

---

# Installation de la bibliothèque SLIP sur les plateformes Linux et Windows

*Version 0.1*

Institut Pprime  
Département Fluides, Thermique et Combustion  
Martin DRUON

16 November 2014



# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Installation sous Linux</b>   | <b>3</b>  |
| 1.1      | Installation des bibliothèques requises par la bibliothèque SLIP . . . . .                           | 3         |
| 1.2      | Installation des fichiers nécessaires à la compilation de SLIP . . . . .                             | 3         |
| 1.3      | Installation de la bibliothèque SLIP . . . . .   | 4         |
| <b>2</b> | <b>Installation sous Windows</b>   | <b>5</b>  |
| 2.1      | Installation de l'environnement MinGW64 . . . . .  | 5         |
| 2.2      | Installation de l'interpréteur MSYS . . . . .  | 8         |
| 2.3      | Installation du script d'installation et des patches . . . . .                                       | 9         |
| 2.4      | Installation des fichiers nécessaires à la compilation de SLIP . . . . .                             | 9         |
| 2.5      | Installation des bibliothèques requises via le script <code>install_SLIP_MinGW64.sh</code> . . . . . | 10        |
| <b>3</b> | <b>Détail du script d'installation <code>install_SLIP_MinGW64.sh</code></b>                          | <b>11</b> |
| 3.1      | Contenu du fichier <code>/etc/fstab</code> . . . . .   | 11        |
| 3.2      | Contenu du fichier <code>~/.profile</code> . . . . .   | 11        |
| 3.3      | Installation de <code>pkg-config 0.28</code> . . . . .   | 12        |
| 3.4      | Installation de <code>cmake 2.8.12.2</code> . . . . .  | 12        |
| 3.5      | Installation de <code>freeglut 2.8.1</code> . . . . .  | 12        |
| 3.6      | Installation de <code>zlib 1.2.8</code> . . . . .  | 13        |
| 3.7      | Installation de <code>bzip2 1.0.6</code> . . . . .   | 13        |
| 3.8      | Installation de <code>FFTW 3.3.3</code> . . . . .  | 14        |
| 3.9      | Installation de <code>libjpeg 9</code> . . . . .   | 14        |
| 3.10     | Installation de <code>libtiff 4.0.3</code> . . . . .   | 14        |
| 3.11     | Installation de <code>libpng 1.6.9</code> . . . . .  | 15        |
| 3.12     | Installation de <code>freetype2 2.5.0.1</code> . . . . .   | 15        |
| 3.13     | Installation d' <code>ImageMagick 6.8.7-10</code> . . . . .  | 16        |
| 3.14     | Installation de <code>HDF5 1.8.10-patch1</code> . . . . .  | 17        |
| 3.15     | Installation de <code>NetCDF 4.1.3</code> . . . . .  | 18        |
| 3.16     | Installation de <code>Boost 1.52.0</code> . . . . .  | 18        |
| 3.17     | Installation de <code>TBB 4.2</code> . . . . .   | 19        |
| 3.18     | Installation de <code>QT 5.2.0</code> . . . . .  | 19        |
| 3.19     | Installation de <code>VTK 5.10.1</code> . . . . .  | 20        |



Ce document détaille la procédure d'installation de la bibliothèque [SLIP](#), ainsi que les bibliothèques dépendantes, pour les plateformes suivantes :

- Linux (testé sous Ubuntu 10.04, 12.04 et 13.04)
- Windows 64 bits (testé sous Windows XP, Seven et 8)

Les outils et bibliothèques requis pour l'installation de la bibliothèque SLIP sont les suivants :

- gcc / g++
- pkg-config
- cmake
- freeglut
- zlib
- bzip2
- FFTW
- libjpeg
- libtiff
- libpng
- freetype
- ImageMagick
- HDF5
- NetCDF
- boost
- tbb (optionnel)
- Qt (optionnel)
- VTK (optionnel)

---

**Note :** Des changements majeurs ont été effectués dans l'A.P.I. de `NetCDF`, depuis la version 4.2. Actuellement, ces changements n'ont pas encore été pris en compte dans la bibliothèque SLIP. Il est donc nécessaire d'installer NetCDF 4.1.3 **au maximum**.

---



# Chapitre 1

## Installation sous Linux

### 1.1 Installation des bibliothèques requises par la bibliothèque SLIP

Sous le système d'exploitation Linux, l'installation des bibliothèques requises par la bibliothèque SLIP peut se faire en utilisant directement le gestionnaire de paquets `apt` ou depuis l'interface `synaptic`.

La liste des paquets à installer est la suivante :

- `gcc`
- `pkg-config`
- `cmake` et `cmake-curses-gui`
- `freeglut3` et `freeglut3-dev`
- `zlib1g` et `zlib1g-dev`
- `bzip2`, `libbz2-1.0` et `libbz2-dev`
- `libblas3gf`, `libblas-dev`, `liblapack3gf` et `liblapack-dev`
- `libfftw3-3` et `libfftw3-dev`
- `libjpeg8` et `libjpeg8-dev`
- `libtiff4` et `libtiff4-dev`
- `libpng12-0` et `libpng12-dev`
- `libfreetype6` et `libfreetype6-dev`
- `imagemagick`, `libmagickcore4`, `libmagickwand4`, `libmagick++4` et `libmagick++-dev`
- `libhdf5-openmpi-1.8.4` et `libhdf5-openmpi-dev`
- `netcdf-bin`, `libnetcdf6` et `libnetcdf-dev`
- `libboost-all-dev`
- `libtbb2` et `libtbb-dev`
- `qt-sdk`
- `libvtk5.8`, `libvtk5.8-qt4`, `libvtk5-dev`, `libvtk5-qt4-dev`

Il convient d'installer également les paquets dépendants.

