cmake(八)Cmake定义安装

原创 wzj_110 于 2021-04-18 19:53:58 发布 阅读量632 收藏 2 点赞数 5 版权
分类专栏： cmake DSL语言

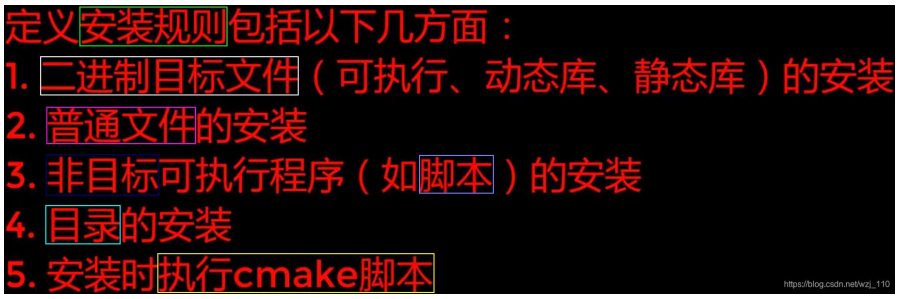
一 cmake安装汇总

目录结构

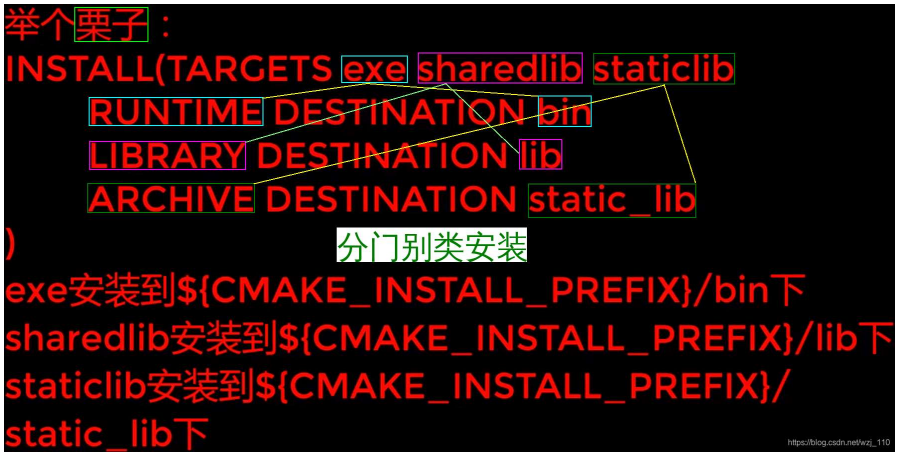
参考博客

① 语法规则

② 安装规则



③ 二进制目标文件



参数中的`TARGET`可以是很多种目标文件，最常见的是**通过ADD_EXECUTABLE或者ADD_LIBRARY定义的目标文件，即可执行二进制、动态库、静态库：**

目标文件	内容	安装目录变量	默认安装文件夹
ARCHIVE	静态库	${CMAKE_INSTALL_LIBDIR}	lib
LIBRARY	动态库	${CMAKE_INSTALL_LIBDIR}	lib
RUNTIME	可执行二进制文件	${CMAKE_INSTALL_BINDIR}	bin
PUBLIC_HEADER	与库关联的PUBLIC头文件	${CMAKE_INSTALL_INCLUDEDIR}	include
PRIVATE_HEADER	与库关联的PRIVATE头文件	${CMAKE_INSTALL_INCLUDEDIR}	include

