

配置 Opencv 开发环境

作者: [Youngjam](#)

原文链接: <https://ld246.com/article/1680922707813>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

作者: yangjian

创建日期: 2023-02-23 22:32:50

版权说明: Copyright (c) 2023 yangjian. All rights reserved.

Tags: #opencv#

References:

□

windows 环境

通过 vcpkg 安装

- 安装

Opencv

- vcpkg install opencv4: x64-windwos

```
set(OpenCV_DIR "${VCPKG_INSTALLED_DIR}/x64-windows/share/opencv4")
find_package(OpenCV REQUIRED)
```

- 在 vscode 中通过 cmake 引用

该值需要设置为全局，在project之前设置

```
set(CMAKE_TOOLCHAIN_FILE E:/vcpkg/scripts/buildsystems/vcpkg.cmake)
set(OpenCV_DIR "${VCPKG_INSTALLED_DIR}/x64-windows/share/opencv4")
```

查找OpenCV包

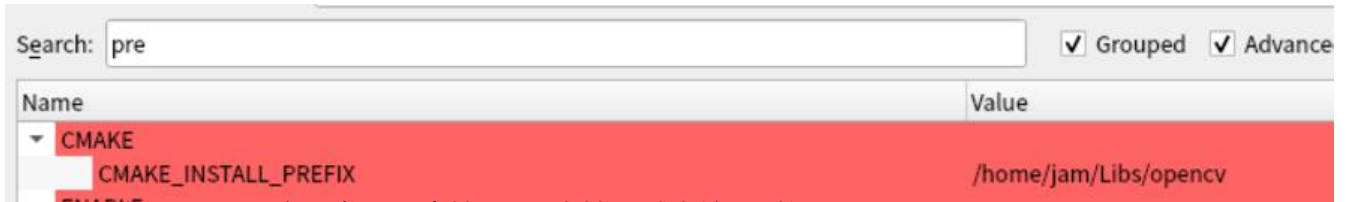
```
find_package(OpenCV REQUIRED)
if(OpenCV_FOUND)
    message("OpenCV found!")
    # 加入头文件路径
    include_directories(${OpenCV_INCLUDE_DIRS})
    # 添加OpenCV链接库
    link_libraries(${OpenCV_LIBS})
endif()
```

Linux

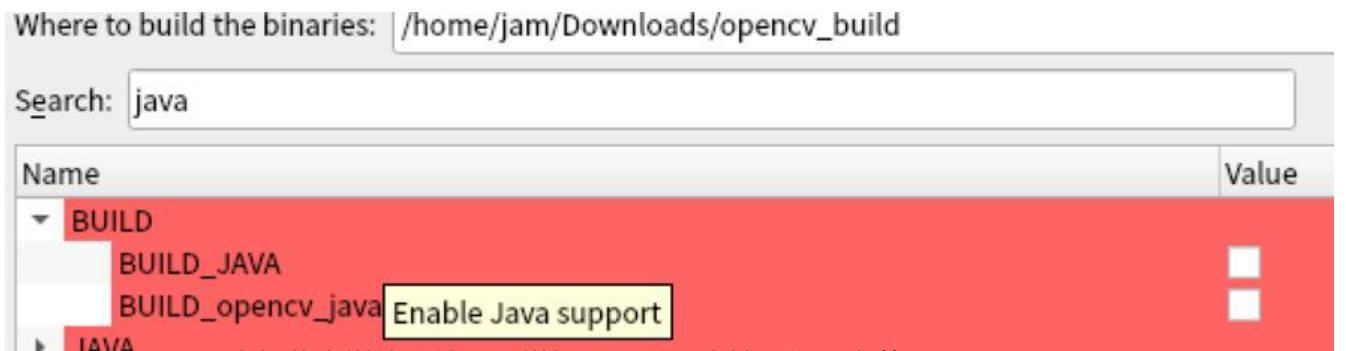
源码编译安装

1. oepncv 的 github 主页下载 opencv 和 opencv_contrib 源码，本文编译版本为 4.6.0
2. 分别创建 build 目录和 install 目录，build 目录用于存放编译的中间文件，install 目录用于存放最生成的 lib 和 include 文件
3. 在 build 目录下执行 cmake-gui

