## CMake入门B01\_Clion搭建QT的开发环境

### 测试环境：

**Windows 10（Ubuntu）+MinGw（gcc/g++/Ninja.exe）+ Clion+QT**

**插图以及内容来自书籍《Modern CMake For C++》**

**辅助工具：ChatGPT**

### 本节主要的内容：

* 为什么要使用Clion IDE
* 安装Clion和QT
* 编写一个QT GUI程序以及详解；
* 在Clion里面使用[Qt Designer](https://doc.qt.io/qt-5/qtdesigner-manual.html)；
* Qt UI Class模板（把启动界面换成QMainWindow）
* 怎么添加其他QT组件；（比如网络组件）
* 生成应用图标icon；
* 添加图片资源；
* 和Vcpkg集成；
* 使用中的常见问题；

1. 找不到QT的问题；
2. 添加Release版本和Debug版本
3. 使用第三方库；(可以参考我07讲视频)
4. Windows平台下，解决生成的执行文件有console 窗口的问题；
5. Ubuntu平台下，找不到头文件的问题；
6. Windows平台下，找不到动态库的问题；

### 为什么要使用Clion IDE

优点：

* IDE本身跨平台；
* 支持cmake构建，方便跨平台开发；
* 智能提示很强大；
* 内存资源占用比VS少；
* 对第三方包管理工具集成的较好；
* 对IDEA和WebStorm开发者上手比较快；

缺点：

* 跨平台不如Qt Creator；
* 需要折腾，解决Qt Creator里面不会出现的问题；

### 安装Clion和QT

#### Clion的安装

安装Clion很简单一步步安装就可以了，

Clion官网的安装教程：

https://www.jetbrains.com/help/clion/installation-guide.html#requirements

#### QT的安装

第一步，去QT的ftp下载一个在线安装器；

<https://download.qt.io/archive/online_installers/>

图形用户界面, 文本, 应用程序

描述已自动生成

推荐使用安装MinGW 64版本；

如果需要MSVC生成器，可以下载MSVC版本；



### 编写一个QT GUI程序；

CMake版本要求：

make\_minimum\_required() 指定应用程序所需的最低 CMake 版本。

Qt 本身至少需要 CMake 版本 3.16。

如果您使用静态构建的 Qt - Qt for iOS 和 Qt for WebAssembly 的默认设置 – 您需要 CMake 3.21.1 或更新版本。

C++版本要求：

Qt 6 需要支持 C++ 版本 17 或更新版本的编译器。

如果编译器太旧，通过设置 CMAKE\_CXX\_STANDARD、CMAKE\_CXX\_STANDARD\_REQUIRED 变量

强制执行此操作将使 CMake 打印错误。

文本

描述已自动生成

*#POST\_BUILD是CMake的一个关键字，表示在构建目标完成后执行指定的命令  
#TARGET\_FILE\_DIR是CMake变量，表示目标二进制文件的输出路径  
#generator expression，只有在生成器中被解析时才会被替换为实际的路径*

*# -E表示命令模式；*

图形用户界面, 文本

描述已自动生成

Cmake -E –help命令，查看命令模式下支持的命令；

文本

描述已自动生成

### 在Clion里面使用[Qt Designer](https://doc.qt.io/qt-5/qtdesigner-manual.html)；

File | Settings | Tools | External Tools

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

* D:\Qt\QT\6.2.4\mingw\_64\bin\designer.exe
* $FileName$
* $FileDir$

### Qt UI Class模板（把启动界面换成QMainWindow）

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

### 怎么添加其他QT组件；

https://doc.qt.io/qt-6.2/qtmodules.html

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

比如我们添加QT的Network的模块;

