

# Cmake

## 笔记

H-X-B L<sup>A</sup>T<sub>E</sub>X

哈尔滨工业大学 航天学院

*292554331@qq.com*

August 29, 2016

```
cmake_minimum_required(VERSION 2.8.3 FATAL_ERROR)
project(Cmake)
set(SRC_LIST main.cpp)
message(STATUS "" $PROJECT_BINARY_DIR)
message(STATUS "" $PROJECT_SOURCE_DIR)
add_executable(cmake1 $SRC_LIST)
make VERBOSE=1 看到 make 构建的详细过程
```

```
SET(VAR [VALUE] [CACHE TYPE DOCSTRING [FORCE]])  
MESSAGE([SEND_ERROR — STATUS — FATAL_ERROR]  
"message to display" ...)
```

SEND\_ERROR 产生错误,生成过程被跳过。

STATUS 输出前缀为—的信息。

FATAL\_ERROR,立即终止所有 cmake 过程。

```
ADD_SUBDIRECTORY(source_dir [binary_dir]
[EXCLUDE_FROM_ALL])
```

这个指令用于向当前工程添加存放源文件的子目录,并可以指定中间二进制和目标二进制存放的位置。EXCLUDE\_FROM\_ALL参数的含义是将这个目录从编译过程中排除,

```
ADD_SUBDIRECTORY(src bin)
```

上面的例子定义了将 src 子目录加入工程,并指定编译输出(包含编译中间结果)路径为 bin 目录。如果不进行 bin 目录的指定,那么编译结果(包括中间结果)都将存放在 build/src 目录(这个目录跟原有的 src 目录对应),指定 bin 目录后,相当于在编译时将 src 重命名为 bin,所有的中间结果和目标二进制都将存放在 bin 目录。

```
SET(EXECUTABLE_OUTPUT_PATH  
$PROJECT_BINARY_DIR/bin)
```

```
SET(LIBRARY_OUTPUT_PATH $PROJECT_BINARY_DIR/lib)
```

在哪里 ADD\_EXECUTABLE 或 ADD\_LIBRARY,如果需要改变目标存放路径,就在哪里加入上述的定义

目标类型也就相对应的有三种,ARCHIVE 特指静态库,LIBRARY 特指动态库,RUNTIME 特指可执行目标二进制。

DESTINATION 定义了安装的路径,如果路径以/开头,那么指的是绝对路径,这时候 CMAKE\_INSTALL\_PREFIX 其实就无效了。如果你希望使用 CMAKE\_INSTALL\_PREFIX 来定义安装路径,就要写成相对路径,即不要以/开头,那么安装后的路径就是 \$CMAKE\_INSTALL\_PREFIX/;DESTINATION 定义的路径;

```
INSTALL(TARGETS myrun mylib mystaticlib  
  RUNTIME DESTINATION bin  
  LIBRARY DESTINATION lib  
  ARCHIVE DESTINATION libstatic  
)
```

可执行二进制myrun安装到\$CMAKE\_INSTALL\_PREFIX/bin 目录  
动态库libmylib安装到\$CMAKE\_INSTALL\_PREFIX/lib 目录  
静态库libmystaticlib安装到\$CMAKE\_INSTALL\_PREFIX/libstatic  
目录







# Example

```
INSTALL(DIRECTORY icons scripts/ DESTINATION
share/myproj PATTERN "CVS" EXCLUDE
PATTERN "scripts/*"
PERMISSIONS OWNER_EXECUTE OWNER_WRITE
OWNER_READ
GROUP_EXECUTE GROUP_READ)
```

将 icons 目录安装到 `jprefixi/share/myproj`, 将 scripts/ 中的内容安装到 `jprefixi/share/myproj` 不包含目录名为 CVS 的目录, 对于 scripts/\* 文件指定权限为 OWNER\_EXECUTE OWNER\_WRITE OWNER\_READ GROUP\_EXECUTE GROUP\_READ.

安装 COPYRIGHT/README: INSTALL(FILES COPYRIGHT  
README DESTINATION share/doc/cmake/t2)

安装 runhello.sh(脚本):INSTALL(PROGRAMS runhello.sh  
DESTINATION bin)

安装 doc 中的 hello.txt: INSTALL(DIRECTORY doc/  
DESTINATION share/doc/cmake/t2)

```
cmake -DCMAKE_INSTALL_PREFIX=/tmp/t2/usr ..  
make  
make install
```

```
SET(LIBHELLO_SRC hello.c)
ADD_LIBRARY(hello SHARED $LIBHELLO_SRC)
SET(LIBRARY_OUTPUT_PATH 路径i)来指定一个新的位置
```

SHARED, 动态库

STATIC, 静态库

`ADD_LIBRARY(hello_static STATIC $LIBHELLO_SRC)` 就可以构建一个 `libhello_static.a` 的静态库了。

`SET_TARGET_PROPERTIES(hello_static PROPERTIES OUTPUT_NAME "hello")` 将 `libhello_static.a` 改名为 `libhello.a`





SET\_TARGET\_PROPERTIES,其基本语法是:

```
SET_TARGET_PROPERTIES(target1 target2 ...  
PROPERTY prop1 value1  
prop2 value2 ...)
```

```
GET_TARGET_PROPERTY(VAR target property)
```

```
Example:GET_TARGET_PROPERTY(OUTPUT_VALUE  
hello_static OUTPUT_NAME)
```

```
MESSAGE(STATUS " This is the hello_static  
OUTPUT_NAME:" $OUTPUT_VALUE)
```

```
SET_TARGET_PROPERTIES(hello PROPERTIES  
  CLEAN_DIRECT_OUTPUT 1)  
SET_TARGET_PROPERTIES(hello_static PROPERTIES  
  CLEAN_DIRECT_OUTPUT 1)  
没有用 有可能版本的问题
```

```
SET_TARGET_PROPERTIES(hello PROPERTIES VERSION 1.2  
SOVERSION 1)
```

VERSION 指代动态库版本,SOVERSION 指代 API 版本。

```
INSTALL(TARGETS hello hello_static  
LIBRARY DESTINATION lib  
ARCHIVE DESTINATION lib)  
INSTALL(FILES hello.h DESTINATION include/hello)
```

# 使用共享库和头文件

方法一: `include|hello/hello.h`

方法二: `INCLUDE_DIRECTORIES(/usr/include/hello)`

方法三: `export CMAKE_INCLUDE_PATH=/usr/include/hello`

`FIND_PATH(myHeader hello.h)`

`IF(myHeader)`

`INCLUDE_DIRECTORIES($myHeader)`

`ENDIF(myHeader)`

`TARGET_LINK_LIBRARIES(main hello)`

`LINK_DIRECTORIES(directory1 directory2 ...)`

添加非标准的共享库搜索路径,比如,在工程内部同时存在共享库和可执行二进制,在编译时就需要指定一下这些共享库的路径。

`TARGET_LINK_LIBRARIES(target library1`

`debug — optimized; library2`

`...)`

使用\$ENVNAME指令就可以调用系统的环境变量

测试:add\_test(mytest \$PROJECT\_BINARY\_DIR/src/cmake)

enable\_testing()

make test

```
if(WIN32)
message(STATUS " This is windows")
elseif(LINUX)
message(STATUS " This is Linux")
elseif(UNIX)
message(STATUS " This is UNIX")
endif(WIN32)
```



```
FOREACH(VAR RANGE 10)  
MESSAGE($VAR)  
输出:0 1 2 3 4 5 6 7 8 9 10
```

```
FOREACH(A RANGE 5 15 3)
MESSAGE($A)
ENDFOREACH(A)
输出:5 8 11 14
```