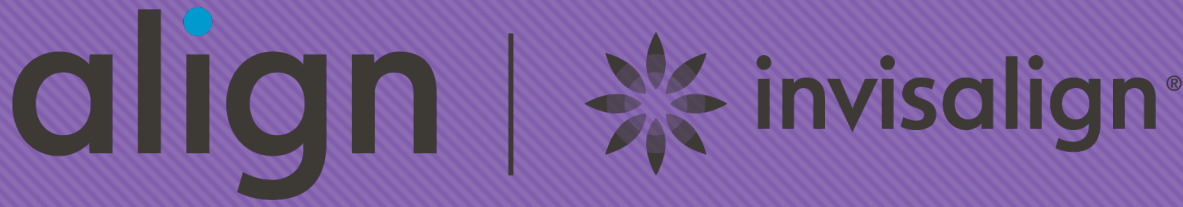


CMake 2.x давно пора закапывать

Alexander Voronkov, Expert Software Developer, Align Technology Inc.

Modern CMake

- Target-based подход
  - Логические зависимости вместо флагов линковки и путей к хедерам
  - Публичные и приватные зависимости
- Модульный дизайн сборки проекта
  - Таргет может переместится в отдельный модуль и стать импортируемым
- Широкий набор генераторных выражений
- Расширенный набор поддерживаемых систем сборки и компиляторов
- Bracket литералы

Кто то использует CMake 2.x?

```
cmake_minimum_required(VERSION 2.8)

add_compile_options(“${CMAKE_CXX_FLAGS} -std=c++14”)
```

Почему ориентируются на CMake 2.x

- Чтобы собиралось на большем количестве платформ
- Когда то написали и до сих пор нормально работает
- Если переключить на версию 3.x то CMake начинает ругаться

Policy и почему всё и так работает

```
cmake_minimum_required(VERSION 2.8)
cmake_policy(VERSION 2.8)
```

Policy и почему всё и так работает

- Каждая новая версия CMake приносит изменения поведения кода
- Каждая новая версия добавляет новые policy – флаги включающие старое или новое поведение
- Набор значений policy задает поведение CMake
- При выставлении policy по конкретной версии, CMake работает как будто он действительно этой версии

Насколько Modern CMake новый

- CMake 3.0.0 June 2014
- GCC 4.8.1 May 2013 (C++ 11 Core)
- GCC 5.1 April 2015 (C++ 11 Library)

Почему ориентируются на CMake 2.x

- Чтобы собиралось на большем количестве платформ
- Когда то написали и до сих пор нормально работает
- Если переключить на версию 3.x то CMake начинает ругаться

Где нам взять свежий CMake

- https://cmake.org/download/#latest
  - Windows x32/x64
  - macOS universal
  - Linux x86_64/aarch64
- Snap https://snapcraft.io/cmake
- PyPI https://pypi.org/project/cmake/

Еще варианты

- Ubuntu - https://apt.kitware.com/
- RHEL7 – EPEL пакет cmake3
- macOS – Homebrew
- Conan
- vcpkg

