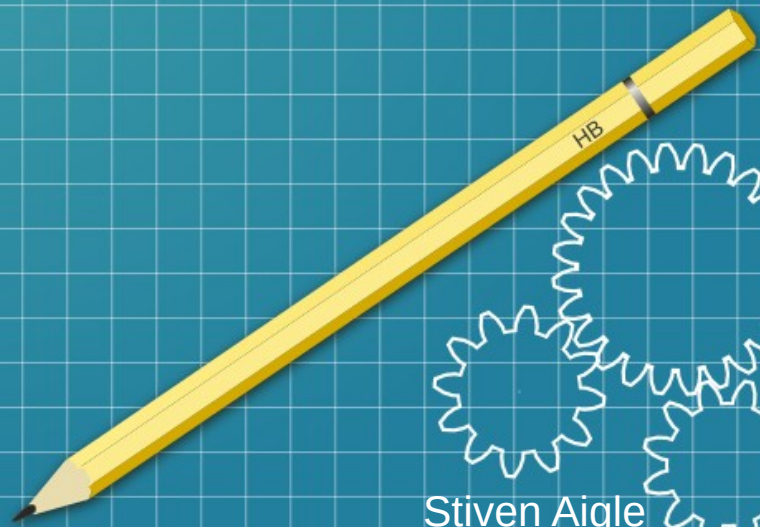
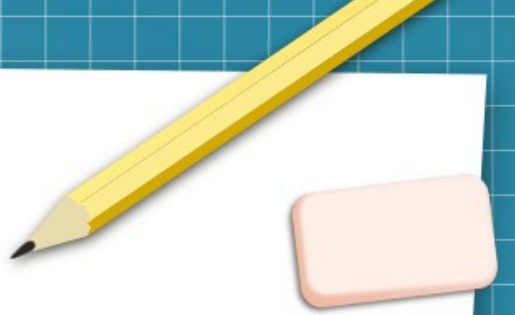
Compiler, tester et packager ses applications



Stiven Aigle

Introduction

- Compiler → produire un exécutable, une bibliothèque
- Tester → s’assurer que l’application fonctionne correctement
- Packager → Créer un paquet rendant l’application facilement utilisable



- Générateur de « build system »
- Développer par Kitware depuis 2000
- Version actuelle 3.21.2 (21 septembre 2021)
- « Modern CMake » depuis 3.0
- Multi-plateforme
- Langage supporté : C, C++, C#, CUDA, …

https://cmake.org/cmake/help/latest/command/project.html#options

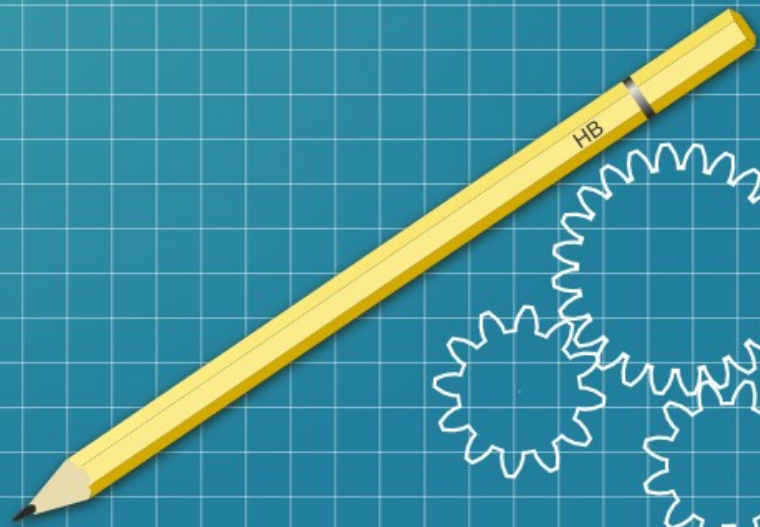

NETFLIX


Qt

Plan

- Compiler via CMake
- Configurer via CMake
- Tester via CTest
- Package via CPack