---

**Note :** Selon les distributions, les numéros de version des bibliothèques peuvent être différents.

---

**Warning :** Actuellement, la bibliothèque SLIP n'est pas compatible avec les dernières versions de la bibliothèque NetCDF. Il convient donc d'installer NetCDF 4.1.3 **au maximum**.

### 1.2 Installation des fichiers nécessaires à la compilation de SLIP

Copier le fichier `FindNetCDF.cmake` (et si nécessaire, le fichier `FindTBB.cmake`) dans le répertoire `Modules` du programme `cmake`. Par défaut, ce répertoire est situé à l'emplacement suivant : `/usr/share/cmake-2.8/Modules`.

## 1.3 Installation de la bibliothèque SLIP

Après l'installation des paquets cités précédemment, il est possible de compiler la bibliothèque SLIP. Pour cela, récupérer la dernière version de la bibliothèque depuis le [site web](#) et exécuter les commandes suivantes :

```
$ cd /usr/local/src
$ mv ~/Telechargements/slip-X.XX.X.tar.gz
$ tar xvf slip-X.XX.X.tar.gz
$ cd slip
$ mkdir build
$ cd build
$ cmake ..
$ make && make install
```

avec `slip-X.XX.X.tar.gz` le nom de l'archive téléchargée. Par défaut, l'installation se fait dans le répertoire `/usr/local`. Pour installer la bibliothèque dans un autre répertoire, il faut remplacer la ligne

```
$ cmake ..
```

par

```
$ cmake -DCMAKE_INSTALL_PREFIX=nom_du_repertoire ..
```

avec `nom_du_repertoire` le répertoire d'installation.



## Chapitre 2

# Installation sous Windows

Le système d'exploitation Windows est livré sans environnement de développement. Par la suite, les différents programmes et bibliothèques requis sont compilés en utilisant le compilateur **MinGW64**. Ce dernier contient l'ensemble des outils GNU nécessaires à la compilation des programmes et bibliothèques développés en C et C++. Utilisé conjointement avec l'interpréteur de commandes **MSYS**, ces deux outils permettent de développer des applications Windows natives (c'est-à-dire sans être obligé de passer par un émulateur) en utilisant directement les outils GNU proposés sous Linux, tels que `gcc` / `g++` ou les `Makefile`.

---

**Note :** Sous Windows, le compilateur **MinGW64** est utilisé par plusieurs environnements de développement tels que **CodeBlocks**, **Dev-Cpp** ou **Eclipse**.

---

Après avoir installé l'environnement **MinGW64** / **MSYS**, un script permet de télécharger et de compiler automatiquement les bibliothèques nécessaires à l'installation de **SLIP**. Ce script a été testé sur les plateformes Windows XP 64bits, Windows 7 et Windows 8.

Les étapes à réaliser sont les suivantes :

- Installation de l'environnement **MinGW64**
- Installation de l'interpréteur **MSYS**
- Installation du script d'installation et des patches
- Installation des bibliothèques depuis le script d'installation

## 2.1 Installation de l'environnement MinGW64

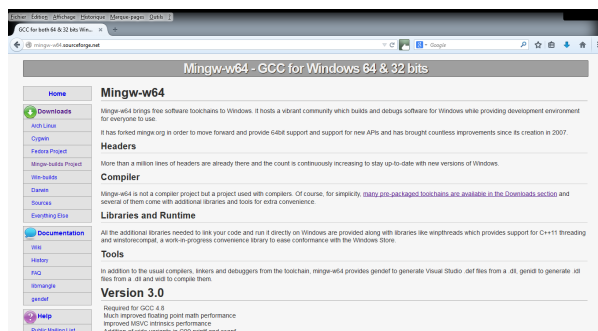
### 2.1.1 Caractéristiques de l'environnement de développement MinGW64

À l'heure actuelle, les caractéristiques de l'environnement de développement **MinGW64** sont :

- **MinGW64** 64-bits
- **GCC** 4.9.2
- **Threads** : `posix`
- **Exceptions** : `seh`
- **Python** 2.7
- Site officiel : <http://mingw-w64.sourceforge.net/>

### 2.1.2 Procédure d'installation

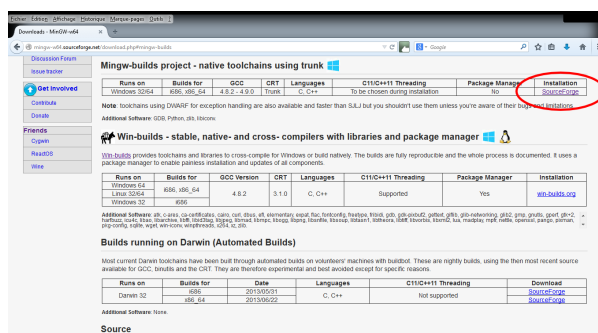
Aller sur le site de téléchargement de **MinGW64** : <http://mingw-w64.sourceforge.net/>



Dans la section “Download” (à gauche), cliquer sur “Mingw-builds Project”



Dans le tableau, cliquer sur le lien “SourceForge” situé dans la colonne “Installation”

