

Utiliser Code avec GoaS

Chantal Pic

5 août 2019

Ce court document vise à introduire l'utilisation de Code (Visual Studio Code sur Linux) avec GoaS, en compilant avec Intel ou avec gfortran sur une station de travail.

Ne pas oublier que la version Linux de Visual Studio (Code) n'offre pas toutes les fonctionnalités de celle sur Windows. En ce sens, Code est plutôt un éditeur avancé, alors que Visual Studio est un véritable environnement de développement intégré.

1 Démarrer code et compiler GoaS

1. Déplacez-vous dans le répertoire principal de Goas.
2. Comme pour une compilation sur la ligne de commande, assurez-vous tout d'abord de charger la bonne configuration, selon la version du compilateur Intel utilisée :

```
. .ssmuse_intel_16
ou
. .ssmuse_intel_19
ou
. .ssmuse_gfortran_5.1.0
ou
. .ssmuse_gfortran_7.4.0.0
```
3. La configuration de défaut utilise gfortran. Si vous souhaitez utiliser le compilateur Intel :
 - éditez le fichier *sources/CMakeLists.txt*,
 - au besoin, décommentez la ligne suivante en enlevant le # en avant :

```
# set(COMPILER intel CACHE INTERNAL compiler)
```
 - au besoin, commentez la ligne suivante en ajoutant un # en avant :

```
set(COMPILER gfortran CACHE INTERNAL compiler)
```
4. Démarrez Code en utilisant le script suivant :

```
./vis-studio
```
5. Code cherche à configurer GoaS dès le démarrage. Il va par contre demander où se trouvent les compilateurs :
 - sélectionnez *[Unspecified]* à la fin de la liste présentée,
 - les commandes de configuration devraient défiler en bas de la fenêtre de Code.

6. Il se peut que de nombreuses fenêtres s'ouvrent pour poser des questions de toutes sortes, notamment pour envoyer les erreurs à Microsoft : refusez les demandes ou ignorez-les !



7. Cliquez ensuite sur *Build* dans la bordure inférieure de Code : la compilation démarre et les commandes défilent dans le bas de la fenêtre.
8. Une autre façon de configurer puis de compiler GoaS est la suivante :



Une fois Code démarré, cliquez sur l'icône de CMake, dans la barre de menus située à gauche, où il est la dernière de la liste : vous pouvez placer le curseur au-dessus de l'icône quelques secondes et vous verrez l'étiquette *CMake*.



Vous avez ainsi accès, dans la deuxième colonne, en haut, à deux petites icônes : celle de CMake à nouveau, pour configurer, puis un marteau, pour compiler. Trois petits points mènent à des fonctions de *clean*.

9. Tapez *Ctrl+Maj+P*, ou sélectionnez le menu *Afficher, Palette de commandes*, et sélectionnez ou tapez *Cmake : Install* pour installer dans le répertoire de travail.
10. Cliquez sur *Terminal* dans la fenêtre inférieure pour ouvrir un terminal dans Code.
11. Déplacez-vous dans le répertoire de travail pour tourner GEM (voir le fichier EC.README pour plus de détails).

2 Développer avec Code

Plusieurs extensions utiles au développement sont préinstallées et chargées lorsque vous démarrez Code. Vous pouvez en consulter les détails soit en cliquant sur l'avant-dernier icône dans la barre de menus située à gauche, soit par le menu *Affichage, Extensions* :

- C/C++ ms-vscode.cpptools, Microsoft. C/C++ IntelliSense, debugging, and code browsing.
- CMake twxs.cmake, twxs. CMake langage support for Visual Studio Code
- CMake Tools vector-of-bool.cmake-tools, vector-of-bool. Extended CMake support in Visual Studio Code
- French Language Pack for Visual Studio Code, ms-ceintl.vscode-language-pack-fr, Microsoft. Language pack extension for French
- Modern Fortran, krvajalm.linter-gfortran, Miguel Carvajal. Modern Fortran language support, including syntax highlighting and error detection
- Git History, donjayamanne.githistory Don Jayamanne. View git log, file history, compare branches or commits
- Git Graph, mhutchie.git-graph, mhutchie. View a Git Graph of your repository, and perform Git actions from the graph.

(Git Graph et Git History sont accessibles par la commande *Ctrl+Maj+P* ou en sélectionnant le menu *Afficher, Palette de commandes*.)

Ces extensions sont installées dans le répertoire */users/dor/armn/cpi/ords/code-extensions*. Au besoin, vous pouvez en copier le contenu dans un répertoire vous appartenant afin de pouvoir en ajouter. N'oubliez pas alors de modifier le fichier *vis-studio* dans le répertoire principal de GoaS pour modifier le chemin du répertoire d'extensions.

Il y a des extensions intéressantes pour l'interface avec git, telles que *GitLens*, mais elles nécessitent une version plus récente de git que celle qui est installée sur les stations de travail.

Pour en apprendre plus sur la configuration et l'utilisation de Code, il y a plusieurs vidéos d'introduction bien faites. Au démarrage de Code, la page principale propose des liens vers ces vidéos, ainsi que vers un *Terrain de jeu interactif*. Voir le site web de Code : <https://code.visualstudio.com/docs>.