[**CMake 复制文件方法**](https://www.cnblogs.com/JoyPoint/p/11629521.html)

    我们经常会遇到将第三方库文件复制到项目运行时文件夹，或者将子项目生成的库文件复制到项目运行时文件夹的情况，本文介绍FILE-COPY、add\_custom\_command、ADD\_CUSTOM\_TARGET三种方法及CMake COMMAND提供的命令说明。

**一、    FILE-COPY**

[复制代码](javascript:void(0);)

file(<COPY|INSTALL> <files>... DESTINATION <dir>

[FILE\_PERMISSIONS <permissions>...]

[DIRECTORY\_PERMISSIONS <permissions>...]

[NO\_SOURCE\_PERMISSIONS] [USE\_SOURCE\_PERMISSIONS]

[FOLLOW\_SYMLINK\_CHAIN]

[FILES\_MATCHING]

[[PATTERN <pattern> | REGEX <regex>]

[EXCLUDE] [PERMISSIONS <permissions>...]] [...])

[复制代码](javascript:void(0);)

COPY将文件，目录和符号链接复制到目标文件夹。相对于当前源目录评估相对输入路径，相对于当前构建目录评估相对目的地。复制会保留输入文件的时间戳，并优化文件（如果该文件存在于具有相同时间戳的目标文件中）。复制将保留输入权限，除非给出明确的权限或NO\_SOURCE\_PERMISSIONS（默认为USE\_SOURCE\_PERMISSIONS）。

如果指定了FOLLOW\_SYMLINK\_CHAIN，则COPY将在给定的路径上递归解析符号链接，直到找到真实文件为止，然后在目标位置为遇到的每个符号链接安装相应的符号链接。对于已安装的每个符号链接，解析都会从目录中剥离，仅保留文件名，这意味着新符号链接指向与符号链接相同目录中的文件。此功能在某些Unix系统上很有用，在这些系统中，库是作为带有符号链接的版本号安装的，而较少特定的版本指向的是特定版本。FOLLOW\_SYMLINK\_CHAIN会将所有这些符号链接和库本身安装到目标目录中。例如，如果您具有以下目录结构：

* /opt/foo/lib/libfoo.so.1.2.3
* /opt/foo/lib/libfoo.so.1.2 -> libfoo.so.1.2.3
* /opt/foo/lib/libfoo.so.1 -> libfoo.so.1.2
* /opt/foo/lib/libfoo.so -> libfoo.so.1

你可以：

file(COPY /opt/foo/lib/libfoo.so DESTINATION lib FOLLOW\_SYMLINK\_CHAIN)

这会将所有符号链接和libfoo.so.1.2.3本身安装到lib中。

请参阅install（DIRECTORY）命令以获取权限，FILES\_MATCHING，PATTERN，REGEX和EXCLUDE选项的文档。即使使用选项来选择文件的子集，复制目录也会保留其内容的结构。

INSTALL与COPY略有不同：它打印状态消息（取决于CMAKE\_INSTALL\_MESSAGE变量），并且默认为NO\_SOURCE\_PERMISSIONS。 install（）命令生成的安装脚本使用此签名（以及一些未记录的内部使用选项）。

**二、    ADD\_CUSTOM\_COMMAND**

add\_custom\_command：

该命令可以为生成的构建系统添加一条自定义的构建规则。这里又包含两种使用方式，一种是通过自定义命令在构建中生成输出文件，另外一种是向构建目标添加自定义命令。命令格式分别为：

(1)生成文件

[复制代码](javascript:void(0);)

add\_custom\_command(OUTPUT output1 [output2 ...]

COMMAND command1 [ARGS] [args1...]

[COMMAND command2 [ARGS] [args2...] ...]

[MAIN\_DEPENDENCY depend]

[DEPENDS [depends...]]

[BYPRODUCTS [files...]]

[IMPLICIT\_DEPENDS <lang1> depend1

[<lang2> depend2] ...]

