

# Modification de fichiers GRIB

## Guide d'installation



- **Description**

Ce document présente une liste des composants logiciels à installer ainsi que les fichiers à configurer pour faire fonctionner l'application de modification de fichiers GRIB.



# Prérequis Windows

L'application nécessite le package ecCodes développé par l'ECMWF. Malheureusement, étant donné que cette organisation n'utilise pas Windows dans le cadre de ses travaux il est possible de rencontrer divers problèmes.

Pour cette plateforme il est donc conseillé d'utiliser Linux via une machine virtuelle (VirtualBox, Docker, WSL2 ...).



# Prérequis Linux

L'application a besoin d'un certain nombre de prérequis, notamment un système dont les paquets sont à jour, le langage C, la commande make ainsi que le langage Python accompagné du gestionnaire de paquets pip.

- **Mise à jour des paquets**

```
> sudo apt update && sudo apt upgrade
```

- **Installer le support pour le langage C et make**

```
> gcc --version  
> sudo apt install build-essential
```

Note : le langage C et la commande make sont pré-installés sur certaines distributions Linux.

- **Installer le support pour le langage Python**

```
> python3 --version  
> sudo apt install software-properties-common  
> sudo add-apt-repository ppa:deadsnakes/ppa  
> sudo apt-get update  
> sudo apt install python3.10
```

Note : à l'heure où cette documentation est écrite le module Python pour ecCodes ne supporte pas encore la version 3.11 du langage, ici on privilégie donc la version 3.10.

- **Installer le gestionnaire de paquets pip**

```
> pip --version  
> sudo apt install python3-pip
```

Il nous faut ensuite le package ecCodes, celui-ci nous fournit une API écrite en C et un ensemble d'outils pour lire ou écrire des messages dans les formats GRIB et BUFR.

- **Télécharger le package ecCodes**

<https://confluence.ecmwf.int/display/ECC/Releases>

- **Installer le package ecCodes**

Se placer par exemple dans "/home/utilisateur" et déposer l'archive téléchargée à cet endroit.

```
> tar -xzf eccodes-x.y.z-Source.tar.gz
> mkdir build
> cd build
> cmake -DCMAKE_INSTALL_PREFIX=/chemin/ou/on/installe/eccodes
../eccodes-x.y.z-Source
> make
> ctest
> make install
```

Source : <https://confluence.ecmwf.int/display/ECC/ecCodes+installation>

Note : le build lancé avec "cmake" peut se terminer avec une erreur si la bibliothèque AEC n'est pas présente, dans ce cas il faut rajouter le flag "-DENABLe\_AEC=OFF" pour contourner ce problème.

Maintenant il ne nous reste plus qu'à installer les éléments additionnels relatifs à Python comme Tkinter et les diverses bibliothèques.

- **Installer Tkinter**

```
> python3 -m tkinter
> sudo apt install python3-tk
```

- **Installer NumPy**

```
> pip install --upgrade numpy
```

- **Installer Pillow**

```
> pip install --upgrade Pillow
```

- **Installer Xarray**

```
> pip install --upgrade xarray
```

- **Installer ecmwflibs**

```
> pip install --upgrade ecmwflibs
```

- **Installer ecCodes (module Python)**

```
> pip install --upgrade eccodes
```

- **Installer cfgrib**

```
> pip install --upgrade cfgrib
```

Note : sachant que le module Python ecCodes dépend du package ecmwflibs il faut veiller à respecter l'ordre d'installation suivant : ecmwflibs, ecCodes, cfgrib.

Pour tester les fichiers GRIB il est également recommandé de télécharger XyGrib.

- **Installer XyGrib**

```
> sudo apt install xygrib  
> XyGrib
```

## Téléchargement

L'application est disponible sur le dépôt suivant : <https://github.com/Kevin-LL/MPA>

## Configuration

Après que tout ait été mis en place il reste quelques détails à corriger. Certains fichiers possèdent des chemins vers le package ecCodes, il faut donc les modifier afin que ceux-ci fonctionnent correctement.

- **Modifier les fichiers contenant des chemins vers le package ecCodes**

api/grib\_wind.c (ligne 8)

```
#include "/chemin/installation/include/eccodes.h"
```

api/grib\_wind\_tool.c (ligne 5)

```
#include "/chemin/installation/include/eccodes.h"
```

api/Makefile (ligne 5)

```
$(CC) main.o grib_wind.o grib_wind_tools.o -o prog -lm  
-I/chemin/installation/include/ -L/chemin/installation/lib/ -Wl,-  
rpath,/chemin/installation/lib/ -leccodes
```

Il est également possible de modifier l'emplacement des tuiles de la carte si jamais elles ne sont pas présentes dans "images/tiles".

- **Modifier le chemin vers les tuiles de la carte**

map.py (ligne 68)

```
tile_path =  
Path(f"/chemin/vers/les/tuiles/{self.zoom_level}/{column}/{row}.png")
```

- **Lancer l'application**

Se placer à la racine du projet.

```
> python3 main.py
```

Normalement on devrait obtenir le résultat suivant :

