



CMake

An introduction

Karstein Brynestad (许云深) 2016400428

Department of Industrial Engineering, Tsinghua University

Industrial Economics and Technology Management, NTNU



What is CMake?



CMake is an open-source, cross-platform family of tools designed to build, test and package software.

”

Consider it a “makefile maker” or a “translator”, if you will.
CMake lets you use the native development tools you love the most.

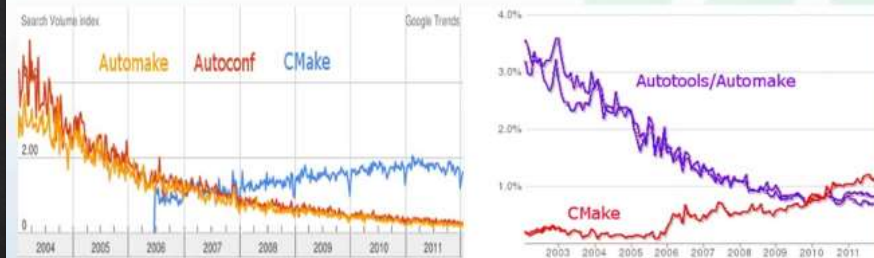


Initiated early 2000



Why CMake? Everyone is using it

KDE 2006 – Tipping Point!



- Google Search Trends and ohloh comparisons with auto*
- 1200+ downloads per day from www.cmake.org
- Major Linux distributions and Cygwin provide CMake packages
- KDE, Second Life, Boost (Experimentally), many others