[WORKING\_DIRECTORY dir]

[COMMENT comment]

[DEPFILE depfile]

[JOB\_POOL job\_pool]

[VERBATIM] [APPEND] [USES\_TERMINAL]

[COMMAND\_EXPAND\_LISTS])

[复制代码](javascript:void(0);)

参数介绍：

OUTPUT：

指定命令预期产生的输出文件。如果输出文件的名称是相对路径，即相对于当前的构建的源目录路径。输出文件可以指定多个output1,output2(可选)等。

COMMAND：

指定要在构建时执行的命令行。如果指定多个COMMAND，它们讲按顺心执行。ARGS参数是为了向后兼容，为可选参数。args1和args2为参数，多个参数用空格隔开。

MAIN\_DEPENDENCY：

可选命令，指定命令的主要输入源文件。

DEPENDS：

指定命令所依赖的文件。

BYPRODUCTS：

可选命令，指定命令预期产生的文件，但其修改时间可能会比依赖性更新，也可能不会更新。

IMPLICIT\_DEPENDS：

可选命令，请求扫描输入文件的隐式依赖关系。给定的语言指定应使用相应的依赖性扫描器的编程语言。目前只支持C和CXX语言扫描器。必须为IMPLICIT\_DEPENDS列表中的每个文件指定语言。从扫描中发现的依赖关系在构建时添加到自定义命令的依赖关系。请注意，IMPLICIT\_DEPENDS选项目前仅支持Makefile生成器，并且将被其他生成器忽略。

WORKING\_DIRECTORY：

可选命令，使用给定的当前工作目录执行命令。如果它是相对路径，它将相对于对应于当前源目录的构建树目录。

COMMENT：

可选命令，在构建时执行命令之前显示给定消息。

DEPFILE：

可选命令，为Ninja生成器指定一个.d depfile。 .d文件保存通常由自定义命令本身发出的依赖关系。对其他生成器使用DEPFILE是一个错误。

COMMAND\_EXPAND\_LISTS：

将扩展COMMAND参数中的列表，包括使用生成器表达式创建的列表，从而允许COMMAND参数，例如${CC} "-I$<JOIN:$<TARGET\_PROPERTY:foo,INCLUDE\_DIRECTORIES>,;-I>" foo.cc为适当扩展。

VERBATIM：

对于构建工具，将正确转义命令的所有自变量，以便调用的命令不变地接收每个自变量。 请注意，在add\_custom\_command甚至没有看到参数之前，CMake语言处理器仍使用一种转义。 建议使用VERBATIM，因为它可以使行为正确。 如果未提供VERBATIM，则该行为是特定于平台的，因为没有针对工具的特殊字符的保护。

使用实例：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | add\_executable(MakeTable MakeTable.cxx)  add\_custom\_command (    OUTPUT ${CMAKE\_CURRENT\_BINARY\_DIR}/Table.h    COMMAND MakeTable ${CMAKE\_CURRENT\_BINARY\_DIR}/Table.h    DEPENDS MakeTable    COMMENT "This is a test"    ) |

(2)自定义构建事件

[复制代码](javascript:void(0);)

add\_custom\_command(TARGET <target>

PRE\_BUILD | PRE\_LINK | POST\_BUILD

COMMAND command1 [ARGS] [args1...]

[COMMAND command2 [ARGS] [args2...] ...]

[BYPRODUCTS [files...]]

[WORKING\_DIRECTORY dir]

[COMMENT comment]

[VERBATIM] [USES\_TERMINAL])

[复制代码](javascript:void(0);)