Код CMake, который устарел

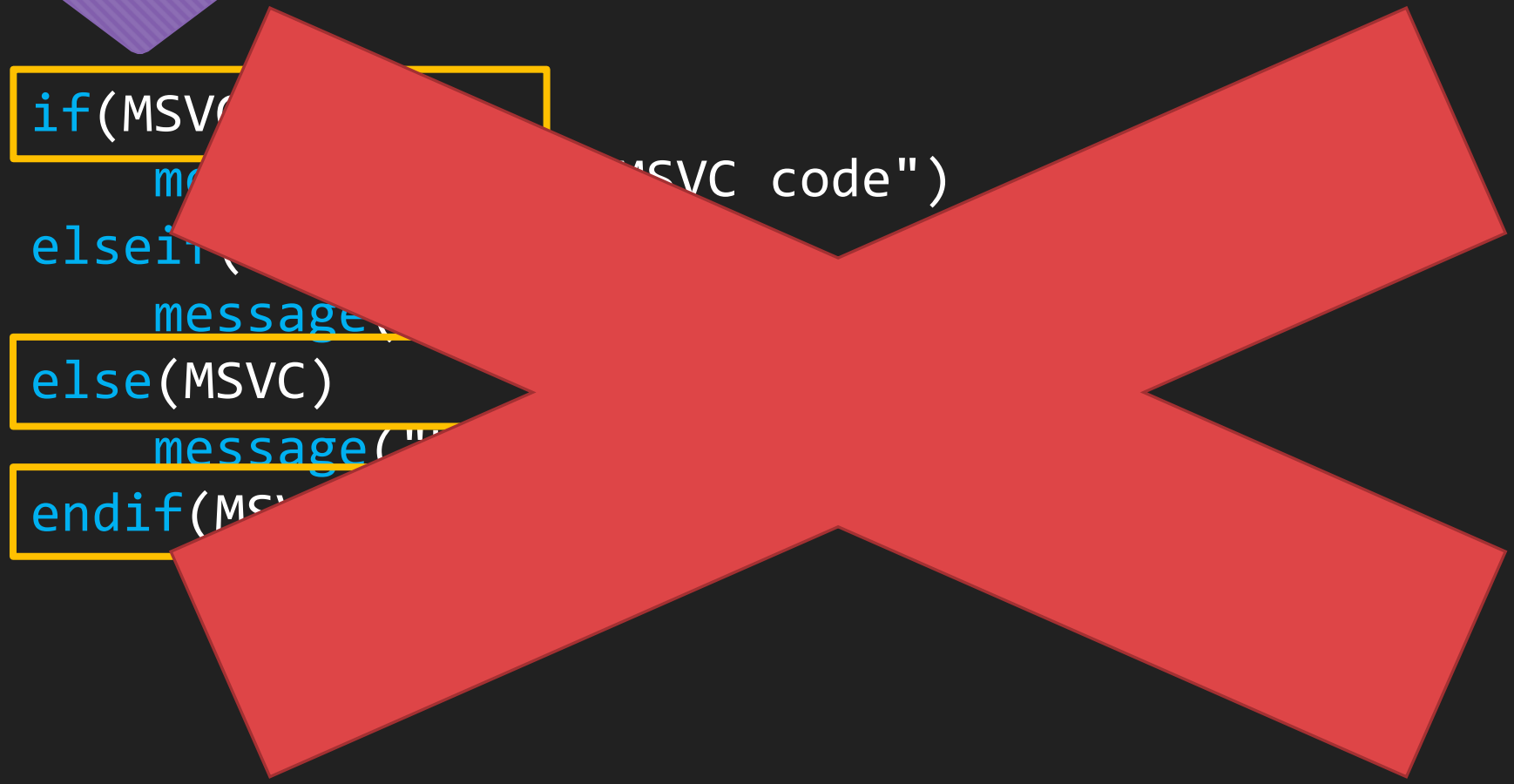
Регистр символов

- Имена команд CMake case-insensitive
- Когда-то писали капсом как в Fortran-е
  TARGET_INCLUDE_DIRECTORIES(MyExe PUBLIC impl ${MY_ADDITIONAL_DIRS})
- Но! Современный стиль
  target_include_directories(MyExe PUBLIC impl ${my_additional_dirs})

Регистр символов

- Lower-cased
  - Команды
  - Пользовательские переменные и свойства
  - Имена функций, макросов и их параметров
- Upper-cased
  - Встроенные переменные и свойства CMake
  - Генераторные выражения
  - Ключевые слова команд
  - Cache-entries

Скобки в else, endif, endforeach



Скобки в else, endif, endforeach

```
if(MSVC)
    message("Using MSVC code")
elseif(APPLE)
    message("Using Apple code")
else()
    message("Using default code")
endif()
```

Переменные в условии if

```
if("${CMAKE_CX                                    LAGS_INIT}")
```

Переменные в условии if

```
if(CMAKE_CXX_FLAGS STREQUAL CMAKE_CXX_FLAGS_INIT)
```

CMAKE_CXX_FLAGS

- Действуют на все таргеты в текущей директории (CMakeLists.txt) и нижестоящих
- Всё, что можно задать через свойства CMake, нужно делать через них
- Если нужно выставить флаги компилятора – используйте toolchain файлы
- Если проект собирается под один-два компилятора можно использовать на самом верху, обязательно с проверкой компилятора