Source: "Introduction to CMake" video, <https://vimeo.com/32212195>

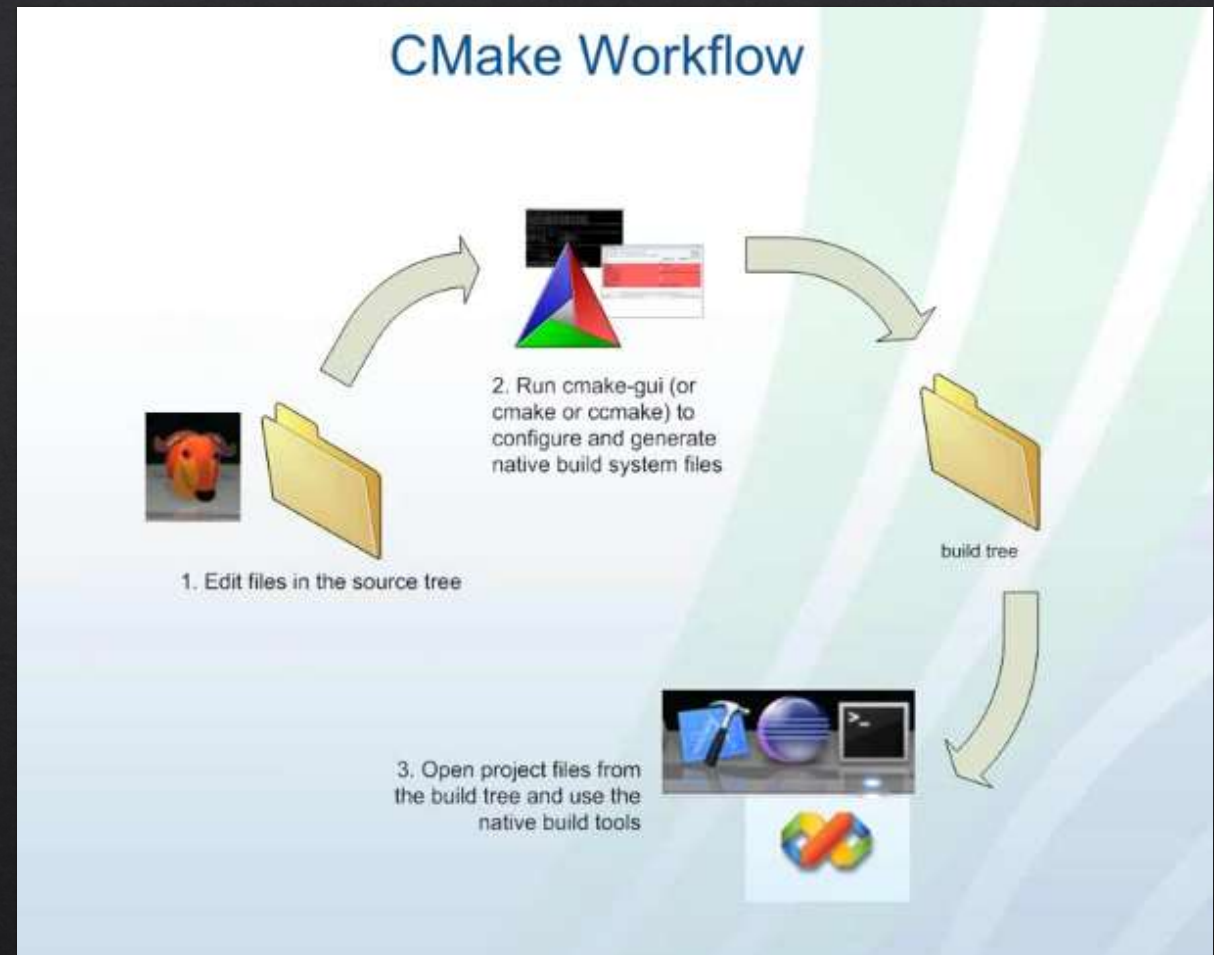


Supported compilers

- MinGW
- NMake
- Unix
- Ninja
- Borland
- MSYS
- Watcom
- Green Hills (work-in-progress)

Supported tools

- Visual Studio (2003-2017)
- For MinGW, NMake, Unix, Ninja:
- CodeBlocks
 - CodeLite
 - Sublime Text 2
 - Kate
 - Eclipse CDT4



Source: "Introduction to CMake" video, <https://vimeo.com/32212195>



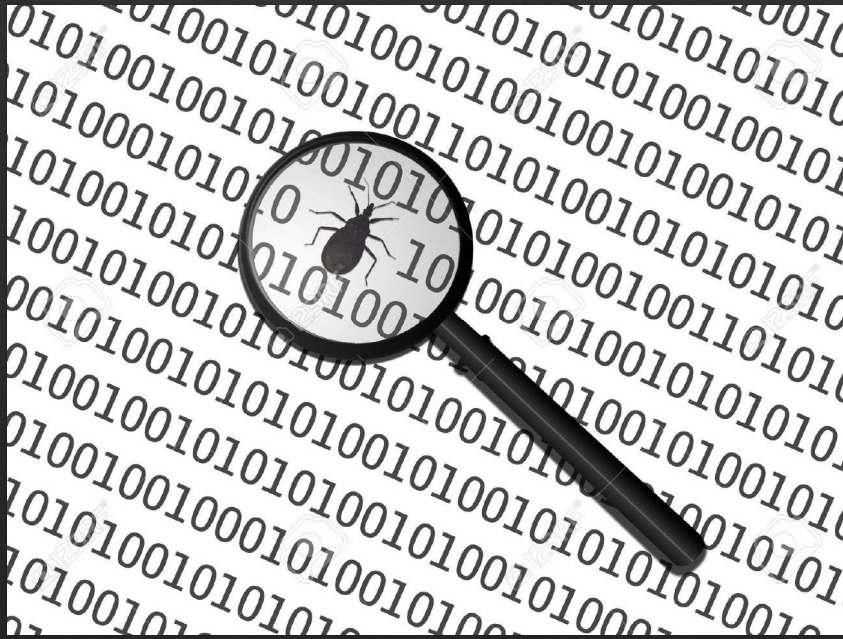
CMake concepts

- ◆ Directory
 - ◆ As usual, where you store your files
- ◆ Source tree
 - ◆ Plain text, the files you write
- ◆ Binary tree
 - ◆ Build tree, made by CMake
- ◆ Generator
 - ◆ Unix, Visual Studio etc
- ◆ Variable
 - ◆ Always a string
- ◆ Command
 - ◆ What to do?
- ◆ Target
 - ◆ The product of your build. Lib, executable etc
- ◆ Others
 - ◆ Single regression test with expected output
 - ◆ Property
 - ◆ Install tree
 - ◆ Installer / Package
 - ◆ Cache entry / Cache variable



Why do we care?