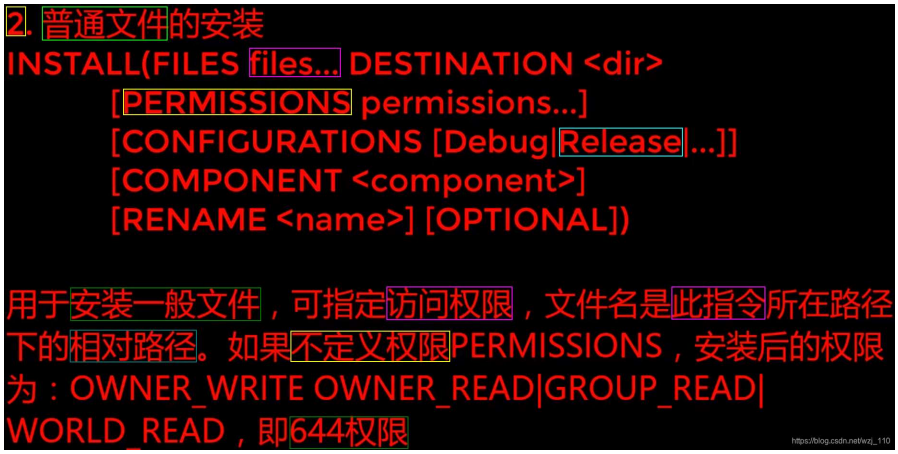
为了符合一般的默认安装路径，如果设置了`DESTINATION`参数，推荐配置在安装目录变量下的文件夹

```
INSTALL(TARGETS myrun mylib mystaticlib
        RUNTIME DESTINATION ${CMAKE_INSTALL_BINDIR}
        LIBRARY DESTINATION ${CMAKE_INSTALL_LIBDIR}
        ARCHIVE DESTINATION ${CMAKE_INSTALL_LIBDIR}
)
```

与后边目标文件一一匹配

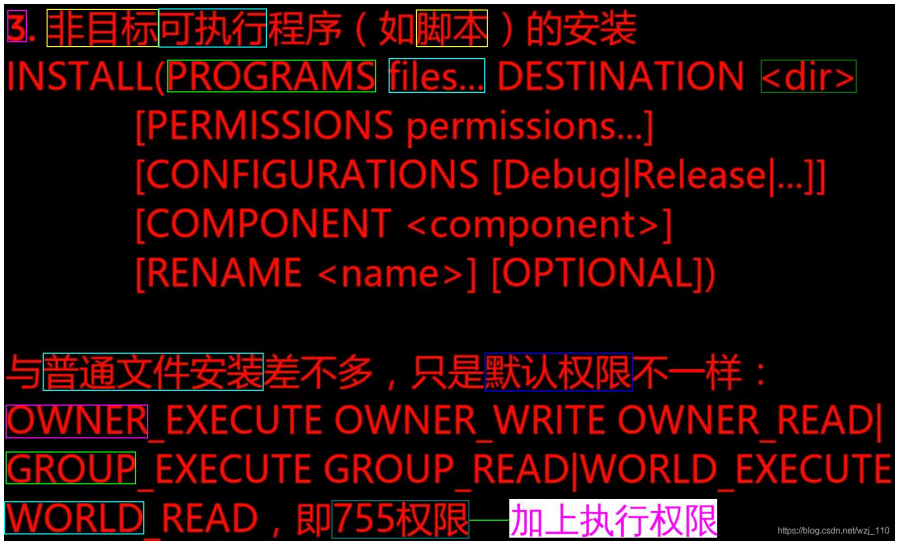
上面的例子会将：可执行二进制`myrun`安装到`${CMAKE_INSTALL_BINDIR}`目录，动态库`libmylib.so`安装到`${CMAKE_INSTALL_LIBDIR}`目录，静态库`libmystaticlib.a`安装到`${CMAKE_INSTALL_LIBDIR}`目录。