Compiler via CMake



Étapes

1) Configuration

2) Génération

3) Compilation

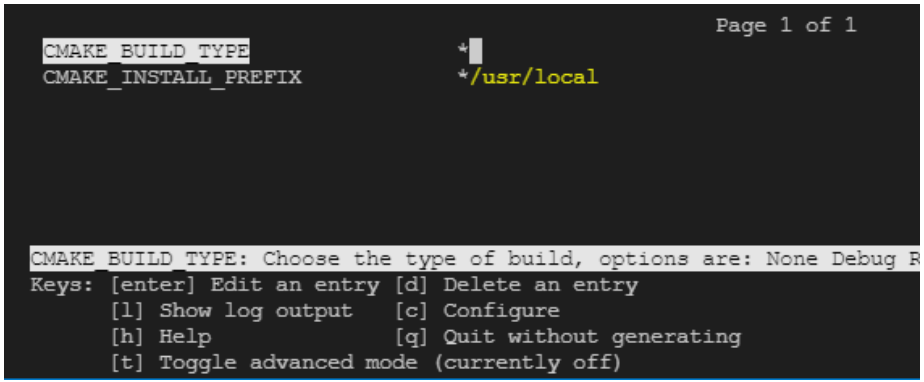
CMake

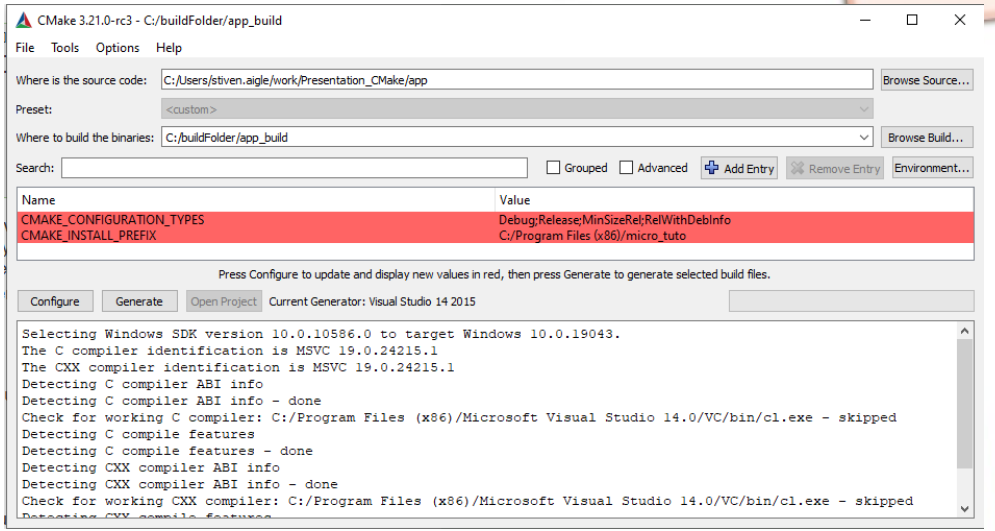
- cmake -S <source_folder> -B <build_folder>
- Options :
  - -G <generator> (Unix Makefiles, Xcode, Visual Studio, …)
  - -D<OPTIONS>=<value>
    - Exemple : -DCMAKE_BUILD_TYPE=Debug
  - ….

https://cmake.org/cmake/help/latest/manual/cmake.1.html

CCMake ou CMake-GUI

- ccmake : GUI par ncurses
- cmake-GUI : GUI par Qt

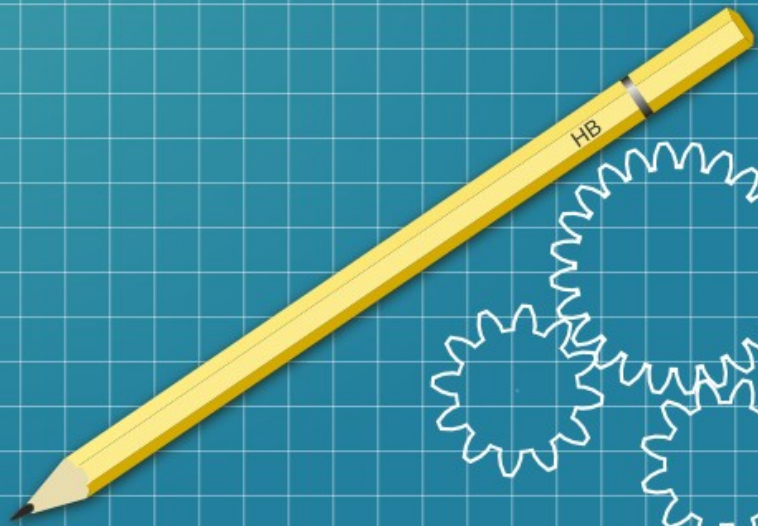




Compiler

- cmake --build <build_folder>
- Avec les outils habituels :
  - ninja
  - make
  - Visual Studio
  - Xcode
  - ...

