

# cmake(六)Cmake添加工程子目录

原创

wzj\_110

于 2021-04-17 11:16:37 发布


阅读量7.8k

收藏 8

点赞数 1

分类专栏: cmake DSL语言

版权

cmake DSL语言 专栏收录该内容

38 篇文章

已订阅

- 1 | 重点: 'cmake3' 和 'make' 命令'输出'的'深刻解读'
- 2 |
- 3 | 备注: 当前阶段暂时不使用'IDE'工具,先'熟悉各指令'

## 一 ADD\_SUBDIRECTORY语法

应用场景: 一般情况下,我们的项目'各个子项目'都在一个'总的项目根'目录下,但有的时候,我们需要使'用外部的文件夹'

### cmake项目模块布局

**ADD\_SUBDIRECTORY指令:**  
**ADD\_SUBDIRECTORY(src\_dir [bin\_dir [EXCLUDE\_FROM\_ALL]])**  
**src\_dir:** 把src\_dir这个子目录加入工程  
**bin\_dir:** 指定编译输出(包含编译的中间结果)的目录为bin\_dir,如果不指定,则默认输出目录为src\_dir,即与src\_dir同名的bin眼录;指定**ignore**其实就是把src\_dir重命名为bin\_dir  
**EXCLUDE\_FROM\_ALL:** 将给定的目录从编译中排除,即**不编译它**,比如一些示例的目录,需要等工程构建完后单独构建的**Demo**  
**疑惑:**子目录的示例目录吗

**CMake提供了ADD\_SUBDIRECTORY指令,用于向当前工程添加子目录,并且可以指定中间二进制和目标二进制文件存放的位置,还可以将指定目录排除出编译过程**

## 二 实践1

### ① 初始化

- 1 | '外部文件夹'引入理解: 不在'CMakeLists'目录下 -->(与其'平级'或者还在'其外'),这里'不是'指的项目'根目录外'的文件夹,更多的是
- 2 |
- 3 | ++++++'初始化的流程'++++++
- 4 |
- 5 | 1) 创建'SubDirectory项目'的文件夹,并'cd'进入
- 6 |
- 7 | 2) 创建'SubDirectory项目'的源文件目录-->'src'
- 8 |
- 9 | 3) 创建'SubDirectory项目'的'CMakeLists.txt'文件

```
kiosk@k8s CmakeProjects $ ls          知道当前的位置
HelloCmake HelloLibrary
kiosk@k8s CmakeProjects $ mkdir SubDirectory
kiosk@k8s CmakeProjects $ cd SubDirectory
kiosk@k8s SubDirectory $ mkdir src
kiosk@k8s SubDirectory $ touch CMakeLists.txt
kiosk@k8s SubDirectory $ ls
CMakeLists.txt src                    初始化后SubDirectory下的内容
kiosk@k8s SubDirectory $
```

## ② 编写SubDirectory项目的CMakeLists.txt文件

1 场景1: 要添加的'src文件夹'是当前项目'SubDirectory'的子目录-->add\_library只指定一个指令'即可

2

3

细节: src既是'要添加的子目录'、也是'(子目录)编译输出保存'的路径-->用'子目录名'主要是'标识'是哪个'子目录'cmake3的'产物'

```
1 #####说明: 每个子项目都需要一个CMakeLists.txt文件在描述自身
2 # 1) cmake版本最低要求
3 cmake_minimum_required(VERSION 3.10)
4 # 2) 项目的名称-->一定要与CMakeLists.txt文件所在的路径对应
5 project(SubDirectory)
6 # 3) 指定源文件子目录为src
7 add_subdirectory(src)
```

SubDirectory目录下

"CMakeLists.txt" 7L, 312C

7,1 All

## ③ 继续

进入'src'源文件子目录,创建'示范性的源文件'-->main.cpp

```
kiosk@k8s SubDirectory $ ls
CMakeLists.txt src
kiosk@k8s SubDirectory $ cd src/
kiosk@k8s src $ ls
kiosk@k8s src $ vim main.cpp
```