参数介绍：

TARGET：

定义了与构建指定相关联的新命令。当已经存在是，相应的command将不再执行。

PRE\_BUILD：

在目标中执行任何其他规则之前运行。这仅在Visual Studio 7或更高版本上受支持。对于所有其他生成器PRE\_BUILD将被视为PRE\_LINK。

PRE\_LINK：

在编译源之后运行，但在链接二进制文件或运行静态库的库管理器或存档器工具之前运行。

POST\_BUILD：

在目标中的所有其他规则都已执行后运行。

**示例：将子项目生成的库文件复制到项目运行时文件夹。**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | #=============Copy Plugins Runtime FILES to Main Project============  #一般将此内容放在lib子项目的CMakelists.txt的最后，该方法采用POST\_BUILD，所以需注意要复制的源应该是一个固定字符串，而不能用FILE GLOB的方法，因为在编译前该源为空，在VS中会出现MSB3073错误，提示copy from 为空值。  SET(Plugins\_TEST\_Debug\_DLL\_FILE      ${CMAKE\_CURRENT\_BINARY\_DIR}/Debug/lib${PROJECT\_NAME}.dll  )  SET(Plugins\_TEST\_Release\_DLL\_FILE      ${CMAKE\_CURRENT\_BINARY\_DIR}/Release/lib${PROJECT\_NAME}.dll  )    add\_custom\_command(TARGET ${PROJECT\_NAME}     POST\_BUILD     COMMAND ${CMAKE\_COMMAND} -E         copy\_if\_different          "$<$<CONFIG:Release>:${Plugins\_TEST\_Release\_DLL\_FILE}>"          "$<$<CONFIG:Debug>:${Plugins\_TEST\_Debug\_DLL\_FILE}>"          "${CMAKE\_BINARY\_DIR}/$<$<CONFIG:Release>:Release>$<$<CONFIG:Debug>:Debug>/Plugins/org\_test\_plugins/"  ) |

**三、    ADD\_CUSTOM\_TARGET**

add\_custom\_target：

该命令可以给指定名称的目标执行指定的命令，该目标没有输出文件，并始终被构建。命令的格式为：

[复制代码](javascript:void(0);)

add\_custom\_target(Name [ALL] [command1 [args1...]]

[COMMAND command2 [args2...] ...]

[DEPENDS depend depend depend ... ]

[BYPRODUCTS [files...]]

[WORKING\_DIRECTORY dir]

[COMMENT comment]

[JOB\_POOL job\_pool]

[VERBATIM] [USES\_TERMINAL]

[COMMAND\_EXPAND\_LISTS]

[SOURCES src1 [src2...]])

[复制代码](javascript:void(0);)

参数介绍：

Name：

指定目标的名称，单独成为一个子项目。

ALL：

表明此目标应添加到默认构建目标，以便每次都将运行（该命令名称不能为ALL）

SOURCES：

指定要包括在自定义目标中的其他源文件。指定的源文件将被添加到IDE项目文件中，以方便编辑，即使它们没有构建规则。

**示例：将第三方库文件复制到项目运行时文件夹**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | #=============Copy Source files to Build Runtime Dir===============  #该内容一般放在项目顶层CMakelists.txt的最后，  #目的是将项目生成后的执行文件所需的第三方库复制到执行程序目录，  #并区分Debug和Release版本。  #该方法中的COMMAND\_EXPAND\_LISTS参数值得关注，可以复制列表内所有文件。  FILE(GLOB Plugin\_Runtime\_Debug\_DLL\_FILES CONFIGURE\_DEPENDS      ${CMAKE\_CURRENT\_SOURCE\_DIR}/Plugin\_Runtime\_Dir/Debug/\*.\*  )  FILE(GLOB Plugin\_Runtime\_Release\_DLL\_FILES CONFIGURE\_DEPENDS      ${CMAKE\_CURRENT\_SOURCE\_DIR}/Plugin\_Runtime\_Dir/Release/\*.\*  )  FILE(GLOB Plugin\_Runtime\_Debug\_Resources\_FILES CONFIGURE\_DEPENDS      ${CMAKE\_CURRENT\_SOURCE\_DIR}/Plugin\_Runtime\_Dir/Debug/Resources/icos/\*.\*  )  FILE(GLOB Plugin\_Runtime\_Release\_Resources\_FILES CONFIGURE\_DEPENDS      ${CMAKE\_CURRENT\_SOURCE\_DIR}/Plugin\_Runtime\_Dir/Release/Resources/icos/\*.\*  )    add\_custom\_target(CopyRuntimeFiles ALL    VERBATIM    COMMAND\_EXPAND\_LISTS    COMMAND ${CMAKE\_COMMAND} -E            make\_directory "${PROJECT\_BINARY\_DIR}/$<$<CONFIG:Release>:Release>$<$<CONFIG:Debug>:Debug>/"  COMMAND ${CMAKE\_COMMAND} -E           copy\_if\_different                  "$<$<CONFIG:Release>:${Plugin\_Runtime\_Release\_DLL\_FILES}>"                   "$<$<CONFIG:Debug>:${Plugin\_Runtime\_Debug\_DLL\_FILES}>"                  "${PROJECT\_BINARY\_DIR}/$<$<CONFIG:Release>:Release>$<$<CONFIG:Debug>:Debug>/"  COMMAND ${CMAKE\_COMMAND} -E          make\_directory "${PROJECT\_BINARY\_DIR}/$<$<CONFIG:Release>:Release>$<$<CONFIG:Debug>:Debug>/Resources/icos/"  COMMAND ${CMAKE\_COMMAND} -E          copy\_if\_different                 "$<$<CONFIG:Release>:${Plugin\_Runtime\_Release\_Resources\_FILES}>"                 "$<$<CONFIG:Debug>:${Plugin\_Runtime\_Debug\_Resources\_FILES}>"                 "${PROJECT\_BINARY\_DIR}/$<$<CONFIG:Release>:Release>$<$<CONFIG:Debug>:Debug>/Resources/icos/"  COMMAND ${CMAKE\_COMMAND} -E          make\_directory "${PROJECT\_BINARY\_DIR}/$<$<CONFIG:Release>:Release>$<$<CONFIG:Debug>:Debug>/Plugins/org\_test\_plugins/"  ) |

