


cmake(四)CMake项目的二进制目录和源目录

原创wzj_110于 2021-04-15 23:50:01 发布阅读量1.8k收藏4点赞数3

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

摘要 这篇博客介绍了如何管理和修改二进制目录与源代码目录，强调了环境变量的调整以及CMakeLists.txt文件的编辑在软件构建过程中的重要性。通过实际操作步骤，包括环境修改、CMakeLists更新和测试验证，来阐述最佳实践，帮助开发者优化项目构建流程。

摘要由CSDN通过智能技术生成

一 二进制目录和源目录

知道‘在哪(DIR)’很重要

CMake会为每个项目的二进制目录和源目录隐式生成两个变量：

```
<project_name>_BINARY_DIR
<project_name>_SOURCE_DIR
```

同时也存在这样的两个变量：

```
PROJECT_BINARY_DIR = <project_name>_BINARY_DIR
PROJECT_SOURCE_DIR = <project_name>_SOURCE_DIR
```

https://blog.csdn.net/wzj_110

① 环境修改

```
kiosk@k8s CmakeProjects $ ls
HelloCmake
kiosk@k8s CmakeProjects $ cd HelloCmake/
kiosk@k8s HelloCmake $ ls
build CMakeLists.txt HelloCmake.cpp
kiosk@k8s HelloCmake $ cd build/
kiosk@k8s build $ ls
CMakeCache.txt CMakeFiles
kiosk@k8s build $ rm -fr *
kiosk@k8s build $ vim ../CMakeLists.txt
```

② 修改CMakeLists.txt文件

```
1 # 项目名称
2 project(HelloCmake)
3 set(SRC_LIST HelloCmake.cpp)
4 message("This is no paraments!")
5 message(STATUS "This is a status level message!")
6 # 添加的信息 -->看相关内置变量
7 # 1) 二进制目录的区别
8 # 备注：隐式变量 --> ${${project_name}_BINARY_DIR} 后缀
9 message(STATUS ${HelloCmake_BINARY_DIR})
10 message(STATUS ${PROJECT_BINARY_DIR})
11 # 2) 源目录的区别
12 message(STATUS ${HelloCmake_SOURCE_DIR})
13 message(STATUS ${PROJECT_SOURCE_DIR})
14 add_executable(hello_cmake ${SRC_LIST})
```

名称固定

https://blog.csdn.net/wzj_110

③ 测试

```

kiosk@k8s build $ rm -fr *
kiosk@k8s build $ vim ../CMakeLists.txt 修改文件
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
This is no paraments!
-- This is a status level message!
-- /var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake/build
-- /var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake/build
-- /var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake
-- /var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake
-- Configuring done
-- Generating done
-- Build files have been written to: /var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake/build
kiosk@k8s build $ ls
CMakeCache.txt CMakeFiles cmake_install.cmake Makefile
kiosk@k8s build $ pwd
/var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake/build

```

小结：从输出看隐式和显示的输出一样

二进制的目录

源码目录

https://blog.csdn.net/wrj_110

HelloCMake_BINARY_DIR = PROJECT_BINARY_DIR
HelloCMake_SOURCE_DIR = PROJECT_SOURCE_DIR

PROJECT_BINARY_DIR对应的是执行cmake命令所在的目录，这里是build这个子目录的路径，而**PROJECT_SOURCE_DIR**则对应源文件的目录，这里是build的上一层目录，即CMakeLists.txt对应的目录

④ 最佳实践

实际使用时，用**PROJECT_XXX**这个变量比较好，因为这两个变量不会因为项目名称的改变而改变，<project_name>_XXX这个变量则要随项目名称的改变而更改

https://blog.csdn.net/wrj_110

显示推荐内容