备注: std::endl '应为' std::endl

```
1 #include<iostream>
2 #include <stdlib.h>
3
4 int main(int argc, char** argv) {
5     std::cout << "this is sub directory project" << std::endl;
6     return EXIT_SUCCESS;
7 }
```

src目录下

"main.cpp" 7L, 164C

3,0-1 All

## ④ 编写src目录的CMakeLists.txt文件

备注: 子目录'src'并没有指定'project'

```
1 # 只简单的添加生成可执行文件
2 # ② 基于main.cp生成可执行二进制文件名-->sub_directory
3 add_executable(sub_directory main.cpp)
```

```
kiosk@k8s src $ ls
main.cpp
kiosk@k8s src $ vim CMakeLists.txt
kiosk@k8s src $ vim CMakeLists.txt
kiosk@k8s src $ ls
CMakeLists.txt  main.cpp
kiosk@k8s src $
```

##### ⑤ 采用外部构建

```
kiosk@k8s src $ vim CMakeLists.txt
kiosk@k8s src $ ls
CMakeLists.txt  main.cpp
kiosk@k8s src $ cd ../
kiosk@k8s SubDirectory $ ls
CMakeLists.txt  src
kiosk@k8s SubDirectory $ mkdir build
kiosk@k8s SubDirectory $ cd build/
kiosk@k8s build $ ls
kiosk@k8s build $
```

采用外部构建新建,SubDirectory目录下新建build目录

##### ⑥ 构建并测试

- 1 细节: 父项目'SubDirectory'根据自身的CMakeFiles.txt文件中的'add\_subdirectory'指令,也会对子项目'src'进行'cmake3'
- 2
- 3 验证: cmake3 .. 之后,子目录也声称了'Makefile'目录

```
kiosk@k8s SubDirectory $ ls
build  CMakeLists.txt  src
kiosk@k8s SubDirectory $ cd build/
kiosk@k8s build $ ls
kiosk@k8s build $ cmake3 .
-- The C compiler identification is GNU 4.8.5
-- The CXX compiler identification is GNU 4.8.5
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc - works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ - works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /var/ftp/pub/pub/cmake/test/CmakeProjects/SubDirectory/build
kiosk@k8s build $ ls
CMakeCache.txt  CMakeFiles  cmake_install.cmake  Makefile  src
kiosk@k8s build $ ll src/
total 16
drwxrwxr-x 3 kiosk kiosk 4096 Apr 17 10:44 CMakeFiles
-rw-rw-r-- 1 kiosk kiosk 1148 Apr 17 10:44 cmake_install.cmake
-rw-rw-r-- 1 kiosk kiosk 6253 Apr 17 10:44 Makefile
kiosk@k8s build $ make
Scanning dependencies of target sub_directory
[ 50%] Building CXX object src/CMakeFiles/sub_directory.dir/main.cpp.o
[100%] Linking CXX executable sub_directory
[100%] Built target sub_directory
kiosk@k8s build $ ls
CMakeCache.txt  CMakeFiles  cmake_install.cmake  Makefile  src
kiosk@k8s build $ ll src/
total 32
drwxrwxr-x 3 kiosk kiosk 4096 Apr 17 10:44 CMakeFiles
-rw-rw-r-- 1 kiosk kiosk 1148 Apr 17 10:44 cmake_install.cmake
-rw-rw-r-- 1 kiosk kiosk 6253 Apr 17 10:44 Makefile
-rwxrwxr-x 1 kiosk kiosk 12712 Apr 17 10:45 sub_directory
kiosk@k8s build $ ./src/sub_directory
This is sub directory project!
kiosk@k8s build $
```

同时在build编译目录下创建src的子目录

build/src目录生成二进制可执行文件

测试

[https://blog.csdn.net/wzj\\_110](https://blog.csdn.net/wzj_110)

- 1 ++++++'add\_subdirectory'效果'++++++
- 2
- 3 1) 首先会'自动'调用该'子目录(src)'下的CMakeLists.txt文件
- 4
- 5 2) 还会在'build目录(out of build:外部构建)'生成对应的'目标(target)'文件
- 6
- 7 备注1: 体现在CMAKE\_CURRENT\_BINARY\_DIR的变量,通过'打印'该变量查看
- 8
- 9 备注2: add\_subdirectory(src target) -->默认是'src',在build目录会变成'target'
- 10
- 11 补充1: 任意一个'目录源文件(Source Tree)'对应一个'构建目录(Build Tree)'的target
- 12
- 13 补充2: include(src/CMakeLists.txt)'破坏'了'target'目录结构构造'的功能

