



Things I Learned While Trying To Avoid Becoming a CMake Expert



Avoid Becoming a CMake Expert



CMake – what and why?

Bloomberg

Why might we want to extend?

A few examples:

- "Compiling" application data
- Code generation
- Simplifying project-specific boilerplate

Bloomberg

Getting Started

TechAtBloomberg.com

Bloomberg

Phases of build

- Configuration
- Build

Bloomberg

Configuration

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

TechAtBloomberg.com



Build

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

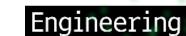
TechAtBloomberg.com

Bloomberg

A minimal CMakeLists.txt

```
cmake_minimum_required(VERSION 3.19)
project(MyProject LANGUAGES CXX)
add_executable(hello main.cpp)
```

TechAtBloomberg.com



Demo Time

TechAtBloomberg.com

Bloomberg

That didn't seem so bad?

TechAtBloomberg.com

Bloomberg

Structuring CMake

- "targets" same terminology as make
- ... add_library, add_executable
- directories
- ... add_subdirectory

Bloomberg

Tips

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

TechAtBloomberg.com

Bloomberg

Tips

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

TechAtBloomberg.com

Bloomberg

Linking dependencies

- target_link_libraries is the most common
- but add_dependencies also exists

Bloomberg

Demo Time linking

TechAtBloomberg.com

Bloomberg

Logic and State

Bloomberg

Variables

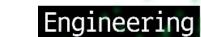
- Regular variables
- Cache variables
- Environment variables

TechAtBloomberg.com

Bloomberg

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





Regular variables

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

TechAtBloomberg.com

Bloomberg

Cache

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

TechAtBloomberg.com

Bloomberg

Environment

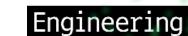
- Inherited from the calling environment
- Not preserved between runs...
- ...but might affect settings in cache variables
- Doesn't directly affect the build process

Tech At Bloomberg.com Bloomberg

Tips

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

TechAtBloomberg.com



Types?

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

TechAtBloomberg.com

Bloomberg

Lists

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

TechAtBloomberg.com



Tips

- Avoid semicolons in lists
- Be aware of all the varities of "false" strings...
- ... e.g. "ends in '-NOTFOUND"

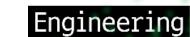
Bloomberg

Cache variables have type labels

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

TechAtBloomberg.com





Variable expansion in commands

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

TechAtBloomberg.com



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

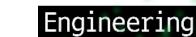
TechAtBloomberg.com

Engineering

Tips

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

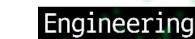
TechAtBloomberg.com



Running custom commands

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

TechAtBloomberg.com



Tips

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

TechAtBloomberg.com



Building a module

TechAtBloomberg.com

Bloomberg

Alternatives

- Create MyCustom.cmake for include (...)
- FindMyCustom.cmake for "module" mode
- MyCustomConfig.cmake for "config" mode

TechAtBloomberg.com



Scopes

New scopes are created by:

- subdirectories
- functions

Bloomberg

Scopes

New scopes are not created by:

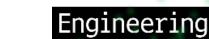
- modules or included CMake files
- macros

TechAtBloomberg.com

Bloomberg

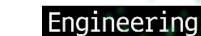
- Regular variables get their initial value from their parent or calling scope.
- UNSET may reveal a cache variable
- Changes to non-cache variables are lost on return to parent (calling) scope

TechAtBloomberg.com



There is no file or module scope

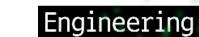
TechAtBloomberg.com



Writing a function

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

TechAtBloomberg.com



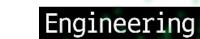
Tips

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

Bloomberg

Demo Time A config mode module

TechAtBloomberg.com



Questions?

Bearing in mind, I am not an expert

TechAtBloomberg.com