图形用户界面, 文本, 应用程序

描述已自动生成

这个地方，自动复制动态库到程序运行目录下：

图形用户界面, 文本

描述已自动生成

### 生成应用图标icon；

第一步：准备一个图标app.ico

App.ico

手机屏幕截图

描述已自动生成

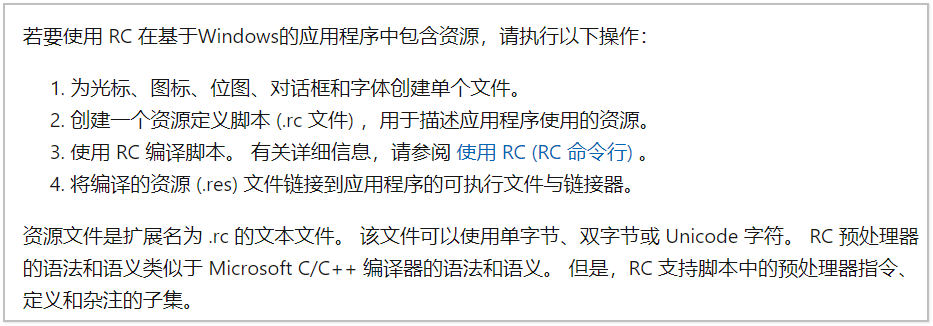
第二步：准备一个app.rc文件

图标

描述已自动生成

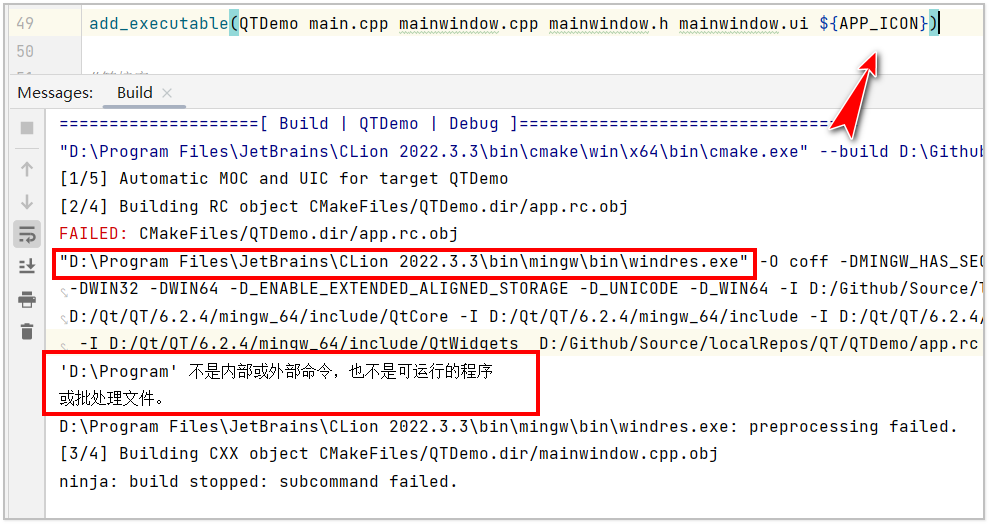


IDI\_ICON1 ICON DISCARDABLE "app.ico"