MSVC Static Runtime

```
foreach(flag C    XE_CXX_FLAGS CMAKE_CXX_FLAGS_DEBU
                  X_FLAGS_RELEASE CMAKE_CX          MINSIZEREL
                        S_RELWITHDEBI
    if(${flag
        string(RE                    "${${flag}}")
    endif()
    if(${flag} MATC
        string(                      "${${flag}}")
    endif()
endforeach()
```

MSVC Runtime CMake 3.15+

```
set(CMAKE_MSVC_RUNTIME_LIBRARY "MultiThreaded$<$<CONFIG:Debug>:Debug>")


set_property(TARGET MyTarget
    PROPERTY MSVC_RUNTIME_LIBRARY "MultiThreaded$<$<CONFIG:Debug>:Debug>")
```

Команды глобальных флагов

- add_compile_opt
- include_directories
- link_directories
- link_libraries

Версия проекта, CMake 2.x

```
cmake_minimum_rec                    2.8.12)
project(MyProjec
set(MyProject_V
set(MyProject_VERSI
set(MyProject_VERSION_PA
set(MyProject_VERSION
   ${MyProject_VERSION_MAJ                    {MyProject_VERSION_PATCH} )
```

Версия проекта, Modern CMake

```
cmake_minimum_required(VERSION 3.0)
project(MyProject VERSION 2.4.7)
```

Начиная с CMake 3.12, также задаются переменные для вызова команды project из самого верхнеуровневого CMakeLists.txt

```
CMAKE_PROJECT_VERSION
CMAKE_PROJECT_VERSION_MAJOR
CMAKE_PROJECT_VERSION_MINOR
CMAKE_PROJECT_VERSION_PATCH
CMAKE_PROJECT_VERSION_TWEAK
```

Списки исходников

```
set(headers
set(sources a.cp
add_executable(MyExe                    rs})
```

Списки исходников

```
add_executable(MyExe
               a.h b.h c.h
               a.cpp b.cpp c.cpp)
```

make check ?

```
enable_testing()
set(CMAKE_CTEST_COM
add_custom_target(chec                    EST_COMMAND})
add_executable(te                         n)
add_test(test
add_dependenc                             t-1)
```

CMake way

```
include(CTest)
if(BUILD_TESTING)
    add_executable(test-1 test1.cpp)
    add_test(NAME test-1 COMMAND test-1)
endif()
```

find_package

```
find_package(ZL

target_include_director                    ${ZLIB_INCLUDE_DIRS})
target_link_librari                        IBRARIES})
```

Используйте таргеты, если они есть

```
find_package(ZLIB REQUIRED)

target_link_libraries(myTarget PRIVATE ZLIB::ZLIB)
```

Macro vs Function

Macro

- Область видимости вызывающего кода
- Параметры - строковые подстановки
- ARGN, ARGC, ARGV, ARGV0, ARGV1и т.д. также строковые подстановки
- Для возврата значения нужно выставить переменную CMake

