BOOL=ON"
```

From here we can run candi.

```
cd ../../

./candi -j<N>
```

(where <N> is the number of processors you have available to run the configuration and installation of deal.ii and it’s dependencies).

Running this on a single processor would take around 5-6 hours.

You are asked to review the set up of candi once the installation begins. Press enter for these parts unless you notice an error. The program will make sure you have the relevant external libraries installed (see the sudo apt-get install… above) and then make sure the compilers are correct: c++ etc.

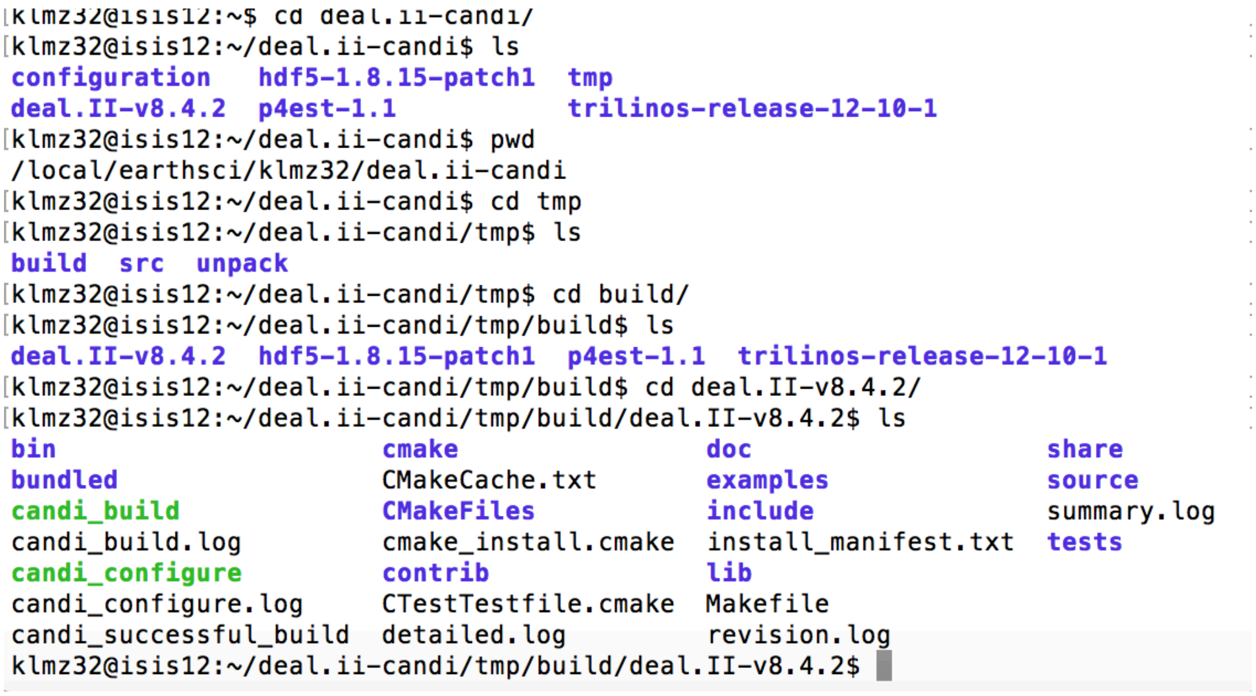
Once complete, you will have a deal.ii-candi folder at ~/

There you will find

```
configuration   hdf5-1.8.15-patch1  tmp
deal.II-v8.4.2  p4est-1.1           trilinos-release-12-10-1
```

Which shows the installation of deal.ii, p4est, hdf5 and trilinos (all latest releases).

The build files are in tmp:



```
klmz32@isis12:~$ cd deal.ii-candi/
klmz32@isis12:~/deal.ii-candi$ ls
configuration   hdf5-1.8.15-patch1  tmp
deal.II-v8.4.2  p4est-1.1           trilinos-release-12-10-1
klmz32@isis12:~/deal.ii-candi$ pwd
/local/earthsci/klmz32/deal.ii-candi
klmz32@isis12:~/deal.ii-candi$ cd tmp
klmz32@isis12:~/deal.ii-candi/tmp$ ls
build  src  unpack
klmz32@isis12:~/deal.ii-candi/tmp$ cd build/
klmz32@isis12:~/deal.ii-candi/tmp/build$ ls
deal.II-v8.4.2  hdf5-1.8.15-patch1  p4est-1.1  trilinos-release-12-10-1
klmz32@isis12:~/deal.ii-candi/tmp/build$ cd deal.II-v8.4.2/
klmz32@isis12:~/deal.ii-candi/tmp/build/deal.II-v8.4.2$ ls
bin                   cmake                doc                   share
bundled               CMakeCache.txt       examples              source
candi_build           CMakeFiles           include               summary.log
candi_build.log       cmake_install.cmake  install_manifest.txt  tests
candi_configure       contrib              lib
candi_configure.log   CTestTestfile.cmake  Makefile
candi_successful_build  detailed.log       revision.log
klmz32@isis12:~/deal.ii-candi/tmp/build/deal.II-v8.4.2$
```

The directories for the packages are found in

```
/~/deal.ii-candi/deal.II-v8.4.2

examples  include  lib  LICENSE  README.md  share
```

From here we can look to install ASPECT.

3. ASPECT

Wherever you would like ASPECT to be downloaded, go into the directory and acquire the development package of ASPECT by:

```
 git clone --recursive https://github.com/geodynamics/aspect.git
```

I would recommend always downloading the development package rather than the official release as the development package is always being added to. This means that features that are not found in the official release are already built in many months in advance. There are issues with this (some bugs may occur and cause problems) but in general this is a great way to keep on top of the development.

I also recommend downloading and installing the newest development code frequently (making sure to save any additions to the code you have made yourself). Add your email to the mailing list to keep up to date with additions, bugs, and other peoples issues - this will help to get more in tune with the code.

https://geodynamics.org/cig/software/aspect/

Once downloaded, we can begin to build and then configure. For the build we need to let ASPECT know where deal.ii is, and from there deal.ii is configured and linked to trilinos and p4est.

```
cd aspect

mkdir build

cd build

cmake -DDEAL_II_DIR=/<PATH TO DEAL.II>/deal.ii-candi/deal.II-
v8.4.2 -DASPECT_USE_FP_EXCEPTIONS=OFF ..
```

In my case the cmake would be:

```
cmake -DDEAL_II_DIR=/local/earthsci/klmz32/deal.ii-candi/
deal.II-v8.4.2 -DASPECT_USE_FP_EXCEPTIONS=OFF ..
```

During my installation I encountered a number of problems with Floating Point Exceptions (where something is divided by zero). Normally this is turned off in ASPECT, but during this installation it was default set to be ON. As a result, I needed to force floating point exceptions to be turned OFF.

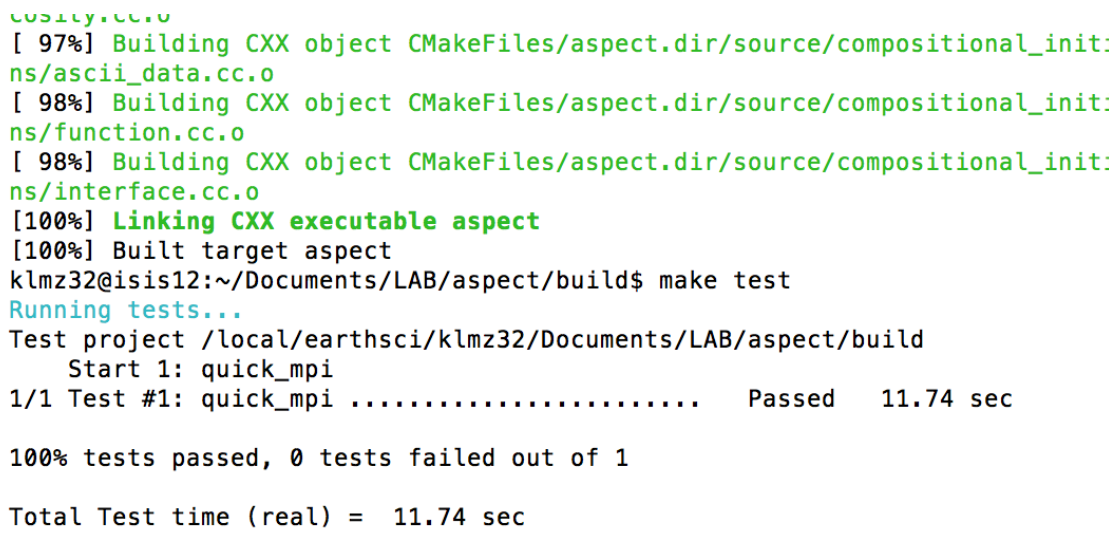
Once the cmake is complete, you need to make the executable:

```
make -j<N>
```

Again, I would recommend using a number of processors for this make, as it can take a while.

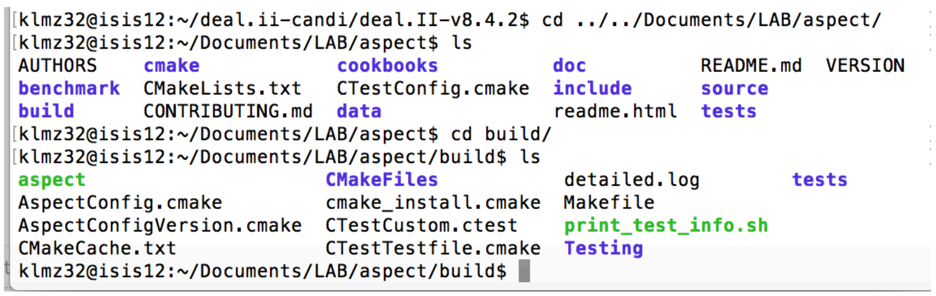
Once complete, run a make test to check to see if the executable is working correct:

```
make test
```



```
cosity.cc.o
[ 97%] Building CXX object CMakeFiles/aspect.dir/source/compositional_init:
ns/ascii_data.cc.o
[ 98%] Building CXX object CMakeFiles/aspect.dir/source/compositional_init:
ns/function.cc.o
[ 98%] Building CXX object CMakeFiles/aspect.dir/source/compositional_init:
ns/interface.cc.o
[100%] Linking CXX executable aspect
[100%] Built target aspect
klmz32@isis12:~/Documents/LAB/aspect/build$ make test
Running tests...
Test project /local/earthsci/klmz32/Documents/LAB/aspect/build
    Start 1: quick_mpi
1/1 Test #1: quick_mpi ........................   Passed   11.74 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =  11.74 sec
```

This should complete your installation. You will have an aspect executable that you can use to run numerical models:



```
[klmz32@isis12:~/deal.ii-candi/deal.II-v8.4.2$ cd ../../Documents/LAB/aspect/
[klmz32@isis12:~/Documents/LAB/aspect$ ls
AUTHORS    cmake           cookbooks        doc          README.md  VERSION
benchmark  CMakeLists.txt  CTestConfig.cmake include     source
build      CONTRIBUTING.md data             readme.html  tests
[klmz32@isis12:~/Documents/LAB/aspect$ cd build/
[klmz32@isis12:~/Documents/LAB/aspect/build$ ls
aspect                    CMakeFiles          detailed.log        tests
AspectConfig.cmake        cmake_install.cmake Makefile
AspectConfigVersion.cmake CTestCustom.ctest   print_test_info.sh
CMakeCache.txt            CTestTestfile.cmake Testing
klmz32@isis12:~/Documents/LAB/aspect/build$
```