**四、    COMMAND**

[复制代码](javascript:void(0);)

cmake version 3.15.2

Usage: <command> [arguments...]

Available commands:

capabilities - Report capabilities built into cmake in JSON format

chdir dir cmd [args...] - run command in a given directory

compare\_files [--ignore-eol] file1 file2 - check if file1 is same as file2

copy <file>... destination - copy files to destination (either file or directory)

copy\_directory <dir>... destination - copy content of <dir>... directories to 'destination' directory

copy\_if\_different <file>... destination - copy files if it has changed

echo [<string>...] - displays arguments as text

echo\_append [<string>...] - displays arguments as text but no new line

env [--unset=NAME]... [NAME=VALUE]... COMMAND [ARG]...

- run command in a modified environment

environment - display the current environment

make\_directory <dir>... - create parent and <dir> directories

md5sum <file>... - create MD5 checksum of files

sha1sum <file>... - create SHA1 checksum of files

sha224sum <file>... - create SHA224 checksum of files

sha256sum <file>... - create SHA256 checksum of files

sha384sum <file>... - create SHA384 checksum of files

sha512sum <file>... - create SHA512 checksum of files

remove [-f] <file>... - remove the file(s), use -f to force it

remove\_directory <dir>... - remove directories and their contents

rename oldname newname - rename a file or directory (on one volume)

server - start cmake in server mode

sleep <number>... - sleep for given number of seconds

tar [cxt][vf][zjJ] file.tar [file/dir1 file/dir2 ...] - create or extract a tar or zip archive

time command [args...] - run command and display elapsed time

touch <file>... - touch a <file>.

touch\_nocreate <file>... - touch a <file> but do not create it.

create\_symlink old new - create a symbolic link new -> old

Available on Windows only:

delete\_regv key - delete registry value

env\_vs8\_wince sdkname - displays a batch file which sets the environment for the provided Windows CE SDK installed in VS2005

env\_vs9\_wince sdkname - displays a batch file which sets the environment for the provided Windows CE SDK installed in VS2008

write\_regv key value - write registry value

[复制代码](javascript:void(0);)

 ++++++++++++++++++++++++

参考：

https://cmake.org/cmake/help/latest/genindex.html

https://yemi.me/2017/10/04/cmake-notes/

https://www.cnblogs.com/alphagl/p/6280061.html

https://stackoverflow.com/questions/14368919/cmake-custom-command-copy-multiple-files/54446162#54446162