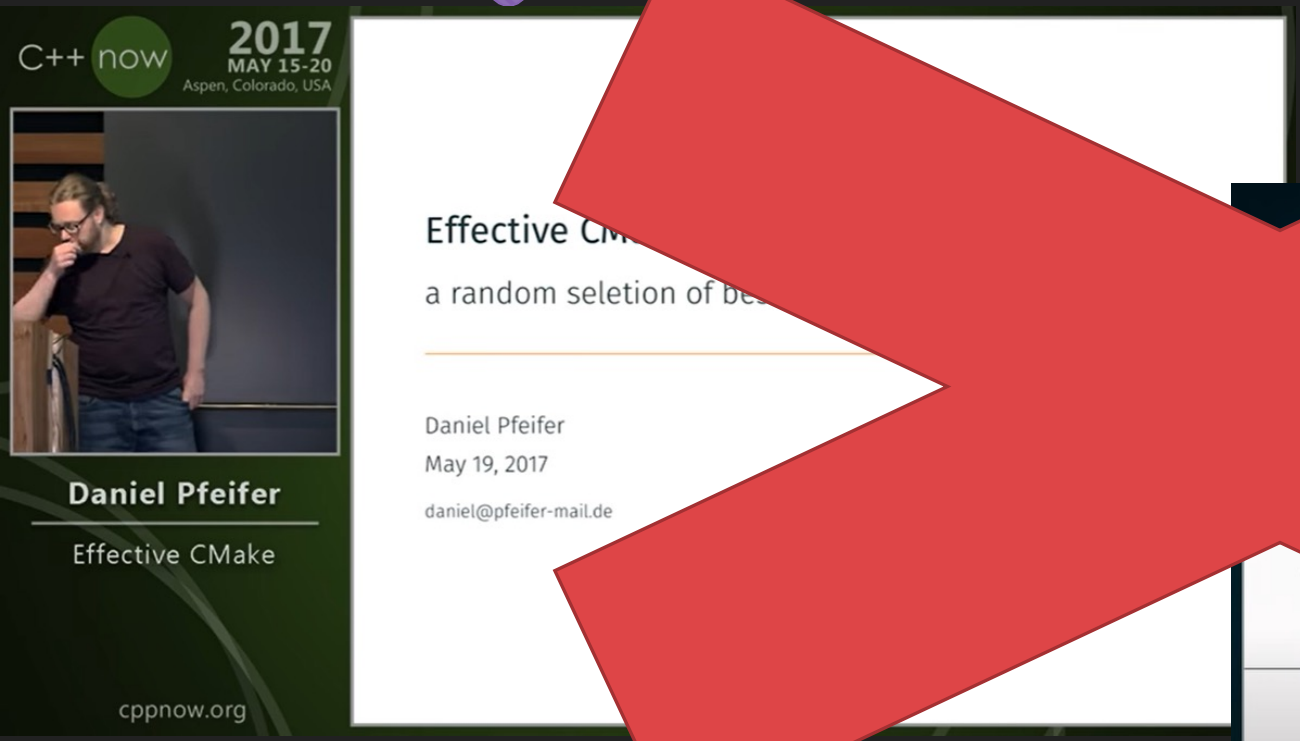
Function

- Отдельная область видимости
- Параметры являются переменными
- ARGN, ARGC, ARGV, ARGV0, ARGV1 и т.д. полноценные переменные
- Для возврата значения, нужно выставить переменную CMake с параметром PARENT_SCOPE

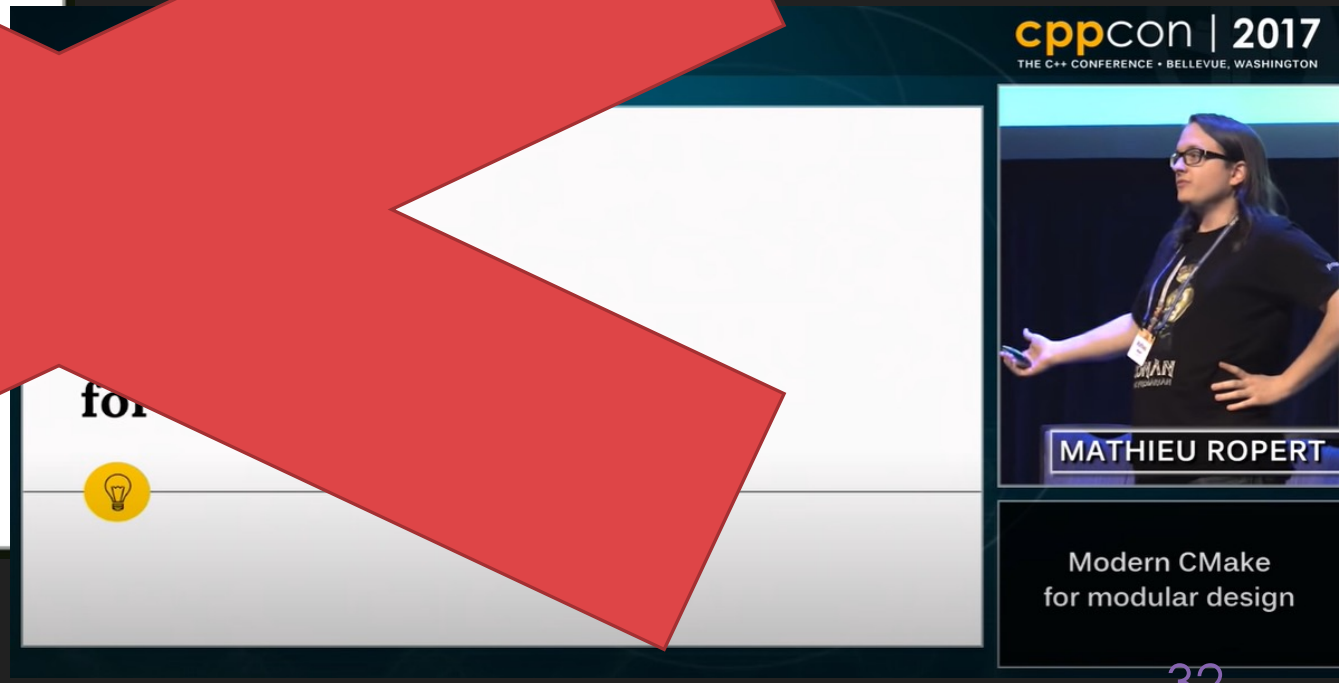
Macro или Function?

