

# 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.

# Téléchargement

Le projet est disponible sur le dépôt suivant : <https://github.com/Kevin-LL/MPA>



## Installation Windows

### • Installer GCC et make

Pour pouvoir compiler un programme C sous Windows, il va nous falloir MinGW-w64. Il s'agit d'une version minimaliste de GCC pour Windows.

Pour télécharger MinGW-w64, il faut se rendre ici : <https://www.mingw-w64.org/downloads/>, puis aller dans la section "Mingw-builds" qui mène vers un dépôt GitHub.

Dans les releases, il faut choisir la bonne version de MinGW-w64. Par exemple, pour un Windows 64 bits on part sur une architecture "x86\_64", threads "posix", exception "seh", runtime "ucrt". Note : le runtime "ucrt" n'est disponible que depuis Windows 10, pour les versions plus anciennes de Windows il faudra utiliser "msvcrt".

Depuis l'archive 7zip, on extrait généralement le dossier "mingw64" dans "C:\" ou "C:\Program Files\".

Enfin, il faut modifier les variables d'environnement. Via les touches "Windows + R" on ouvre "sysdm.cpl". Dans "Paramètres système avancés" on clique sur "Variables d'environnement...". Dans la liste des variables système on sélectionne "Path" puis on clique sur "Modifier...", ici on ajoute le chemin absolu vers le dossier "mingw64\bin".

Si tout s'est bien déroulé, les commandes suivantes devraient retourner un résultat dans un terminal Cmd ou Powershell :

```
> gcc --version  
> mingw32-make --version
```

### • Installer Miniconda et créer l'environnement Conda

Il nous faut ensuite installer Miniconda. Il s'agit d'un gestionnaire de paquets et un système de gestion d'environnements.

Pour télécharger Miniconda, il faut se rendre ici : <https://docs.conda.io/projects/miniconda/en/latest/>, puis on sélectionne "Miniconda3 Windows 64-bit" et on suit les étapes d'installation.

Une fois l'installation terminée, on ouvre "Anaconda Prompt" ou "Anaconda Powershell Prompt". Via le terminal, on se rend à la racine du projet où se trouve le fichier "environment.yml", puis on exécute la commande :

```
> conda env create -f environment.yml
```

### • Configurer et compiler l'API

Sur le projet, dans "src\api" se trouve l'API qui permet de modifier les fichiers GRIB. Pour pouvoir la compiler il faut tout d'abord éditer "eccodes\_dir.txt", ce fichier texte doit contenir une seule ligne qui est le chemin absolu vers l'installation d'ecCodes. Dans notre cas il faut indiquer le chemin vers le dossier "miniconda3\envs\grib-wind\Library".

Ensuite, toujours depuis "src\api", on lance la commande :

```
> mingw32-make
```

Si jamais un problème survient on peut toujours nettoyer les fichiers générés :

```
> mingw32-make clean
```

### • Activer l'environnement Conda et lancer l'application

Pour pouvoir lancer l'application, il faut activer notre environnement Conda depuis un terminal "Anaconda Prompt" ou "Anaconda Powershell Prompt" :

```
> conda activate grib-wind
```

L'environnement courant est précisé entre parenthèses devant chaque ligne du terminal. Lorsque "grib-wind" est actif, depuis la racine du projet on peut faire la commande :

```
> python main.py
```



## Installation Linux

### • Mise à jour des paquets

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

## • Installer GCC et make

Pour compiler un programme C il nous faut GCC et make. Ils sont pré-installés sur la plupart des distributions Linux. Afin de vérifier leur installation on lance les commandes suivantes depuis le terminal :

```
> gcc --version  
> make --version
```

Si GCC et make ne sont pas installés :

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

## • Installer Miniconda et créer l'environnement Conda

Il nous faut ensuite installer Miniconda. Il s'agit d'un gestionnaire de paquets et un système de gestion d'environnements.

Pour télécharger Miniconda, il faut se rendre ici :

<https://docs.conda.io/projects/miniconda/en/latest/>, puis on sélectionne "Miniconda3 Linux 64-bit".

Lorsque le téléchargement est terminé, on exécute la commande suivante depuis l'emplacement du fichier et on suit les étapes d'installation :

```
> bash Miniconda3-latest-Linux-x86_64.sh
```

Une fois l'installation terminée, via le terminal, on se rend à la racine du projet où se trouve le fichier "environment.yml", puis on exécute la commande :

```
> conda env create -f environment.yml
```

## • Configurer et compiler l'API

Sur le projet, dans "src/api" se trouve l'API qui permet de modifier les fichiers GRIB. Pour pouvoir la compiler il faut tout d'abord éditer "eccodes\_dir.txt", ce fichier texte doit contenir une seule ligne qui est le chemin absolu vers l'installation d'ecCodes. Dans notre cas il faut indiquer le chemin vers le dossier "miniconda3/envs/grib-wind".

Ensuite, toujours depuis "src/api", on lance la commande :

```
> make
```

Si jamais un problème survient on peut toujours nettoyer les fichiers générés :

```
> make clean
```

- **Activer l'environnement Conda et lancer l'application**

Pour pouvoir lancer l'application, il faut activer notre environnement Conda depuis le terminal :

```
> conda activate grib-wind
```

L'environnement courant est précisé entre parenthèses devant chaque ligne du terminal. Lorsque "grib-wind" est actif, depuis la racine du projet on peut faire la commande :

```
> python main.py
```

## Annexes

- **Installer XyGrib**

Pour tester les fichiers GRIB, il est recommandé d'utiliser XyGrib :

<https://opengribs.org/fr/téléchargements>

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

Pour modifier le dossier dans lequel sont chargées les tuiles de la carte, il faut se rendre dans "src/map.py" (ligne 68)

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