(slide intentionally left blank)

CMAKE 101

FROM SOURCE TO INSTALLER

Jakub Neruda | Safetica

1. INTRODUCTION (01:30)

WHAT WILL BE IN THIS TALK?

- CMake basics
  - From a library source to installer
- Tips and tricks
- github.com/nerudaj/CMakeTalk

WHAT WILL NOT BE IN THIS TALK?

- Dependency management (maybe a little bit)
- Generator expressions
- Programming using CMake

WHY CMAKE?

AutoTools

Visual Studio

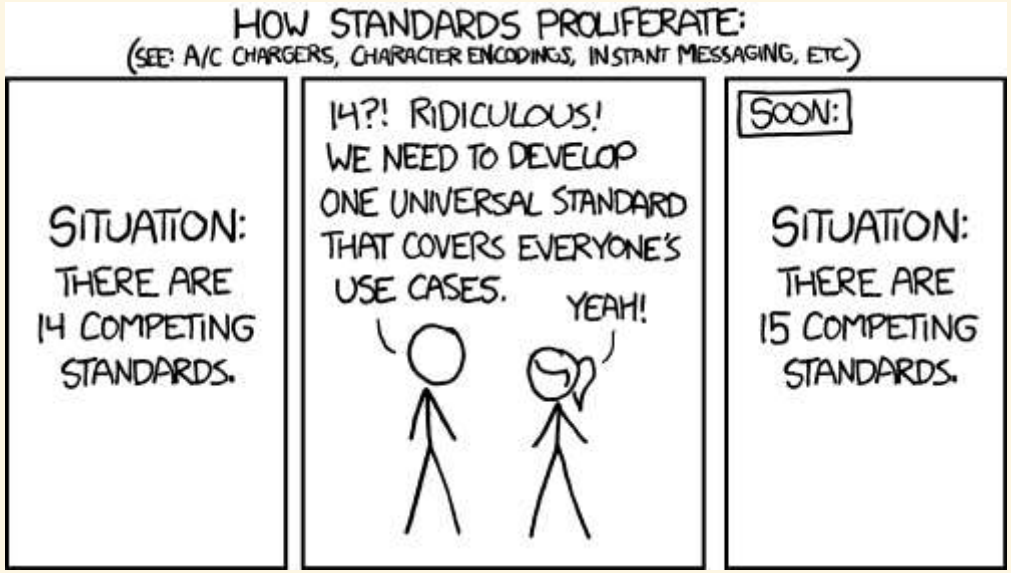
clang

CMake

CONAN.io

C/C++ Package Manager

NOT ANOTHER COMPETING STANDARD



UNIVERSAL CI

```
cmake ..
cmake --build . --c Release
ctest
cpack
```

Pics or didnt happen

2. BASICS (07:00)

CMAKE PHILOSOPHY

- Makefile generator
  - Customization through variables
- Out-of-source build
- Projects vs targets
- CMakeLists.txt is a project file
- .cmake is an includable script

BASIC WORKFLOW

```
mkdir _build
cd _build
cmake .. [-DOPTIONAL_PARAM=ON]
cmake --build . --config <Debug|Release|Whatever> -j 8
ctest -C <Debug|Release>
cpack -C <Debug|Release>
```

EVERYTHING IS A FUNCTION

```
set ( VARNAME VALUE_1 ... VALUE_N )

function ( FUNCNAME PARAM_1 ... PARAM_N )
# ...
endfunction()

if ( ${CMD_OPTION} )
# ...
else()
# ...
endif()
```

EVERYTHING IS A LIST OF STRINGS

```
set ( PROJECT_PARAMS FIRST SECOND PARAM )
myfoo ( ${PROJECT_PARAMS} ) # calling myfoo with three params
myfoo ( "${PROJECT_PARAMS}" ) # calling it just one param
```

3. DEMO TIME (12:00)

4. TRICKS (35:00)

CORE VARIABLES

```
CMAKE_CURRENT_SOURCE_DIRECTORY
CMAKE_SOURCE_DIRECTORY
PROJECT_SOURCE_DIRECTORY
CMAKE_BINARY_DIRECTORY (and others...)
PROJECT_IS_TOP_LEVEL
```

CONFIGURE_FILE

```
#pragma once // MyHeader.hpp.in

#include <filesystem>

const std::filesystem SRC_DIR = "@CMAKE_CURRENT_SOURCE_DIR@";
```

```
configure_file (
    "${CMAKE_CURRENT_SOURCE_DIR}/tmp/MyHeader.hpp.in"
    "${CMAKE_CURRENT_SOURCE_DIR}/include/MyHeader.hpp"
)
```

SOURCE_GROUP

```
set ( HEADERS
    "${CMAKE_CURRENT_SOURCE_DIR}/include/FileA.hpp"
    "${CMAKE_CURRENT_SOURCE_DIR}/include/Subfolder/FileB.hpp"
    "${CMAKE_CURRENT_SOURCE_DIR}/include/SubfolderB/FileC.hpp"
)

source_group (
    TREE "${CMAKE_CURRENT_SOURCE_DIR}"
    FILES ${HEADERS}
)
```

GLOBBING

```
file ( GLOB_RECURSE
    HEADERS # <-- target variable
    "${CMAKE_CURRENT_SOURCE_DIR}/include**/*.hpp"
)
```

OVERRIDE RUNTIME DIRECTORY

```
set (
    CMAKE_RUNTIME_OUTPUT_DIRECTORY
    "${CMAKE_BINARY_DIR}/Compiled"
)
```

- Puts all executables and dynamic libraries into the same folder

5. CLOSING THOUGHTS (43:00)

FURTHER READING

- Modern CMake best practices - cliutils.gitlab.io
- Dependency Management in CMake (medium.com/@nerudaj)
- CMake init
- Official docs

WHERE I CAN FIND THIS TALK?

github.com/nerudaj/CMakeTalk