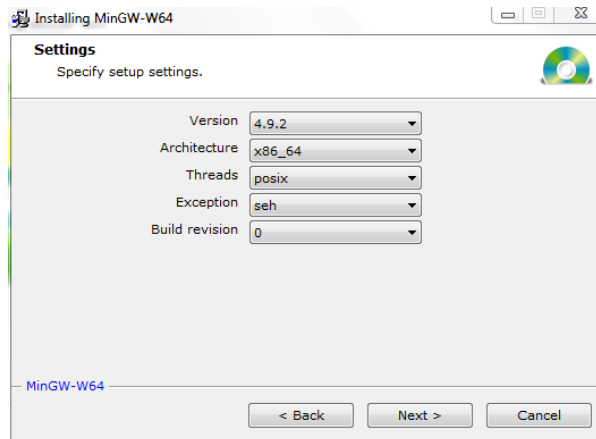


Télécharger et exécuter le programme mingw-w64-install.exe

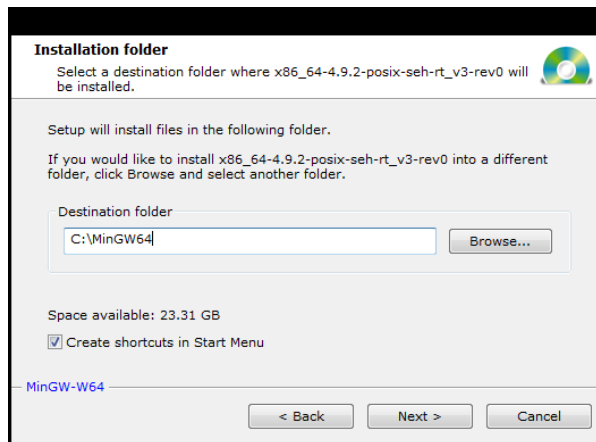


Sélectionner les paramètres :

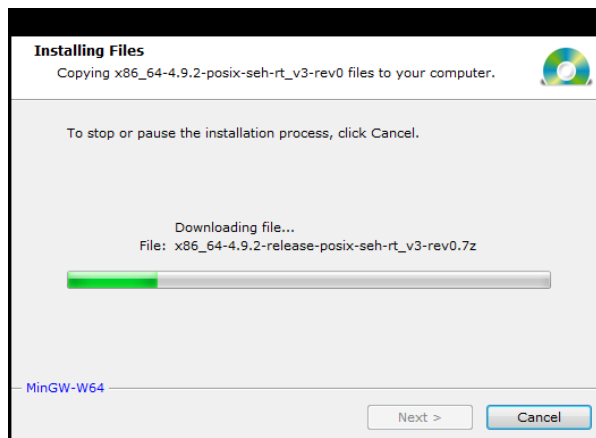
- Version : dernière version de GCC (actuellement 4.9.2)
- Architecture : x86\_64
- Threads : posix
- Exception : seh
- Build revision : dernière révision (actuellement 0)

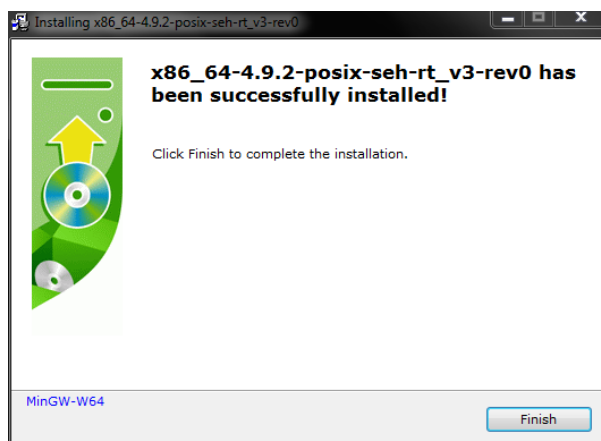


Choisir le répertoire d'installation. Par la suite, l'installation est effectuée dans le répertoire C:\MinGW64

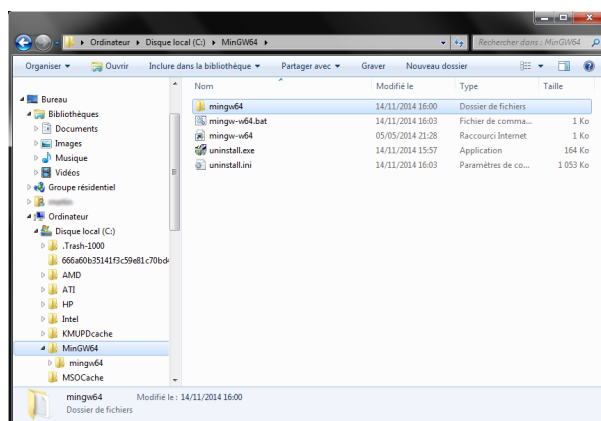


Attendre la fin de l'installation





L'arborescence de l'installation est alors la suivante :



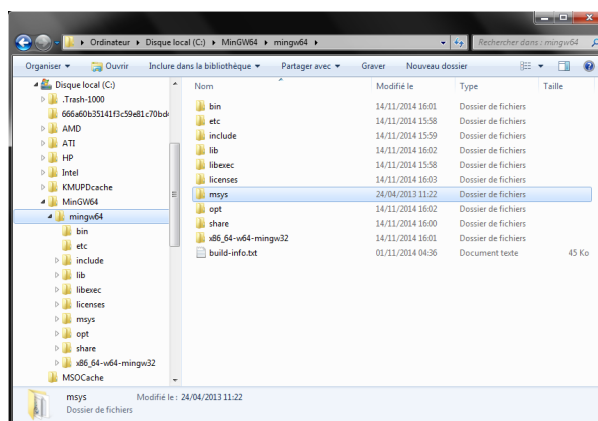
## 2.2 Installation de l'interpréteur MSYS

