CMake course for Elbit-Elisra

Alex Kushnir

CLARITUNE
Making Things Happen

Summary of the first day

What CMake is?

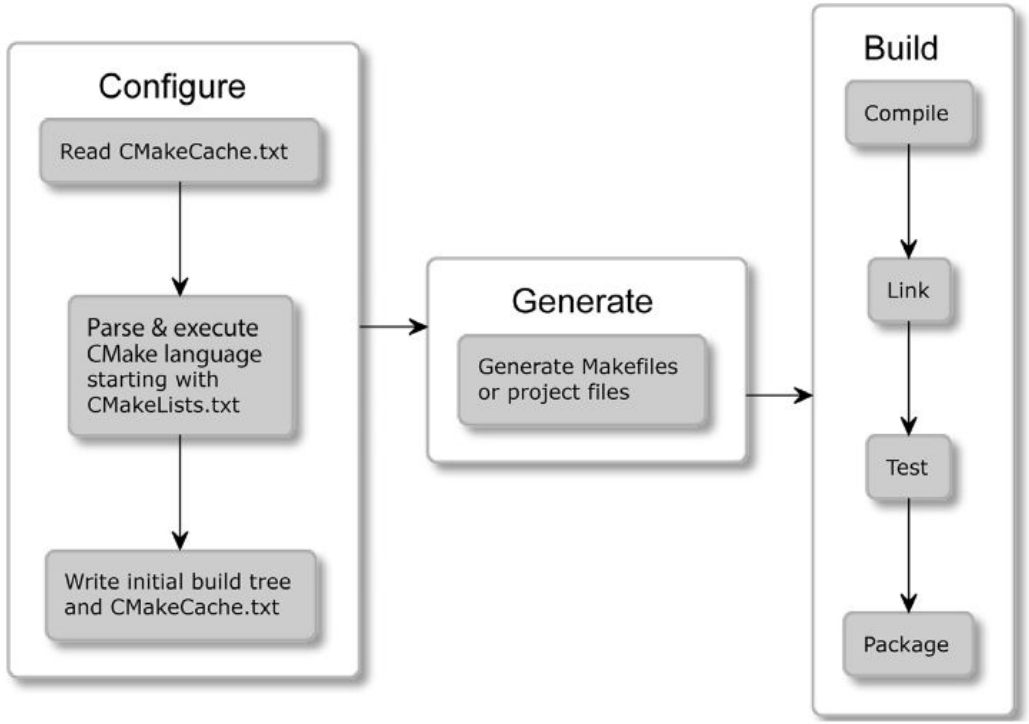
- A family of tools designed to build, test and package software.
- Cross-platform (Windows/Linux/macOS/Cygwin).
- Most modern compilers and toolchains are supported.
- Able to generate project files for all popular IDEs (Visual Studio, Eclipse, CLion, etc.)
- Able to generate Makefiles and ninjafiles
- Rich ecosystem.
- Old features get deprecated to keep CMake lean.
- Developed and maintained by Kitware Inc.

Motivation

- Managing artifacts creation is complicated
- Dependent on environment
- Abstraction hiding complexities
- We never want to “reinvent the wheel”
- Can also fetch, test and install

How it works?

- The process of building has 3 stages
  - Configuration
  - Generation
  - Building



Generators

Linux

```
Generators

The following generators are available on this platform (* marks default):
  Green Hills MULTI           = Generates Green Hills MULTI files
                                (experimental, work-in-progress).
* Unix Makefiles              = Generates standard UNIX makefiles.
  Ninja                       = Generates build.ninja files.
  Ninja Multi-Config          = Generates build-<Config>.ninja files.
  Watcom WMake                = Generates Watcom WMake makefiles.
  CodeBlocks - Ninja          = Generates CodeBlocks project files
                                (deprecated).
  CodeBlocks - Unix Makefiles = Generates CodeBlocks project files
                                (deprecated).
  CodeLite - Ninja            = Generates CodeLite project files
                                (deprecated).
  CodeLite - Unix Makefiles   = Generates CodeLite project files
                                (deprecated).
  Eclipse CDT4 - Ninja        = Generates Eclipse CDT 4.0 project files
                                (deprecated).
  Eclipse CDT4 - Unix Makefiles= Generates Eclipse CDT 4.0 project files
                                (deprecated).
  Kate - Ninja                = Generates Kate project files (deprecated).
  Kate - Ninja Multi-Config   = Generates Kate project files (deprecated).
  Kate - Unix Makefiles       = Generates Kate project files (deprecated).
  Sublime Text 2 - Ninja      = Generates Sublime Text 2 project files
                                (deprecated).
  Sublime Text 2 - Unix Makefiles
                              = Generates Sublime Text 2 project files
                                (deprecated).
```

