

**ACCU
2022**

THINGS I LEARNT WHILE TRYING TO AVOID BECOMING A CMAKE EXPERT

CB BAILEY

Things I Learned While Trying To Avoid Becoming a CMake Expert

CB Bailey

 @hashpling

Bloomberg

Wednesday, 6th April 2022

Avoid Becoming a CMake Expert

CB Bailey

 @hashpling

they/them

CMake – what and why?

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering

Why might we want to extend?

A few examples:

- “Compiling” application data
- Code generation
- Simplifying project-specific boilerplate

Getting Started

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering



Phases of build

- Configuration
- Build



Configuration

- Sequential interpretation of `CMakeLists.txt`
- Creates build tree and writes out build instructions

Build

- Runs the build tool (e.g. `make` or `ninja`)
- Potential for high concurrency

A minimal CMakeLists.txt

```
cmake_minimum_required(VERSION 3.19)

project(MyProject LANGUAGES CXX)

add_executable(hello main.cpp)
```

Demo Time

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering



That didn't seem so bad?

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering

Structuring CMake

- “targets” - same terminology as make
- ... `add_library`, `add_executable`
- directories
- ... `add_subdirectory`

Tips

- Choose a consistent structure
- Name your targets after their output
- Keep your subdirectories portable

Tips

- Use “out of source” builds
- Try mounting your source tree read-only

Linking dependencies

- `target_link_libraries` is the most common
- **but** `add_dependencies` also exists

Demo Time linking

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering

Logic and State

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering

Variables

- Regular variables
- Cache variables
- Environment variables

```
set (MyVar "A value")  
  
message (STATUS "MyVar = \"${MyVar}\"")
```

Regular variables

- Have dynamic scope (more on that later)
- Not preserved from run to run

Cache

- Have global scope
- Preserved from run to run
- Stored in `CMakeCache.txt`

Environment

- Inherited from the calling environment
- Not preserved between runs...
- ...but might affect settings in cache variables
- Doesn't *directly* affect the build process

Tips

- Watch for caching a “bad” values
- SET doesn't always set (cache variables)
- Use environment variables sparingly

Types?

- Everything is a string...
- ...but we can treat strings as lists
- ...or boolean values

Lists

- Implemented as semicolon delimited strings
- `\;` can escape a semicolon...
- ...but many commands don't do escaping
- Lots of list manipulation subcommands in `list(...)`

Tips

- Avoid semicolons in lists
- Be aware of all the varieties of “false” strings...
- ... e.g. “ends in ‘-NOTFOUND’”

Cache variables have type labels

- BOOL, FILEPATH, PATH, STRING, INTERNAL
- Paths pass on the commandline get normalized
- BOOL cache variables are also called options

```
set (MyCacheVar "cached value"  
    CACHE STRING "Important build setting")  
  
option (MyBuildOption "Enable picky assertions")
```

Variable expansion in commands

- Disabled in bracket arguments: `[[${Var}]]`
- Expands to a single word: `" ${Var} "`
- Expands and split into multiple words: `${Var}`

Variable expansion in `if`

- Expansion is attempted for bare words:
`if (VarName)`
- Unquoted, a variable naming another variable is expanded again
- Quoted variable references are only expanded once

Tips

- Avoid `if (${varname})`, the potential double expansion is too surprising
- Avoid directly referencing cache variables in most scenarios (`${CACHE {VarName}}`)

Running custom commands

- `execute_process` - configure time
- `add_custom_target` - build time
- `add_custom_command` - build time

Tips

- `execute_process` always runs in series, keep it fast!
- Prefer `add_custom_command` if you are able to specify dependencies and output

Building a module

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering

Alternatives

- Create `MyCustom.cmake` for `include(...)`
- `FindMyCustom.cmake` for “module” mode
- `MyCustomConfig.cmake` for “config” mode

Scopes

New scopes are created by:

- subdirectories
- functions

Scopes

New scopes are **not** created by:

- modules or included CMake files
- macros

- Regular variables get their initial value from their parent or calling scope.
- UNSET may reveal a cache variable
- Changes to non-cache variables are lost on return to parent (calling) scope

There is no file or module scope

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering

Writing a function

```
function(function_name named_arg)  
  
    # ...  
  
endfunction()
```

Tips

- Limit the number of required, positional parameters
- Use `cmake_parse_arguments`
- Use lower case for function names
- Unconditionally set local variables

Demo Time

A config mode module

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved

Bloomberg

Engineering

Questions?

Bearing in mind, I am not an expert