### 2.2.1 Caractéristiques de l'interpréteur MSYS

- Interpréteur de commandes linux (équivalent `bash`)
- Compression/décompression de fichiers via l'outil `7zip`
- Téléchargement de fichier via l'outil `wget`
- Gestionnaires de version : `svn`, `git`, `mercurial` et `cvs`

### 2.2.2 Procédure d'installation

- Aller sur le site de MSYS pour MinGW64 : <http://sourceforge.net/projects/mingwbuilds/files/external-binary-packages/>
- Télécharger le fichier la dernière version de MSYS (actuellement `msys+7za+wget+svn+git+mercurial+cvs-rev13.7z`)
- Décompresser l'archive
- Déplacer le répertoire `msys` dans le répertoire `C:\MinGW64\mingw64` créé précédemment



### 2.2.3 Exécution de l'interpréteur MSYS

Exécuter le fichier `msys.bat` situé dans le répertoire `C:\MinGW64\mingw64\msys`.

Lors du premier lancement de l'interpréteur MSYS, le répertoire utilisateur `C:\MinGW64\mingw64\msys\home\nom_utilisateur` va être créé.

**Warning :** Par défaut, les répertoires utilisateurs sont stockés dans `C:\MinGW64\mingw64\msys\home\`. De ce fait, les comptes ne sont pas dans un répertoire généralement sauvegardés par défaut. Il faut donc penser à sauvegarder ce répertoire manuellement. De plus, ce répertoire, et donc toutes les données qu'il contient, peut être accessible par tous les utilisateurs de l'ordinateur (selon les droits définis sur la machine). Une solution à ces deux problèmes consisterait à déplacer le répertoire contenant les comptes utilisateurs dans un répertoire sécurisé, en modifiant la variable `HOME` au démarrage de MSYS. Cette solution n'a pas été testée.

Dans l'interpréteur MSYS, les chemins respectent la norme Linux. Le tableau suivant liste certains répertoires fréquemment utilisés sous Linux et leur équivalent sous Windows :

| Chemins sous MSYS                  | Equivalent sous Windows                             |
|------------------------------------|---|
| <code>/mingw</code>                | <code>C:\MinGW\mingw64</code>                       |
| <code>/usr</code>                  | <code>C:\MinGW\mingw64\msys</code>                  |
| <code>/usr/local/bin</code>        | <code>C:\MinGW\mingw64\msys\local\bin</code>        |
| <code>/usr/home/utilisateur</code> | <code>C:\MinGW\mingw64\msys\home\utilisateur</code> |
| <code>/c</code>                    | <code>C:\</code>                                    |
| <code>/c/Users/utilisateur</code>  | répertoire utilisateur                              |

## 2.3 Installation du script d'installation et des patches

Copier le script d'installation `install_SLIP_MinGW64.sh` dans le dossier utilisateur `C:\MinGW64\mingw64\home\nom_utilisateur`.

Copier les patches `bzip2-1.0.6-mingw.patch` et `netcdf-4.1.3-mingw.patch` dans le répertoire `C:\MinGW64\mingw64\msys\local\src` (créer l'arborescence si nécessaire).

## 2.4 Installation des fichiers nécessaires à la compilation de SLIP

Copier le fichier `FindNetCDF.cmake` (et si nécessaire, le fichier `FindTBB.cmake`) dans le répertoire `Modules` du programme `cmake`. Par défaut, ce répertoire est situé à l'emplacement suivant : `/usr/share/cmake-2.8/Modules`.

## 2.5 Installation des bibliothèques requises via le script install\_SLIP\_MinGW64.sh

Démarrer l'interpréteur MSYS, se déplacer dans le répertoire utilisateur et exécuter le script.

```
$ cd ~
$ ./install_SLIP_MinGW64.sh
```

Exécuter la première étape du script et **quitter le terminal à la fin de l'exécution de cette étape** pour prendre en compte les modifications dans les fichiers de configuration.

Relancer l'interpréteur MSYS et effectuer les autres étapes **dans l'ordre**. Le script va alors automatiquement télécharger, extraire, configurer, compiler et installer chaque bibliothèque.

À la fin de chaque installation, vérifier que tout s'est déroulé correctement.