④ 普通文件安装

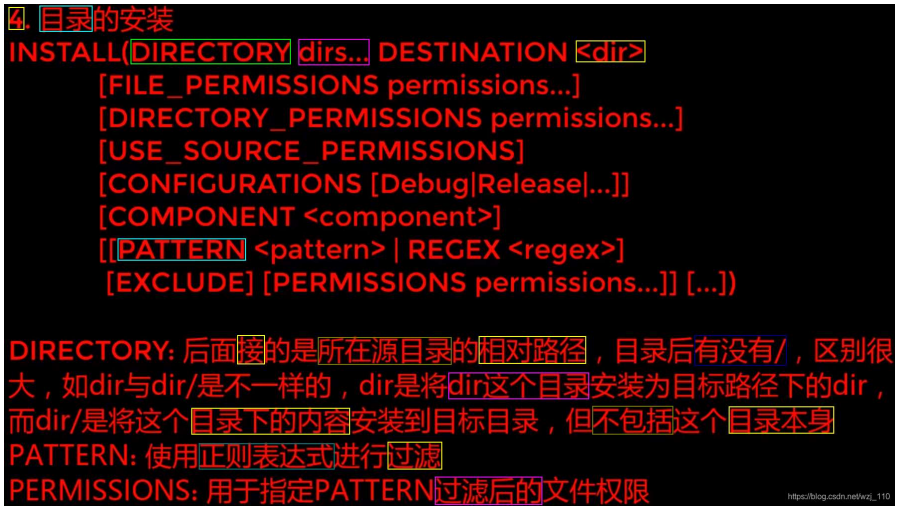


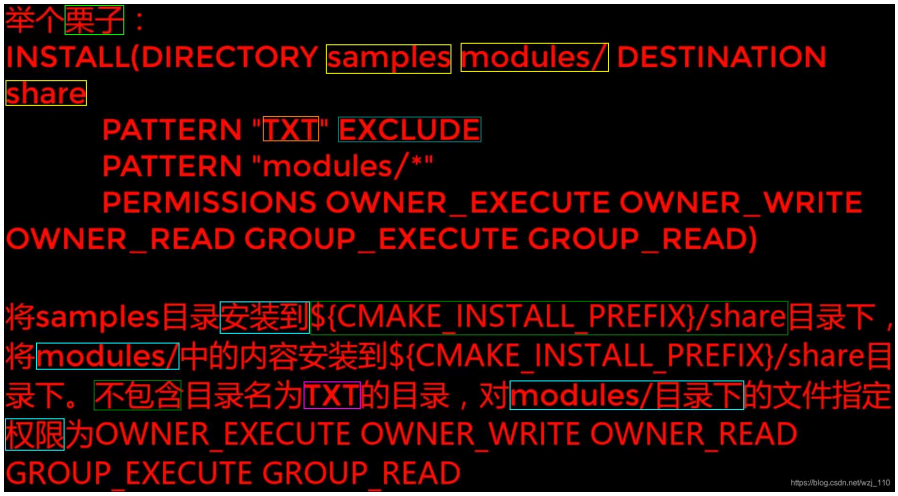
⑤ 非目标可执行程序

备注： 一般是'sh'、'python'脚本



⑥ 目录的安装



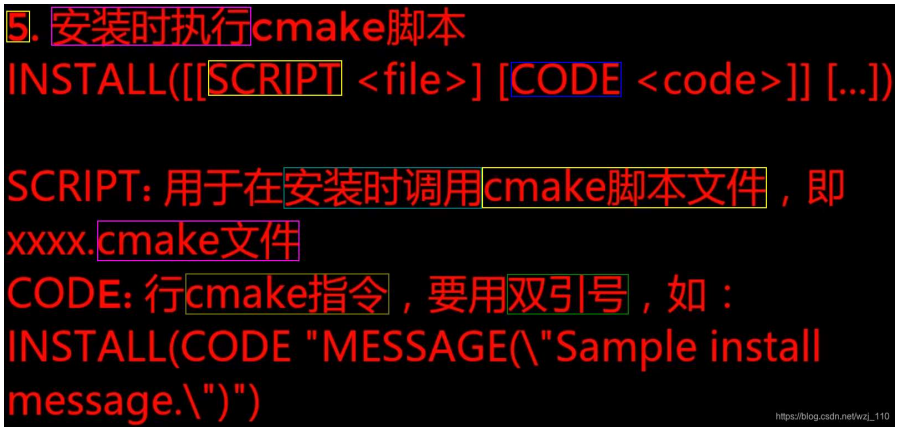


⑦ 安装时执行的cmake脚本

```
find  / -name *.cmake

备注： 注意'cmake'后缀文件的'编写'语法

备注： []是'非必须'
```



参考博客

二 实践

① 项目初始化

```
1  ++++++++++++++'执行步骤'++++++++++++++
2
3  1）创建'项目'目录,'进入'该'项目'目录
4
5  2）创建一个'src'子目录 -->存放'源文件目录'
6
7  3）创建一个'doc'子目录 -->存放'文档文件'
8
9  4）创建'copyright版权文件'、'readme操作文件'
```

```
kiosk@k8s CmakeProjects $ ls
HelloCmake  HelloLibrary  OutputPath  SubDirectory
kiosk@k8s CmakeProjects $ mkdir CustomizeInstall        项目名称
kiosk@k8s CmakeProjects $ cd CustomizeInstall/
kiosk@k8s CustomizeInstall $ mkdir {src,doc}
kiosk@k8s CustomizeInstall $ touch {copyright,readme}
kiosk@k8s CustomizeInstall $ tree .
.
├── copyright
├── doc
├── readme
└── src

2 directories, 2 files
```