Printed as part of cmake --help

Windows

```
Generators

The following generators are available on this platform (* marks default):
* Visual Studio 17 2022        = Generates Visual Studio 2022 project files.
                                 Use -A option to specify architecture.
  Visual Studio 16 2019        = Generates Visual Studio 2019 project files.
                                 Use -A option to specify architecture.
  Visual Studio 15 2017 [arch] = Generates Visual Studio 2017 project files.
                                 Optional [arch] can be "Win64" or "ARM".
  Visual Studio 14 2015 [arch] = Generates Visual Studio 2015 project files.
                                 Optional [arch] can be "Win64" or "ARM".
  Borland Makefiles            = Generates Borland makefiles.
  NMake Makefiles              = Generates NMake makefiles.
  NMake Makefiles JOM          = Generates JOM makefiles.
  MSYS Makefiles               = Generates MSYS makefiles.
  MinGW Makefiles              = Generates a make file for use with
                                 mingw32-make.
  Green Hills MULTI            = Generates Green Hills MULTI files
                                 (experimental, work-in-progress).
  Unix Makefiles               = Generates standard UNIX makefiles.
  Ninja                        = Generates build.ninja files.
  Ninja Multi-Config           = Generates build-<Config>.ninja files.
  Watcom WMake                 = Generates Watcom WMake makefiles.
  CodeBlocks - MinGW Makefiles = Generates CodeBlocks project files
                                 (deprecated).
  CodeBlocks - NMake Makefiles = Generates CodeBlocks project files
                                 (deprecated).
  CodeBlocks - NMake Makefiles JOM
                               = Generates CodeBlocks project files
                                 (deprecated).
  CodeBlocks - Ninja           = Generates CodeBlocks project files
                                 (deprecated).
  CodeBlocks - Unix Makefiles  = Generates CodeBlocks project files
                                 (deprecated).
  CodeLite - MinGW Makefiles   = Generates CodeLite project files
                                 (deprecated).
  CodeLite - NMake Makefiles   = Generates CodeLite project files
                                 (deprecated).
  CodeLite - Ninja             = Generates CodeLite project files
                                 (deprecated).
  CodeLite - Unix Makefiles    = Generates CodeLite project files
                                 (deprecated).
  Eclipse CDT4 - NMake Makefiles
                               = Generates Eclipse CDT 4.0 project files
                                 (deprecated).
  Eclipse CDT4 - MinGW Makefiles
                               = Generates Eclipse CDT 4.0 project files
                                 (deprecated).
  Eclipse CDT4 - Ninja         = Generates Eclipse CDT 4.0 project files
                                 (deprecated).
  Eclipse CDT4 - Unix Makefiles= Generates Eclipse CDT 4.0 project files
                                 (deprecated).
  Kate - MinGW Makefiles       = Generates Kate project files (deprecated).
  Kate - NMake Makefiles       = Generates Kate project files (deprecated).
  Kate - Ninja                 = Generates Kate project files (deprecated).
  Kate - Ninja Multi-Config    = Generates Kate project files (deprecated).
  Kate - Unix Makefiles        = Generates Kate project files (deprecated).
  Sublime Text 2 - MinGW Makefiles
                               = Generates Sublime Text 2 project files
                                 (deprecated).
  Sublime Text 2 - NMake Makefiles
                               = Generates Sublime Text 2 project files
                                 (deprecated).
  Sublime Text 2 - Ninja       = Generates Sublime Text 2 project files
                                 (deprecated).
  Sublime Text 2 - Unix Makefiles
                               = Generates Sublime Text 2 project files
                                 (deprecated).
```

Basic Example

```
cmake_minimum_required(VERSION 3.28)
set(CMAKE_CXX_STANDARD 23)

project(Calculator)

add_executable(calc Main.cpp Calculator.cpp)
```

Glossary

- **Source tree**
- **Build tree**
- **Project file**
- **Cache file**
- **Generated files**
- **Preset files**
- **Package definition file**

Topics Brief – First Day

- CMake Commands (slides 29 – 38)
- Variables and Lists (slides 39 – 51)
- Scripting: Conditions (slides 52 – 57)
- Scripting: Loops (slides 58 – 62)
- Scripting: Functions and Macros (slides 63 – 73)
- Partitioning the Project (slides 78 – 83)
- Using the Project Environment (slides 84 – 86)
- Configuring the Toolchain (slides 87 – 95)
- CMake Targets (slides 98 – 109)
- Compilation and Linking (slides 110 – 155)
  - Compiler options configuration (slides 120 - 121)
  - Adding source files to targets (slides 122 – 127)
  - Preprocessor configuration (slide 128)
  - Shared and Static Libraries (slides 147 – 149)

Topics Brief – Second Day

- Warm-up and alignment (this presentation + exercises)
- Managing dependencies with Cmake (slide numbers)
- Writing CMake presets (slide numbers)
- Testing with CMake (slide numbers)

Conan vs. CMake FetchContent module

Feature	Conan + find_package	FetchContent
Dependency Handling	Manages external libraries and dependencies	Downloads and integrates projects directly
Use Case	Best for public and mature dependencies Able to fetch binary artifacts	Ideal for internal or actively developed projects
Integration	Provides dependency info directly to CMake	Integrates with CMake build system
Cross-Compilation	Strong support for cross-compiling	Limited cross-compilation support

Presets – Top Level

```
{
  "version": 6,
  "cmakeMinimumRequired": {
    "major": 3,
    "minor": 26,
    "patch": 0
  },
  "include": [],
  "configurePresets": [],
  "buildPresets": [],
  "testPresets": [],
  "packagePresets": [],
  "workflowPresets": [],
  "vendor": {
    "data": "IDE-specific information"
  }
}
```

Today!

- Program-analysis tools
- Installation and packaging
- Your questions!