- Старайтесь всегда использовать function
- Контролируйте переменные используемые для возврата значений
- Можно передавать имя переменной для возврата значения через аргументы
- Используйте macro, если бы вы использовали препроцессор C для данной задачи
- macro не должен оставлять после себя лишних заданных переменных (unset)

Источники информации



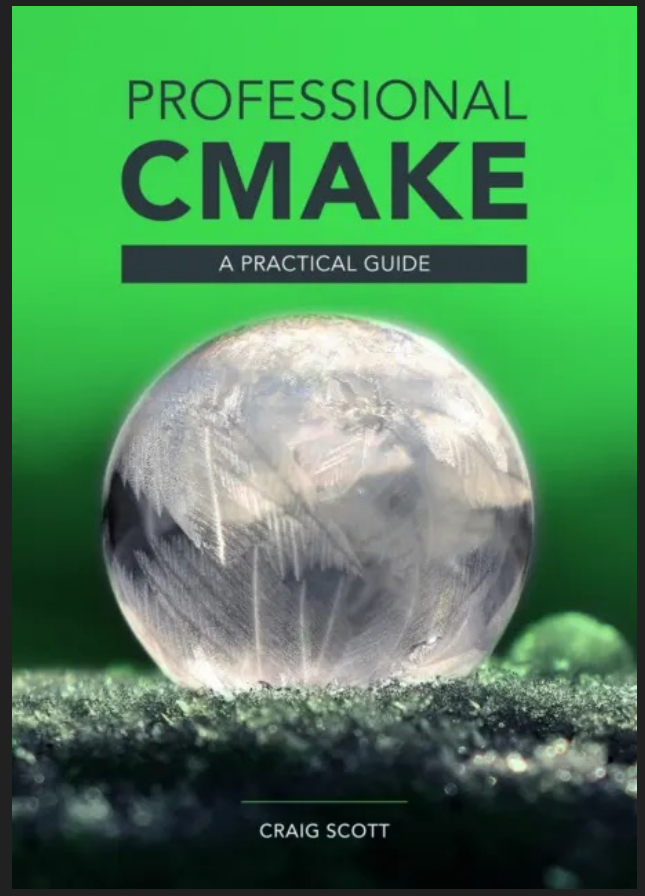
https://youtu.be/b3xLMQ6Wglk

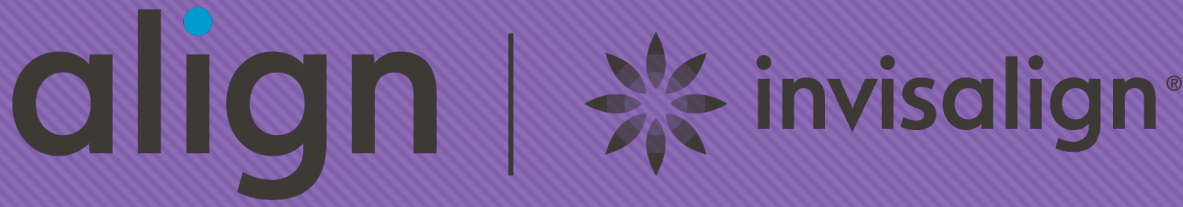


https://youtu.be/eC9-iRN2b04

Источники информации

https://crascit.com/professional-cmake/





Спасибо!

Alexander Voronkov, Expert Software Developer, Align Technology Inc.