## Chapitre 3

# Détail du script d'installation install\_SLIP\_MinGW64.sh

### 3.1 Contenu du fichier /etc/fstab

```
C:/MinGW64/mingw64      /mingw
```

### 3.2 Contenu du fichier ~/.profile

```
alias ls="ls --color=auto"
alias rm="rm -i"

export H5_CFLAGS="-DH5_HAVE_MINGW -DH5_HAVE_WIN32_API"

export CFLAGS+=" -fPIC -m64 -mthreads -DNDEBUG -DHAVE_USABLE_WCHAR_T -DUNICODE "
export CPPFLAGS+=" -fPIC -m64 -mthreads -DNDEBUG -DHAVE_USABLE_WCHAR_T -DUNICODE \
-I/usr/local/include -I/usr/local/include/boost-1_52 \
-I/usr/local/include/ImageMagick-6 -I/usr/local/include/GL \
-I/usr/local/include/freetype2 -I/usr/local/include/libpng16 \
-I/usr/local/share/Qt5.2.0/include -I/usr/include -I/mingw/include \
-I/mingw/x86_64-w64-mingw32/include "

export CXXFLAGS+=${CPPFLAGS}
export LDFLAGS+=" -fPIC -m64 -mthreads -L/usr/local/lib -L/usr/local/share/Qt5.2.0/lib \
-L/usr/lib -L/mingw/lib -L/mingw/x86_64-w64-mingw32/lib "

export INCLUDE_PATH=/usr/local/include:/usr/local/include/boost-1_52:\
/usr/local/include/ImageMagick-6:/usr/local/include/GL:\
/usr/local/include/freetype2:/usr/local/include/libpng16:\
/usr/local/share/Qt5.2.0/include:/usr/include:/mingw/include:\
/mingw/x86_64-w64-mingw32/include:${INCLUDE_PATH}
export LIBRARY_PATH=/usr/local/lib:/usr/local/share/Qt5.2.0/lib:/usr/lib:/mingw/lib:\
/mingw/x86_64-w64-mingw32/lib:${LIBRARY_PATH}

export PATH=/usr/local/bin:/usr/bin:/usr/sbin:/mingw/bin:/mingw/opt/bin:\
/usr/local/share/Qt5.2.0/bin:${PATH}
export LD_LIBRARY_PATH=/usr/local/lib:/usr/local/share/Qt5.2.0/lib:/mingw/lib:\
/mingw/x86_64-w64-mingw32/lib:${LD_LIBRARY_PATH}
export LD_RUN_PATH=/usr/local/lib:/mingw/lib:/mingw/x86_64-w64-mingw32/lib:${LD_RUN_PATH}
```

### 3.3 Installation de pkg-config 0.28

pkg-config est un utilitaire permettant de simplifier la phase de compilation d'une application.

1. Récupérer l'archive `pkg-config-x.x.x.tar.gz` contenant les sources à partir du site [pkg-config](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/pkg-config-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf pkg-config-x.x.x.tar.gz
$ cd pkg-config-x.x.x
$ ./configure --prefix=/usr/local --build=x86_64-w64-mingw32 --with-internal-glib
$ make && make install
```

### 3.4 Installation de cmake 2.8.12.2

cmake est une bibliothèque destinée à simplifier la compilation et la maintenance des projets, quel que soit le système d'exploitation utilisé. C'est une alternative aux autotools.

1. Récupérer l'archive `cmake-x.x.x.tar.gz` contenant les sources à partir du site [cmake](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/cmake-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf cmake-x.x.x.tar.gz
$ cd cmake-x.x.x
$ CFLAGS="-fPIC -m64 -DNDEBUG" CPPFLAGS="-fPIC -m64 -DNDEBUG" \
  CXXFLAGS="-fPIC -m64 -DNDEBUG" \
  ./configure --prefix=/usr/local
$ make && make install
```

Pour compiler un programme depuis l'environnement MSYS, il est nécessaire de préciser l'environnement de façon explicite. De plus, sous Windows, le répertoire d'installation par défaut est : `C:\Programme Files\`. Pour modifier ce répertoire d'installation, il est nécessaire de modifier la variable `CMAKE_INSTALL_PREFIX`.

```
$ mkdir build
$ cd build
$ cmake -G"MSYS Makefiles" -DCMAKE_INSTALL_PREFIX=/usr/local ..
$ make
$ make install
```

### 3.5 Installation de freeglut 2.8.1

freetglut est une alternative OpenSource à la bibliothèque *OpenGL Utility Toolkit* (GLUT).

1. Récupérer l'archive `freetglut-x.x.x.tar.gz` contenant les sources à partir du site [freetglut](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé



#### 5. Compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/freeglut-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf freeglut-x.x.x.tar.gz
$ cd freeglut-x.x.x
$ ./configure --prefix=/usr/local --build=x86_64-w64-mingw32
$ make && make install
```

## 3.6 Installation de zlib 1.2.8

zlib est une bibliothèque C permettant de compresser/décompresser des données sans perte d'information.

1. Récupérer l'archive `zlib-x.x.x.tar.gz` contenant les sources à partir du site [zlib](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/zlib-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf zlib-x.x.x.tar.gz
$ cd zlib-x.x.x
$ BINARY_PATH=/usr/local/bin INCLUDE_PATH=/usr/local/include \
  LIBRARY_PATH=/usr/local/lib make install -f./win32/Makefile.gcc SHARED_MODE=1
```

Il faut que les variables `BINARY_PATH`, `INCLUDE_PATH` et `LIBRARY_PATH` aient été correctement définies avant l'étape de compilation.

## 3.7 Installation de bzip2 1.0.6

bzip2 est une bibliothèque C permettant de compresser/décompresser des données sans perte d'information.

1. Récupérer l'archive `bzip2-x.x.x.tar.gz` contenant les sources à partir du site [bzip2](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Sous Windows, patcher la bibliothèque en utilisant le patch `bzip2-1.0.6-mingw.patch`
6. Compiler les exécutables puis compiler la bibliothèque
7. Copier la bibliothèque générée dans le répertoire `/usr/local/lib`

```
$ cp ~/Téléchargements/bzip2-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf bzip2-x.x.x.tar.gz
$ cd bzip2-x.x.x
$ patch -p2 -i ../bzip2-1.0.6-mingw.patch
$ make && make install PREFIX=/usr/local && make -f Makefile-libbz2_so
$ cp libbz2.so.1.0 libbz2.so.1.0.6 /usr/local/lib
```

## 3.8 Installation de FFTW 3.3.3

FFTW est une bibliothèque C permettant de calculer des transformées de Fourier discrètes multidimensionnelles, sur des données réelles ou complexes.