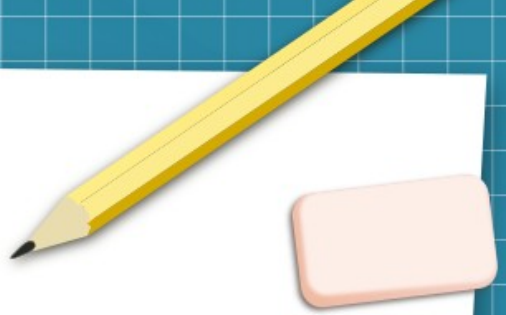
Configurer via CMake



Les bases

- Le code s’écrit dans un fichier CMakeLists.txt
- Doit contenir :
  - cmake_minimum_required(VERSION X.YY)
    https://cmake.org/cmake/help/latest/command/cmake_minimum_required.html
  - project(<project_name>)
    https://cmake.org/cmake/help/latest/command/project.html?highlight=project
- Doit être traiter comme un code source classique

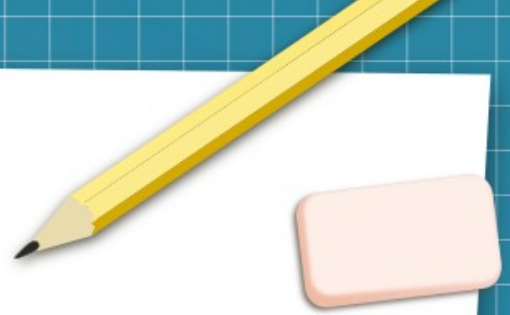
Définir un exécutable



- add_executable(<executable_name> ….<source_files>)
  https://cmake.org/cmake/help/latest/command/add_executable.html?highlight=add_executable
  - Crée une target
- target_sources(<target_name> PRIVATE <source_files>)

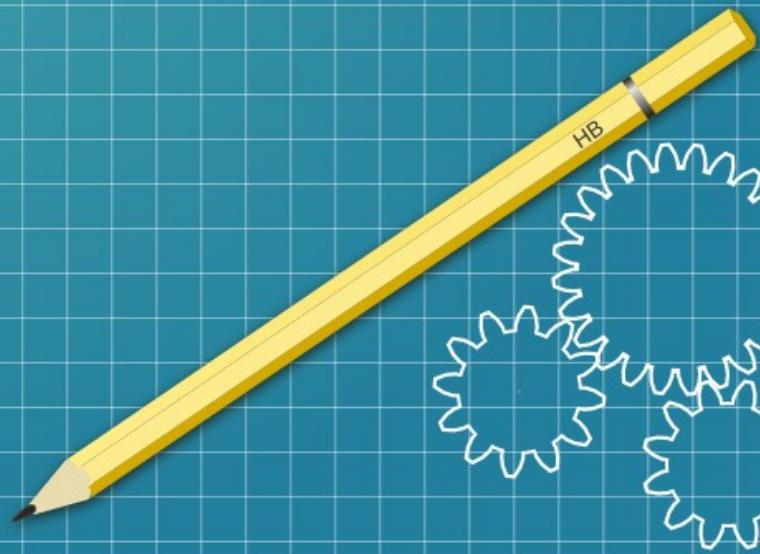
  https://cmake.org/cmake/help/latest/command/target_sources.html#command:target_sources

Target CMake

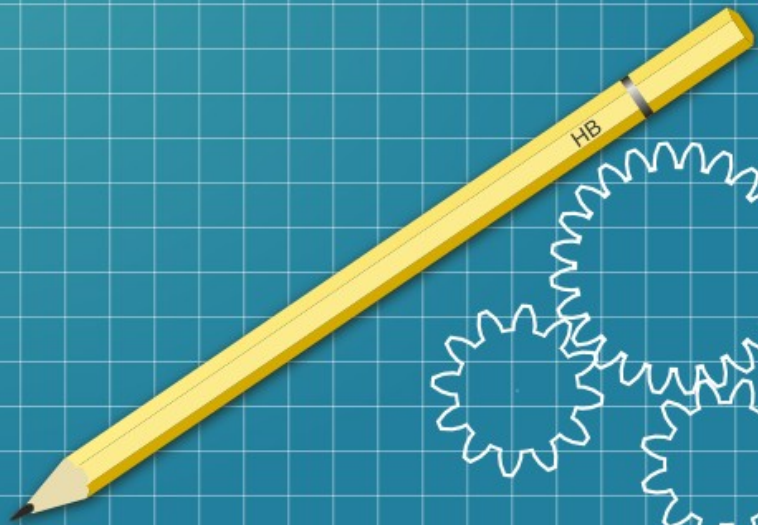


- Créé par add_executable, add_library, add_custom_target, ...
- Possède des propriétés
  https://cmake.org/cmake/help/latest/manual/cmake-properties.7.html#properties-on-targets
  - Décrivent les options de compilations, de linkage, les includes directories,...
  - Peuvent être transmise ou non (PUBLIC, PRIVE)

Tester via CTest



Packager via CPack



Sources

- Professional CMake, a practical guide. Craig Scott
- https://cmake.org/documentation/
- https://cliutils.gitlab.io/modern-cmake/
- https://pabloariasal.github.io/2018/02/19/its-time-to-do-cmake-right/
- https://www.youtube.com/watch?v=y7ndUhdQuU8
- https://www.youtube.com/watch?v=y9kSr5enrSk
- https://www.youtube.com/watch?v=m0DwB4OvDXk

QUESTIONS ?