

## chapter3.1 cmake代码生成配置文件

### 1. 代码生成介绍：

代码生成是一个很好用的功能，它可以使用一份公共的描述文件，生成不同语言下的源代码。这个功能使得需要人工编写的代码大幅减少，同时也增加了互操作性。

下面的例子展示了如何使用CMake变量和其他常用的工具进行代码生成。

- [configure-file](#)
  - Using the CMake `configure_file` function to inject CMake variables.
  - 使用CMake中的`configure_file`函数注入CMake变量
- [Protocol Buffers](#)
  - Using Google Protocol Buffers to generate C++ source.
  - 使用Google Protocol Buffers来生成C++源码

### 2. 文件结构：

- [CMakeLists.txt](#) - 描述你希望能够运行的CMake命令
  - [CMakeLists.txt](#) - Contains the CMake commands you wish to run
- [main.cpp](#) - 包含主函数的源文件
  - [main.cpp](#) - The source file with main
- [path.h.in](#) - 描述待构建目录的文件
  - [path.h.in](#) - File to contain a path to the build directory
- [ver.h.in](#) - 描述工程版本信息的文件
  - [ver.h.in](#) - File to contain the version of the project

```
unix
1  | .
2  |   CMakeLists.txt
3  |   README.adoc
4  |   build
5  |   main.cpp
6  |   path.h.in
7  |   ver.h.in
```

### 3. 文件解析：

在CMake中，你可以在文件中使用 `configure_file()` 函数进行变量的替换，这一函数的必要参数是源文件和目标文件

```
configure_file(ver.h.in ${PROJECT_BINARY_DIR}/ver.h)

configure_file(path.h.in ${PROJECT_BINARY_DIR}/path.h @ONLY)
```

第一个例子，在`ver.h.in`文件中，CMake可以将使用 `${}` 或 `@@` 的语法来定义一个CMake变量。  
在执行代码生成之后，在 `PROJECT_BINARY_DIR` 目录下 (此处就是build文件夹) 将会出现一个新的`ver.h`文件。

```
const char* ver = "${cf_example_VERSION}";
```

第二个例子，在`path.h.in`文件中，`@ONLY` 指定了它只能用 `@@` 的语法来定义一个CMake变量。同样地，在执行代码生成之后，在 `PROJECT_BINARY_DIR` 目录下将会出现一个新的`path.h`文件。

```
const char* path = "@CMAKE_SOURCE_DIR@";
```

过程详解:

(1) 文件属性说明:

```
|— main.cpp  -----提供主程序, 但是其中ver和path两个变量未定义
|— path.h.in -----提供: path的具象化
|— ver.h.in  -----提供: ver的具象化
```

(2) CMakeList控制文本说明:

```
cmake_minimum_required(VERSION 3.5)

# (1) 设置项目名, called: cf_example
project (cf_example)

# (2) set a project version 版本设置 (我们可以忽略)
set (cf_example_VERSION_MAJOR 0)
set (cf_example_VERSION_MINOR 2)
set (cf_example_VERSION_PATCH 1)
set (cf_example_VERSION
"${cf_example_VERSION_MAJOR}.${cf_example_VERSION_MINOR}.${cf_example_VERSION_PATCH}")

# (3.1) 确定指代文本: ver变量 从 ver.h.in 里面获取
# Call configure files on ver.h.in to set the version.
# Uses the standard ${VARIABLE} syntax in the file
configure_file(ver.h.in ${PROJECT_BINARY_DIR}/ver.h)

# (3.2) 确定指代文本: path变量 从 path.h.in 里面获取
# configure the path.h.in file.
# This file can only use the @VARIABLE@ syntax in the file
configure_file(path.h.in ${PROJECT_BINARY_DIR}/path.h @ONLY)

# (4) 利用main.cpp制作可执行文件, called: cf_example
add_executable(cf_example
    main.cpp
)

# include the directory with the new files
# (5) 引入库链接
target_include_directories( cf_example
    PUBLIC
    ${CMAKE_BINARY_DIR}
)
```