1. Récupérer l'archive `fftw-x.x.x.tar.gz` contenant les sources à partir du site [FFTW3](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/fftw-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf fftw-x.x.x.tar.gz
$ cd fftw-x.x.x
$ ./configure --prefix=/usr/local --build=x86_64-w64-mingw32 --with-our-malloc \
    --with-windows-f77-mangling --enable-shared \
    --enable-static --enable-sse2
$ make && make install
```

## 3.9 Installation de libjpeg 9

libjpeg est une bibliothèque C permettant de lire et d'écrire des images au format JPEG.

1. Récupérer l'archive `jpegsrc.vx.tar.gz` contenant les sources à partir du site [libjpeg](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/jpegsrc.vx.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf jpegsrc.vx.tar.gz
$ cd jpeg-vx
$ ./configure --prefix=/usr/local --build=x86_64-w64-mingw32 --enable-shared --enable-static
$ make && make install
```

## 3.10 Installation de libtiff 4.0.3

libtiff est une bibliothèque C permettant de lire et d'écrire des images au format TIFF (*Tag Image File Format*).

1. Récupérer l'archive `tiff-x.x.x.tar.gz` contenant les sources à partir du site [libtiff](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/tiff-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf tiff-x.x.x.tar.gz
$ cd tiff-x.x.x
$ ./configure --prefix=/usr/local --build=x86_64-w64-mingw32 --enable-shared --enable-static
$ make && make install
```

Il faut que les variables CFLAGS et LDFLAGS aient été correctement définies avant l'étape de compilation.

**Warning :** À la fin de l'étape de compilation (après le `./configure`), vérifier que les bibliothèques `zlib` et `jpeg` ait bien été détectées. Vérifier également que le langage C++ est supporté.

Libtiff is now configured for x86\_64-unknown-linux-gnu

```

Installation directory:      /usr/local
Documentation directory:    ${prefix}/share/doc/tiff-3.8.2
C compiler:                 gcc -g -O2 -Wall -W
C++ compiler:               g++ -g -O2
Enable runtime linker paths: no
Support Microsoft Document Imaging: yes

Support for internal codecs:
CCITT Group 3 & 4 algorithms: yes
Macintosh PackBits algorithm: yes
LZW algorithm:              yes
ThunderScan 4-bit RLE algorithm: yes
NeXT 2-bit RLE algorithm:   yes
LogLuv high dynamic range encoding: yes

Support for external codecs:
ZLIB support:               yes          <----- support de zlib
Pixar log-format algorithm: yes
JPEG support:               yes          <----- support de libjpeg
Old JPEG support:           no

C++ support:                yes          <----- support du langage C++

OpenGL support:             yes

```

### 3.11 Installation de libpng 1.6.9

`libpng` est une bibliothèque C permettant de lire et d'écrire des images au format PNG (*Portable Network Graphics*).

1. Récupérer l'archive `libpng-x.x.x.tar.gz` contenant les sources à partir du site [libpng](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Configurer, compiler et installer la bibliothèque

```

$ cp ~/Téléchargements/libpng-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf libpng-x.x.x.tar.gz
$ cd libpng-x.x.x
$ ./configure --prefix=/usr/local --build=x86_64-w64-mingw32 --enable-shared --enable-static
$ make && make install

```

### 3.12 Installation de freetype2 2.5.0.1

FreeType 2 est une bibliothèque permettant de gérer les polices de caractères (*fonts*).

1. Récupérer l'archive `freetype-x.x.x.tar.gz` contenant les sources à partir du site [FreeType](#)
2. Copier l'archive dans le répertoire `/usr/local/src`

3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/freetype-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf freetype-x.x.x.tar.gz
$ cd freetype-x.x.x
$ sh autogen.sh && ./configure --prefix=/usr/local --build=x86_64-w64-mingw32 \
  --enable-shared --enable-static
$ make && make install
```

### 3.13 Installation d'ImageMagick 6.8.7-10

ImageMagick est une bibliothèque permettant de créer, modifier ou convertir des fichiers graphiques. Elle permet de lire et d'écrire des images dans plus de 100 formats différents. Elle propose également un ensemble de fonctions pour la lecture, l'écriture et la modification d'images à partir des langages C et C++.