1. 选择源码路径和 build 路径，最好将 Ground 和 Advanced 勾选上，方便查看需要配置的编译选项
2. 选择编译工具链，可使用默认的本地工具链或指定交叉编译工具链
3. 配置 lib 和 include 文件的安装路径

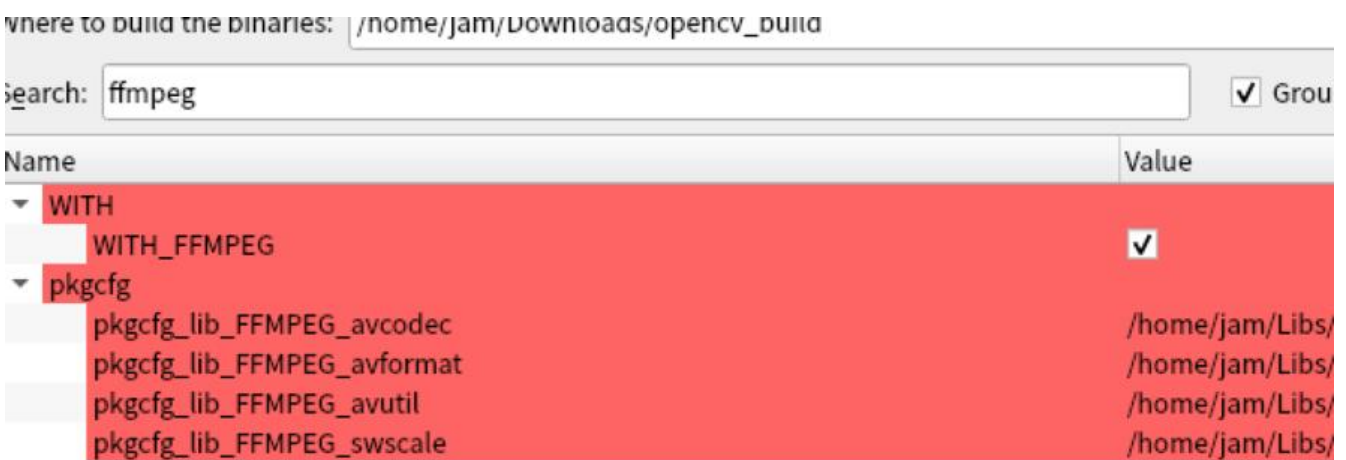


4. 只使用 c++ 语言开发，可去掉 java 支持，减少编译时间



5. 为了可以正常加载多媒体文件需要增加 ffmpeg 支持，如果安装了 pkg-config，且配置了 ffmpeg 的 pkgconfig 目录，勾选后会自动搜索到相关 lib 路径

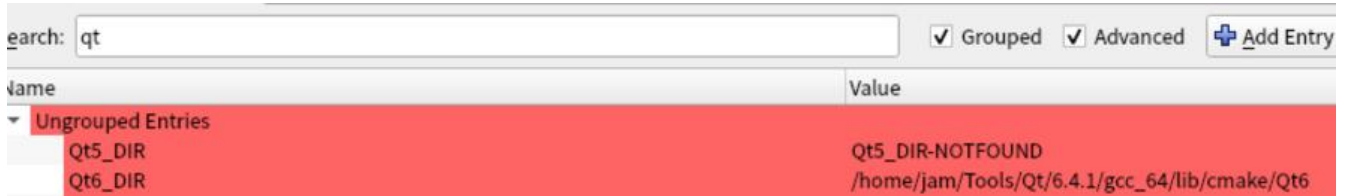
- 添加 ffmpeg 后编译时出现了，找不到相关定义的问题，原因是 ffmpeg 从 4.4 的版本开始出现了较大变化，导致 opencv 编译失败，需要重新编译低版本的 ffmpeg，实测 ffmpeg4.3.5 适配 opencv4.6.0