## 参考博客

## 三 实践2

### ① 清理上次实验的环境

```
kiosk@k8s build $ ./src/sub_directory
This is sub_directory project!
kiosk@k8s build $ ls
CMakeCache.txt CMakeFiles cmake_install.cmake Makefile src
kiosk@k8s build $ rm -fr *
kiosk@k8s build $
```

### ② 演示bin\_dir

```
1 #####说明：每个子项目都需要一个CMakeLists.txt文件在描述自身
2 # 1) cmake版本最低要求
3 cmake_minimum_required(VERSION 3.10)
4 # 2) 项目的名称 -->一定要与CMakeLists.txt文件所在的路径对应
5 project(SubDirectory)
6 # 3) 指定源文件子目录为src编译输出保存到bin目录下
7 add_subdirectory(src bin)
```

### ③ 测试

```
kiosk@k8s build $ rm -fr *
kiosk@k8s build $ vim ../CMakeLists.txt
kiosk@k8s build $ cmake3 .
-- The C compiler identification is GNU 4.8.5
-- The CXX compiler identification is GNU 4.8.5
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc - works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ - works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /var/ftp/pub/pub/cmake/test/CmakeProjects/SubDirectory/build
kiosk@k8s build $ ls
bin CMakeCache.txt CMakeFiles cmake_install.cmake Makefile
kiosk@k8s build $ ll bin/
total 16
drwxrwxr-x 3 kiosk kiosk 4096 Apr 17 11:04 CMakeFiles
-rw-rw-r-- 1 kiosk kiosk 1148 Apr 17 11:04 cmake_install.cmake
-rw-rw-r-- 1 kiosk kiosk 6253 Apr 17 11:04 Makefile
kiosk@k8s build $ make
Scanning dependencies of target sub_directory
[ 50%] Building CXX object bin/CMakeFiles/sub_directory.dir/main.cpp.o
[100%] Linking CXX executable sub_directory
[100%] Built target sub_directory
kiosk@k8s build $ ll bin/
total 32
drwxrwxr-x 3 kiosk kiosk 4096 Apr 17 11:04 CMakeFiles
-rw-rw-r-- 1 kiosk kiosk 1148 Apr 17 11:04 cmake_install.cmake
-rw-rw-r-- 1 kiosk kiosk 6253 Apr 17 11:04 Makefile
-rwxrwxr-x 1 kiosk kiosk 12712 Apr 17 11:05 sub_directory
kiosk@k8s build $ ./bin/sub_directory
This is sub_directory project!
```

## 四 注意事项

### ① 超出平级目录

只要'不是上下级关系' (比如: hello/ 和 world/ 目录在其他目录下或者是 test/ 目录的上级), 在使用 add\_subdirectory() 时都'需要指

在使用 add\_subdirectory(source\_dir [binary\_dir] [EXCLUDE\_FROM\_ALL]) 命令时, 如果 source\_dir 不是当前目录 (CMakeLists.txt 所在目录, 例子中的 test/ 目录) 的子目录, 那么就需要显式指定 [binary\_dir] 参数, 用于存储 source\_dir 相关文件。

### ② 报错信息

- 1 注意: 如果 src\_dir '不是'当前目录(CMakeLists.txt 所在目录)的'子目录',那么就需要'显式指定' '[binary\_dir]' 参数,用于'存储
- 2
- 3 解读: 如果指定'src\_dir(子目录)',但是'src\_dir'不是当前项目的'子目录',则'cmake3'输出'默认会保存在'src\_dir'目录,而'当前项



```
CMake Error at CMakeLists.txt:10 (add_subdirectory):
  add_subdirectory not given a binary directory but the given source
  directory "/home/sdc/pro/hello-world/hello" is not a subdirectory of
  "/home/sdc/pro/hello-world/test".  When specifying an out-of-tree source a
  binary directory must be explicitly specified.
```

编译生成的输出必须保存在当前项目下

[参考博客1](#)

[参考博客2](#)

cmake生成完毕后[自动执行](#)其他命令

显示推荐内容

1 条评论



xcp881012 [热评](#) 老哥，代码有对应的git 地址 可以clone下嘛

[写评论](#)