Build project files for your fav tool



Adaptable to different users



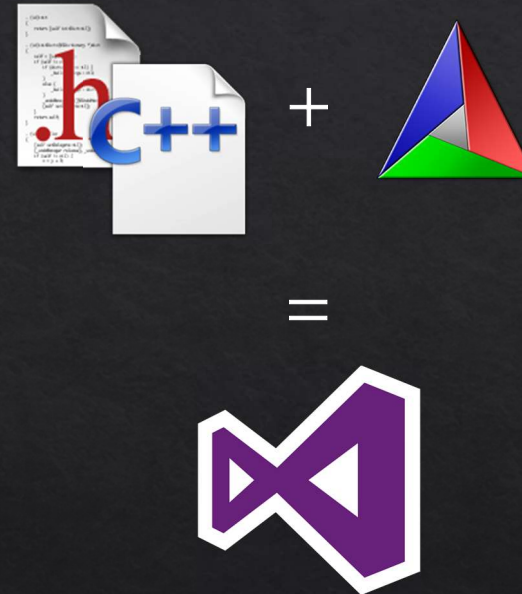


A simple example

```
#CmakeLists.txt for HW02 task 2.1
cmake_minimum_required (VERSION 3.8)
set(SRC
    Source/main.cpp
    Source/reviewerassigner.cpp
)
set(HDR
    Source/reviewerassigner.h
)
project(Reviewer)
add_executable(Reviewer ${SRC} ${HDR})
```

```
\Project\Lecture\Code>cd Build
```

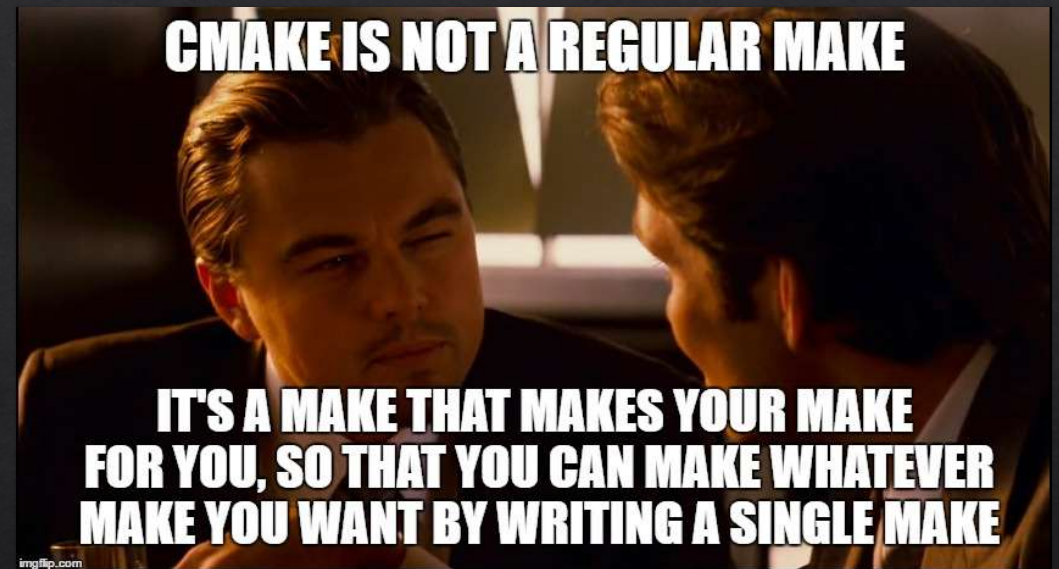
```
\Project\Lecture\Code\Build>cmake .. -G "Visual Studio 14"
```





What to remember

- ❖ CMake doesn't "make" or "build" the program for you – it generates the build or makefile of your choice
- ❖ Separate your source tree from your build tree
- ❖ Behaves differently for different generators
- ❖ Make sure the tool/compiler you are generating for is already installed!



Sources:

"CMake overview": <https://cmake.org/overview/>

"CMake tutorial": <https://cmake.org/cmake-tutorial/>

"Introduction to CMake" video: <https://vimeo.com/32212195> (VPN Required)