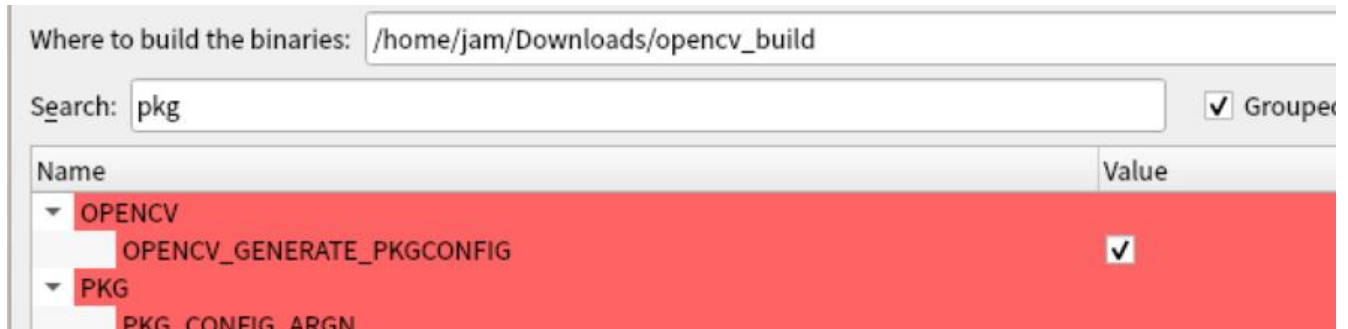
6. 如果安装了已经安装了 QT，可以开启 QT 支持，配置以下两个路径后，cmake 会自动搜索其库的路径

- 配置 qmake 目录，我安装的是 6.4.1，所以路径为 ".../6.4.1/gcc_64/bin/qmake"
- 配置 QT5 或 QT6 目录，路径为 ".../6.4.1/gcc_64/lib/cmake/QT6"



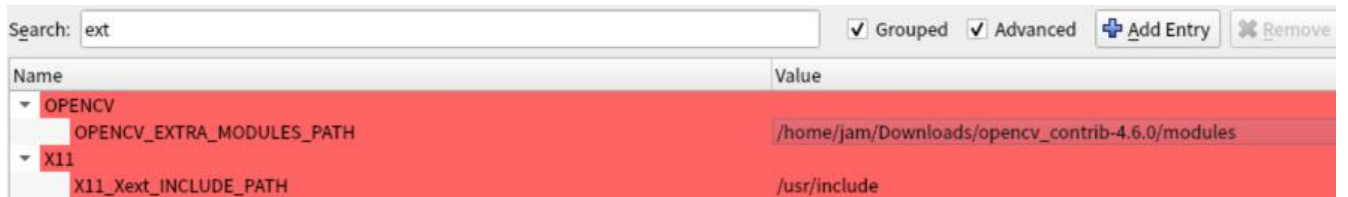


- 此外还可以开启 vulkan 和 opengl 支持, 方法同上
- 为了减少编译时间, 可以关闭所有 test
- 为了方便引用 opencv 库, 可以生成 pkg-config 文件



- 此外还可以通过配置 `BUILD_opencv_world` 将所有库打包为一个文件, 配置 `BUILD_SHARED_LIBS` 来决定生成动态库或静态库
- 如果需要使用人脸检测等功能, 需要添加 `opencv_contrib`, 配置如下路径

- 如果需要编译 contrib, 最好关闭 QT 编译支持, 实践发现需要将 QT 的 include 下的子目配置为系统路径, 且 QT 版本不匹配会导致编译失败



- 配置完后选择 `configure`, 检查输出信息是否有找不到的库或文件, 如果出现红色错误或警告最好解决后再进行编译

- 找不到库就手动安装, 或者去掉不需要的编译项
- 网络原因导致无法下载 github.raw 上的数据文件, 可以手动下载后替换, 具体方案可自行索

- 然后 `generate`, 没有错误后, 退出 `cmake-gui`

- 在 `build` 目录下执行 `make`
- 编译成功后, 再执行 `make install` 即可
- 如果生成的动态库, 需要配置动态库加载路径, 或者编译时指定 `rpath`

□