直接使用rc文件，因为安装路径导致创建目标文件失败；

Building RC object CMakeFiles/QTDemo.dir/app.rc.obj

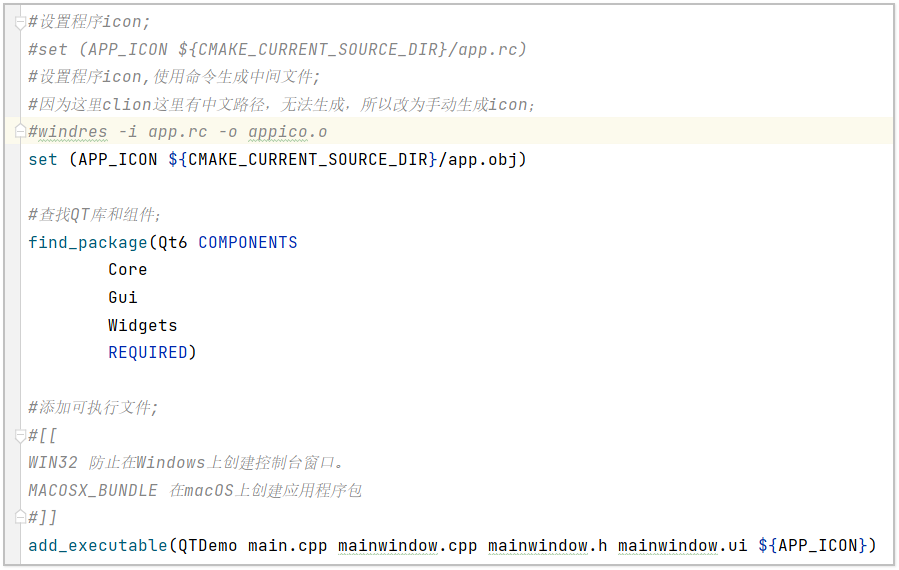


手动生成：

windres.exe -i app.rc -o app.obj

文本

描述已自动生成



### 添加资源文件；

图形用户界面, 应用程序

描述已自动生成

可以自行了解；

### 和vcpkg集成；

-DCMAKE\_TOOLCHAIN\_FILE=<vcpkg-root>/scripts/buildsystems/vcpkg.cmake

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

### 使用中的常见问题；

#### 找不到QT的问题；

*#设置QT的安装路径；*set(CMAKE\_PREFIX\_PATH "D:/Qt/QT/6.2.4/mingw\_64/lib/cmake")

#### 添加Release版本和Debug版本

图形用户界面, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 应用程序, Word

描述已自动生成

#### 使用第三方库；(可以参考我07讲视频)

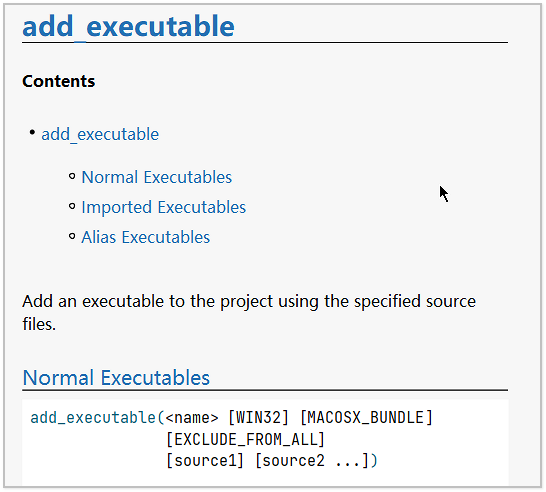
#### Windows平台下，解决生成的执行文件有console 窗口的问题；

直接运行.exe会出现黑色窗口；

图形用户界面, 文本, 应用程序

描述已自动生成

add\_executable(QTDemo WIN32 main.cpp mainwindow.cpp mainwindow.h mainwindow.ui)



#### .Windows平台下，找不到动态库的问题；

把动态库拷贝到执行目录；

add\_custom\_command(TARGET ${PROJECT\_NAME} POST\_BUILD  
 COMMAND ${CMAKE\_COMMAND} -E copy  
 "${QT\_INSTALL\_PATH}/bin/Qt6${QT\_LIB}${DEBUG\_SUFFIX}.dll"  
 "$<TARGET\_FILE\_DIR:${PROJECT\_NAME}>")

第三库也是如此，方法类似；

#### Ubuntu平台下，找不到头文件的问题；

大小写的问题；

### 总结

### 参考文献：

#### QT官网使用CMake

<https://doc.qt.io/qt-6.2/cmake-get-started.html>

#### QT所有可用组件；

<https://doc.qt.io/qt-6.2/qtmodules.html>

#### QT资源系统

https://doc.qt.io/qt-6.2/resources.html

#### jetbrains 官网创建Qt projects

<https://www.jetbrains.com/help/clion/qt-tutorial.html>

#### jetbrains 官网Clion集成Vcpkg

<https://blog.jetbrains.com/clion/2023/01/support-for-vcpkg-in-clion/>

#### 微软vcpkg官网CMake使用vcpkg

<https://learn.microsoft.com/en-us/vcpkg/users/buildsystems/cmake-integration>