

Tutorial for compilation on MacOS 10.8 MountainLion (french)

by [Olivier Coulon](#)

Pour avoir quelque-chose qui marche presque complètement.

- Installer Xcode puis dans Xcode installer les commandes en ligne, pour avoir gcc.
- Installer aussi homebrew, c'est nécessaire pour quelques packages. Mais je ne l'utilise pas pour scipy, numpy, etc..., pour éviter d'avoir plusieurs python sur le système.
- Utiliser le python du système.
- On utilise le ScipySuperpack qui se charge des choses difficiles comme SciPy ou NumPy.:
https://raw.githubusercontent.com/fonnesbeck/ScipySuperpack/master/install_superpack.sh
Le script est aussi [ici](#)
- On installe QT4:
Maintenant c'est là: <http://releases.qt-project.org/qt4/source/qt-mac-opensource-4.8.3.dmg> la seule version à supporter Mountain Lion. C'est OK
- On installe SIP:
<http://www.riverbankcomputing.co.uk/static/Downloads/sip4/sip-4.13.2.tar.gz>

```
python configure.py
make
sudo make install
```

Attention, l'exécutable doit être détecté. Aller linker vers /usr/bin:

```
ln -s /usr/local/sip-4.13.2/sipgen/sip /usr/bin/sip
```

- On installe PyQt.
J'ai téléchargé <http://sourceforge.net/projects/pyqt/files/PyQt4/PyQt-4.9.4/PyQt-mac-gpl-4.9.4.tar.gz/download>
Puis:

```
python configure.py
make
```

-> Ca marche DIRECT !

- Installation qwt:
<http://sourceforge.net/projects/qwt/>
Il faut, pour l'instant utiliser le qwt-5.2.1, la version 6.0.1 ne marche pas.

Faut faire :

```
qmake -spec macx-g++
```

car sinon il génère des fichiers xcodes
puis

```
make
sudo make install
```

suivi par:

```
sudo mkdir /usr/local/include
sudo ln -s /usr/local/qwt-5.2.1/lib/* /usr/local/lib/
sudo ln -s /usr/local/qwt-5.2.1/include/* /usr/local/include/
```

- Installation dcmtk:
brew install dcmtk -> marche toujours pas
- installation: doxygen
<http://www.stack.nl/~dimitri/doxygen/download.html#latestsrc>
- installation graphviz et dot:
<http://www.ryandesign.com/graphviz/>

puis dans le bash_profile rajouter:

```
export HAVE_DOT=YES
export DOT_PATH=/usr/local/graphviz-x.y/bin
```

- installation boost

```
brew install boost
```

- installation libsigc++-2.10
<http://ftp.gnome.org/pub/GNOME/sources/libsigc++/2.2/>
on le met dans /usr/local
Puis dans le répertoire:

```
./configure
make
make install
```

- installation epydocs
<http://sourceforge.net/projects/epydoc/files/latest/download>

on le met dans /usr/local puis

```
sudo make install
sudo make installdocs
```

- Il faut aller télécharger cmake sur
<http://www.cmake.org/cmake/resources/software.html>, prendre le package dmg approprié et installer.
- Le fortran n'est pas trouvé. Il faut faire:

```
sudo ln -s /usr/bin/gfortran-4.2 /usr/bin/gfortran
```

COMPILATION BRAINVISA:

- rajouter (entre autres) le path pour qmake dans le .bash_profile (pour moi: /Users/olivier/QtSDK/Desktop/Qt/474/gcc/bin/)
- installer les sources, compil de bv_maker etc...
(https://bioproj.extra.cea.fr/redmine/projects/brainvisa-devel/wiki/How_to_compile_BrainVISA_projects)

Mon .brainvisa/bv_maker.cfg :

```
[ source $HOME/brainvisa/source ]
+ standard trunk
+ perso/coulon
+ perso/operto
- communication
```

```
- nuclear_processing trunk

[ build $HOME/brainvisa/build/bug_fix ]
build_type = Release
make_options = -j1
standard bug_fix $HOME/brainvisa/source
- soma-io
- connectomist-*
- communication
- fmri

[ build $HOME/brainvisa/build/trunk ]
make_options = -j1
standard trunk $HOME/brainvisa/source
- soma-io
- connectomist-*
- communication
- fmri
```

- Attention, par défaut le compilateur C++ pour moi est Clang, il faut changer la variable d'environnement:

```
export CXX=/usr/bin/g++
```

Je l'ai mise dans mon .bash_profile.

- Puis classique:

```
bv_maker sources
bv_maker configure
bv_maker build
```

Files

| | | | |
|----------------------|---------|------------|----------------|
| install_superpack.sh | 2.81 KB | 01/11/2012 | Riviere, Denis |
|----------------------|---------|------------|----------------|