② Main.cpp编写

```
#include <iostream>
#include <stdlib.h>

int main(int agrc, char** agrv) {
    std::cout << "Customize Install:自定义安装" << std::endl;
    return EXIT_SUCCESS;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
"Main.cpp" [Modified][New file] 7 lines --85%--                6,24          All
```

③ src子目录下CMakeLists.txt文件编写

```
# 1) 设置可执行目标输出的路径
# 备注：bin目录会自动创建
set(EXECUTABLE_OUTPUT_PATH ${PROJECT_BINARY_DIR}/bin)
# 2) 生成可执行的二进制文件
add_executable(customize_install Main.cpp)
# 3) 指定这个可执行二进制要安装到的目录
# 备注：会安装到${CMAKE_INSTALL_PREFIX}/CustomizeInstall/bin
install(TARGETS customize_install DESTINATION CustomizeInstall/bin)
```

CMakeLists.txt

安装的前缀：可命令行或文件中指定

src目录自身的

"CMakeLists.txt" 8 lines --100%-- 8,1 All

```
kiosk@k8s CmakeProjects $ tree CustomizeInstall/
CustomizeInstall/
├── copyright
├── doc
├── readme
└── src
    ├── CMakeLists.txt
    └── Main.cpp

2 directories, 4 files
kiosk@k8s CmakeProjects $
```

子目录自身的

当前阶段的文件预览

④ 项目主目录编写CMakeLists.txt文件

项目主目录下的CMakeLists.txt文件

```
cmake_minimum_required(VERSION 3.8)
project(CmakeProjects)

#1) 把src源文件子目录加入到项目中
add_subdirectory(src bin)

#2) 把copyright、readme这两个文件安装到目录-->${CMAKE_INSTALL_PREFIX}/CustomizeInstall下
#备注：FILES是关键字      -->普通文件类型
install(FILES copyright readme DESTINATION CustomizeInstall)

#3) 非目标的'可执行程序'的安装
#备注1：PROGRAMS是关键字 -->非目标的可执行程序
#备注2：把run_customize_install.sh安装到目录：${CMAKE_INSTALL_PREFIX}/CustomizeInstall/bin下
install(PROGRAMS run_customize_install.sh DESTINATION CustomizeInstall/bin)

#4) 把doc/目录下的文件(由于/的原因-->不包含这个目录自身)安装到目录：${CMAKE_INSTALL_PREFIX}/
CustomizeInstall/share/doc下
#备注1：DIRECTORY是关键字
#备注2：等价-->doc DESTINATION CustomizeInstall/share/
install(DIRECTORY doc/ DESTINATION CustomizeInstall/share/doc)
```

"CMakeLists.txt" 19 lines --5%-- 1,1 All

```
项目的'主目录下'编写'run_customize_install.sh'文件

vim run_customize_install.sh
```

```
kiosk@k8s CustomizeInstall $ cat run_customize_install.sh
./customize_install
kiosk@k8s CustomizeInstall $
```

脚本作用：简单执行customize_install命令

⑤ 项目主目录创建doc目录

```
kiosk@k8s CustomizeInstall $ mkdir doc
kiosk@k8s CustomizeInstall $ cd doc/
kiosk@k8s doc $ echo 'CustomizeInstall项目的文档说明' > customize_install_doc.txt
kiosk@k8s doc $ ls
customize_install_doc.txt
kiosk@k8s doc $
```

简单来个文件表示文档文件

⑥ 项目主目录创建build目录

说明： build目录来进行'外部构建'

```
kiosk@k8s doc $ cd ../
kiosk@k8s CustomizeInstall $ mkdir build
kiosk@k8s CustomizeInstall $ tree .
.
├── build
├── CMakeLists.txt
├── copyright
├── doc
│   └── customize_install_doc.txt
├── readme
├── run_customize_install.sh
└── src
    ├── CMakeLists.txt
    └── Main.cpp

3 directories, 7 files
```

cmake之前项目最终的结构

⑦ 执行cmake

CMake命令行参数

```
-D <var>:<type>=<value>, -D <var>=<value>
      Create or update a CMake CACHE entry.

      When CMake is first run in an empty build tree, it creates a CMakeCache.txt
      file and populates it with customizable settings for the project. This option
      may be used to specify a setting that takes priority over the project's default
      value. The option may be repeated for as many CACHE entries as desired.

      If the :<type> portion is given it must be one of the types specified by the
      set() command documentation for its CACHE signature. If the :<type> portion is
      omitted the entry will be created with no type if it does not exist with a type
      already. If a command in the project sets the type to PATH or FILEPATH then
      the <value> will be converted to an absolute path.

      This option may also be given as a single argument: -D<var>:<type>=<value> or
      -D<var>=<value>.
```