1. Récupérer l'archive `ImageMagick-x.x.x.tar.gz` contenant les **sources UNIX** (ne pas utiliser la version précompilée pour Windows) à partir du site d'[ImageMagick](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/ImageMagick-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf ImageMagick-x.x.x.tar.gz
$ cd ImageMagick-x.x.x
$ LDFLAGS+=" -L/usr/lib64 -L/usr/lib" ./configure --prefix=/usr/local \
  --build=x86_64-w64-mingw32 --enable-shared --enable-static
$ make && make install
```

**Warning :** À la fin de l'étape de compilation (après le `./configure`), vérifier que les bibliothèques `FFTW`, `freetype`, `libjpeg`, `libpng`, `libtiff` et `zlib` ait bien été détectées. Vérifier également que le langage C++ est supporté via l'interface `Magick++`.

|                            | Option                         | Value                           |                               |
|----------------------------|--------------------------------|---------------------------------|-------------------------------|
| Shared libraries           | --enable-shared=yes            | yes                             |                               |
| Static libraries           | --enable-static=yes            | yes                             |                               |
| Module support             | --with-modules=no              | no                              |                               |
| GNU ld                     | --with-gnu-ld=yes              | yes                             |                               |
| Quantum depth              | --with-quantum-depth=16        | 16                              |                               |
| High Dynamic Range Imagery | --enable-hdri=no               | no                              |                               |
| Delegate Configuration:    |                                |                                 |                               |
| BZLIB                      | --with-bzlib=yes               | yes                             |                               |
| Autotrace                  | --with-autotrace=no            | no                              |                               |
| Dejavu fonts               | --with-dejavu-font-dir=default | none                            |                               |
| DJVVU                      | --with-djvu=yes                | yes                             |                               |
| DPS                        | --with-dps=yes                 | no                              |                               |
| FFTW                       | --with-fftw=yes                | yes                             | <----- support de FFTW        |
| FlashPIX                   | --with-fpx=yes                 | no                              |                               |
| FontConfig                 | --with-fontconfig=yes          | yes                             |                               |
| FreeType                   | --with-freetype=yes            | yes                             | <----- support de FreeType    |
| GhostPCL                   | None                           | pcl6 (unknown)                  |                               |
| GhostXPS                   | None                           | gxps (unknown)                  |                               |
| Ghostscript                | None                           | gs (8.71)                       |                               |
| Ghostscript fonts          | --with-gs-font-dir=default     | /usr/share/fonts/type1/gsfonts/ |                               |
| Ghostscript lib            | --with-gslib=no                | no                              |                               |
| Graphviz                   | --with-gvc=no                  |                                 |                               |
| JBIG                       | --with-jbig=yes                | no                              |                               |
| JPEG v1                    | --with-jpeg=yes                | yes                             | <----- support de libjpeg     |
| JPEG-2000                  | --with-jp2=yes                 | yes                             |                               |
| LCMS v1                    | --with-lcms=yes                | yes                             |                               |
| LCMS v2                    | --with-lcms2=yes               | no                              |                               |
| LQR                        | --with-lqr=yes                 | no                              |                               |
| LTDL                       | --with-ltdl=yes                | no                              |                               |
| LZMA                       | --with-lzma=yes                | no                              |                               |
| Magick++                   | --with-magick-plus-plus=yes    | yes                             | <----- support du langage C++ |
| OpenEXR                    | --with-openexr=yes             | yes                             |                               |
| PERL                       | --with-perl=no                 | no                              |                               |
| PANGO                      | --with-pango=yes               | no                              |                               |
| PNG                        | --with-png=yes                 | yes                             | <----- support de libpng      |
| RSVG                       | --with-rsvg=no                 | no                              |                               |
| TIFF                       | --with-tiff=yes                | yes                             | <----- support de libtiff     |
| WEBP                       | --with-webp=yes                | no                              |                               |
| Windows fonts              | --with-windows-font-dir=       | none                            |                               |
| WMF                        | --with-wmf=no                  | no                              |                               |
| X11                        | --with-x=                      | yes                             |                               |
| XML                        | --with-xml=yes                 | yes                             |                               |
| ZLIB                       | --with-zlib=yes                | yes                             | <----- support de zlib        |

**Note :** Des problèmes de compilation semblent se produire en utilisant la version 6.8.3-10. Il est donc conseillé de ne pas utiliser cette version.

### 3.14 Installation de HDF5 1.8.10-patch1

HDF5 est une bibliothèque utilisée pour sauvegarder de gros volumes de données.

1. Récupérer l'archive `hdf5-x.x.x.tar.gz` contenant les sources à partir du site [HDF5](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/hdf5-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf hdf5-x.x.x.tar.gz
$ cd hdf5-x.x.x
$ ./configure --prefix=/usr/local --build=x86_64-w64-mingw32 --enable-shared \
--enable-static --enable-cxx
$ make && make install
```

### 3.15 Installation de NetCDF 4.1.3

NetCDF4 est une bibliothèque utilisée pour sauvegarder de gros volumes de données. Elle est basée sur la bibliothèque HDF5.