## 4. 总览:

```
huluobo@huluobodeMacBook-Pro ► ~/cmake-examples/myCmake/chapter3.1/build ► ʘ main ± ► cmake ..
-- The C compiler identification is AppleClang 15.0.0.15000040
-- The CXX compiler identification is AppleClang 15.0.0.15000040
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /Library/Developer/CommandLineTools/usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /Library/Developer/CommandLineTools/usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done (0.3s)
```

```

-- Generating done (0.0s)
-- Build files have been written to: /Users/huluobo/cmake-examples/myCmake/chapter3.1/build
huluobo@huluobodeMacBook-Pro ➤ ~/cmake-examples/myCmake/chapter3.1/build ➤ ⌘ main ± ➤ ls
CMakeCache.txt      CMakeFiles          Makefile            cmake_install.cmake path.h              ver.h
huluobo@huluobodeMacBook-Pro ➤ ~/cmake-examples/myCmake/chapter3.1/build ➤ ⌘ main ± ➤ cat path.h
#ifndef __PATH_H__
#define __PATH_H__

// version variable that will be substituted by cmake
// This shows an example using the @ variable type
const char* path = "/Users/huluobo/cmake-examples/myCmake/chapter3.1";

#endif

huluobo@huluobodeMacBook-Pro ➤ ~/cmake-examples/myCmake/chapter3.1/build ➤ ⌘ main ± ➤ cat ver.h
#ifndef __VER_H__
#define __VER_H__

// version variable that will be substituted by cmake
// This shows an example using the $ variable type
const char* ver = "0.2.1";

#endif

huluobo@huluobodeMacBook-Pro ➤ ~/cmake-examples/myCmake/chapter3.1/build ➤ ⌘ main ± ➤ make
[ 50%] Building CXX object CMakeFiles/cf_example.dir/main.cpp.o
[100%] Linking CXX executable cf_example
[100%] Built target cf_example
huluobo@huluobodeMacBook-Pro ➤ ~/cmake-examples/myCmake/chapter3.1/build ➤ ⌘ main ± ➤ ./cf_example
Hello Version 0.2.1!
Path is /Users/huluobo/cmake-examples/myCmake/chapter3.1
huluobo@huluobodeMacBook-Pro ➤ ~/cmake-examples/myCmake/chapter3.1/build ➤ ⌘ main ± ➤ cd ..
huluobo@huluobodeMacBook-Pro ➤ ~/cmake-examples/myCmake/chapter3.1 ➤ ⌘ main ± ➤ tree
.
├── CMakeLists.txt
├── build
│   ├── CMakeCache.txt
│   ├── CMakeFiles
│   │   ├── 3.28.0-rc5
│   │   │   ├── CMakeCCompiler.cmake
│   │   │   ├── CMakeCXXCompiler.cmake
│   │   │   ├── CMakeDetermineCompilerABI_C.bin
│   │   │   ├── CMakeDetermineCompilerABI_CXX.bin
│   │   │   ├── CMakeSystem.cmake
│   │   │   ├── CompilerIdC
│   │   │   │   ├── CMakeCCompilerId.c
│   │   │   │   ├── CMakeCCompilerId.o
│   │   │   │   └── tmp
│   │   │   └── CompilerIdCXX
│   │   │       ├── CMakeCXXCompilerId.cpp
│   │   │       ├── CMakeCXXCompilerId.o
│   │   │       └── tmp
│   │   ├── CMakeConfigureLog.yaml
│   │   ├── CMakeDirectoryInformation.cmake
│   │   ├── CMakeScratch
│   │   ├── Makefile.cmake
│   │   ├── Makefile2
│   │   ├── TargetDirectories.txt
│   │   ├── cf_example.dir
│   │   │   ├── DependInfo.cmake
│   │   │   ├── build.make
│   │   │   ├── cmake_clean.cmake
│   │   │   ├── compiler_depend.make
│   │   │   ├── compiler_depend.ts
│   │   │   ├── depend.make
│   │   │   ├── flags.make
│   │   │   └── link.txt

```

```
| | | | └─ main.cpp.o
| | | | └─ main.cpp.o.d
| | | | └─ progress.make
| | | └─ cmake.check_cache
| | └─ pkgRedirects
| | └─ progress.marks
| └─ Makefile
| └─ cf_example
| └─ cmake_install.cmake
| └─ path.h
| └─ ver.h
└─ main.cpp
└─ path.h.in
└─ ver.h.in
```

11 directories, 37 files

huluobo@huluobodeMacBook-Pro ► ~/cmake-examples/myCmake/chapter3.1 ► ↵ main ± ►