部分

省略

```
1 一般： 我们会在'CMakeLists.txt'文件指定默认的'PREFIX',如果需要'自定义',则命令行进行'覆盖'
2
3 cmake_install.cmake '验证'是否'覆盖'
```

```
kiosk@k8s build $ pwd
/var/ftp/pub/pub/cmake/test/CmakeProjects/CustomizeInstall/build
kiosk@k8s build $ ls
kiosk@k8s build $ cmake3 -DCMAKE_INSTALL_PREFIX=/tmp/install ..
-- 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/CustomizeInstall/build
kiosk@k8s build $ ls
bin  CMakeCache.txt  CMakeFiles  cmake_install.cmake  Makefile
kiosk@k8s build $ ll bin/
total 20
drwxrwxr-x 3 kiosk kiosk 4096 Apr 18 19:38 CMakeFiles
-rw-rw-r-- 1 kiosk kiosk 2147 Apr 18 19:38 cmake_install.cmake
-rw-rw-r-- 1 kiosk kiosk 8393 Apr 18 19:38 Makefile
kiosk@k8s build $ make
Scanning dependencies of target customize_install
[ 50%] Building CXX object bin/CMakeFiles/customize_install.dir/Main.cpp.o
[100%] Linking CXX executable customize_install
[100%] Built target customize_install
kiosk@k8s build $ ll bin/
total 36
drwxrwxr-x 3 kiosk kiosk  4096 Apr 18 19:38 CMakeFiles
-rw-rw-r-- 1 kiosk kiosk  2147 Apr 18 19:38 cmake_install.cmake
-rwxrwxr-x 1 kiosk kiosk 12712 Apr 18 19:39 customize_install
-rw-rw-r-- 1 kiosk kiosk  8393 Apr 18 19:38 Makefile
kiosk@k8s build $ bin/customize_install
Customize Install:自定义安装
kiosk@k8s build $
```

通过命令名指定

二进制可执行文件

测试

⑧ make install 之后观察/tmp/install目录

备注： 符合'预期'

```
kiosk@k8s build $ ll /tmp/install/
total 0
kiosk@k8s build $ make install                    提示信息可知：按照CMakeList.txt指定路径进行安装
[100%] Built target customize_install
Install the project...
-- Install configuration: ""
-- Installing: /tmp/install/CustomizeInstall/copyright
-- Installing: /tmp/install/CustomizeInstall/readme
-- Installing: /tmp/install/CustomizeInstall/bin/run_customize_install.sh
-- Installing: /tmp/install/CustomizeInstall/share/doc
-- Installing: /tmp/install/CustomizeInstall/share/doc/customize_install_doc.txt
-- Installing: /tmp/install/CustomizeInstall/bin/customize_install
kiosk@k8s build $ tree /tmp/install/
/tmp/install/                                     安装之后的结构
└── CustomizeInstall
    ├── bin
    │   ├── customize_install
    │   └── run_customize_install.sh
    ├── copyright
    ├── readme
    └── share
        └── doc
            └── customize_install_doc.txt

4 directories, 5 files
```

```
kiosk@k8s build $ tree /tmp/install/
/tmp/install/
└── CustomizeInstall
    ├── bin
    │   ├── customize_install
    │   └── run_customize_install.sh
    ├── copyright
    ├── readme
    └── share
        └── doc
            └── customize_install_doc.txt

4 directories, 5 files
kiosk@k8s build $ cd /tmp/install/CustomizeInstall/bin/
kiosk@k8s bin $ cat run_customize_install.sh
./customize_install
kiosk@k8s bin $ ./run_customize_install.sh          符合预期
Customize Install:自定义安装
kiosk@k8s bin $
```

显示推荐内容