1. Récupérer l'archive `netcdf-x.x.x.tar.gz` contenant les sources à partir du site [NetCDF](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Appliquer le patch `netcdf-4.1.3-mingw.patch`
6. Copier le fichier `libsrc/onstack.h` dans le répertoire `libdispatch` pour éviter une erreur de compilation
7. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/netcdf-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf netcdf-x.x.x.tar.gz
$ cd netcdf-x.x.x
$ patch -p2 -i ../netcdf-4.1.3-mingw.patch
$ cp /usr/local/src/netcdf-x.x.x/libsrc/onstack.h /usr/local/src/netcdf-x.x.x/libdispatch
$ ./configure --prefix=/usr/local --build=x86_64-w64-mingw32 --enable-shared --enable-static \
--enable-dll --enable-netcdf-4 --enable-cxx-4
$ make && make install
```

### 3.16 Installation de Boost 1.52.0

Boost est un ensemble d'algorithmes et de bibliothèques destinés à simplifier et à accélérer les calculs scientifiques développés en C++. De nombreuses bibliothèques sont développées et testées sous Boost avant d'être intégrées définitivement dans la norme du langage C++. Le coeur de la bibliothèque est composée uniquement de fichiers d'entête (`.hpp`). Ces derniers ne nécessitent pas de compilation pour pouvoir être utilisés. Par contre, Boost est également livré avec un grand nombre de bibliothèques annexes, dont certaines sont utilisées par la bibliothèque SLIP. Il est donc nécessaire de compiler l'intégralité des bibliothèques pour pouvoir utiliser toutes les fonctionnalités de SLIP. Parmi les bibliothèques utilisées, on peut citer :

- `Boost.Chrono` : gestion des timers, indépendamment du système d'exploitation utilisé
- `Boost.ProgramOptions` : gestion des paramètres d'un programme, passés depuis la ligne de commande
- `Boost.PropertyTree` : lecture/écriture des fichiers de paramètres, sous différents formats (XML, ini, ...)
- `Boost.Python` : interface entre les langages C++ et Python

- Boost.Thread : gestions de *threads*
- Boost.uBLAS : interface C++ de la bibliothèque BLAS.

Contrairement aux bibliothèques précédemment installées, la configuration de Boost ne s'effectue pas depuis l'environnement MSYS mais depuis la console Windows.

1. Récupérer l'archive `boost_x_x_x.tar.gz` à partir du site de [Boost](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Ouvrir un terminal MS-DOS et exécuter le programme `bootstrap.bat` depuis ce terminal
5. **Une fois la configuration terminée**, revenir sur le terminal MSYS et lancer la compilation et l'installation de la bibliothèque

```
$ cp ~/Téléchargements/boost_x_x_x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf boost_x_x_x.tar.gz
$ cd boost_x_x_x
$ start bootstrap.bat gcc
```

[ Attendre la fin de la configuration dans le terminal MS-DOS avant de continuer ]

```
$ b2.exe install --prefix=/usr/local --build-type=complete --build-dir=./build \
  toolset=gcc variant=release threading=multi link=shared pch=off \
  define=BOOST_USE_WINDOWS_H
```

---

**Note :** Si l'erreur `Duplicate name of actual target...` se produit lors de la compilation, essayer de compiler la bibliothèque avec l'option "link=static".

---

## 3.17 Installation de TBB 4.2

TBB une bibliothèque C++ permettant d'écrire simplement du code parallélisable.

1. Récupérer l'archive `tbbxx_xxxoss.tar.gz` contenant les sources à partir du site [TBB](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/tbbxx_xxxoss.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf tbbxx_xxxoss.tar.gz
$ cd tbbxx_xxx
$ mingw32-make.exe compiler=gcc arch=intel64 runtime=mingw
```

## 3.18 Installation de Qt 5.2.0

Qt est une bibliothèque C++ permettant de réaliser des interfaces graphiques sur différents systèmes d'exploitation, comme, par exemple, Linux, Mac OS X et Windows.

1. Récupérer l'archive `qt-everywhere-opensource-src-x.x.x.tar.gz` contenant les sources à partir du site [Qt](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Appliquer le patch `qt-everywhere-opensource-src-5.2.0-disabled_modules.patch`

### 6. Configurer, compiler et installer la bibliothèque

```
$ cp ~/Téléchargements/qt-everywhere-opensource-src-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf qt-everywhere-opensource-src-x.x.x.tar.gz
$ cd qt-everywhere-opensource-src-x.x.x
$ patch -p2 -i ../qt-everywhere-opensource-src-5.2.0-disabled_modules.patch
$ ./configure -prefix /usr/local/share/Qt5.2.0 -opensource -confirm-license -release \
  -developer-build -opengl desktop -no-javascript-jit -platform win32-g++ -v
$ CFLAGS+=" -Wno-error " CXXFLAGS+=" -Wno-error " \
  PATH=$PATH:/usr/local/qt-everywhere-opensource-src/build/qtbase/lib make && make install
```

## 3.19 Installation de VTK 5.10.1

VTK est un ensemble de classe C++ orientée vers le calcul scientifique, le traitement d'images et la visualisation 3D.

1. Récupérer l'archive `vtk-x.x.x.tar.gz` contenant les sources à partir du site [VTK](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Créer un répertoire `build`
6. Dans ce répertoire, configurer, compiler et installer la bibliothèque en utilisant l'outil `cmake`

```
$ cp ~/Téléchargements/vtk-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf vtk-x.x.x.tar.gz
$ cd vtk-x.x.x
$ mkdir build
$ cd build
$ cmake -G"MSYS Makefiles" -DCMAKE_INSTALL_PREFIX=/usr/local -DBUILD_SHARED_LIBS=ON \
  -DCMAKE_BUILD_TYPE=Release -DVTK_USE_QT=ON ..
$ make && make install
```

### 3.19.1 Installation de SLIP

1. Récupérer l'archive `slip-x.x.x.tar.gz` contenant les sources à partir du site [SLIP](#)
2. Copier l'archive dans le répertoire `/usr/local/src`
3. Décompresser l'archive
4. Rentrer dans le répertoire créé
5. Créer un répertoire `build`
6. Dans ce répertoire, configurer, compiler et installer la bibliothèque en utilisant l'outil `cmake`

```
$ cp ~/Téléchargements/slip-x.x.x.tar.gz /usr/local/src
$ cd /usr/local/src
$ tar xvf slip-x.x.x.tar.gz
$ cd slip-x.x.x
$ mkdir build
$ cd build
$ cmake -G"MSYS Makefiles" -DCMAKE_INSTALL_PREFIX=/usr/local -DCMAKE_BUILD_TYPE=Release ..
$ make && make install
```