

## Windows10 环境中 CMake+VS2017 编译 OpenCV

### 1、概述

目前 OpenCV 官方发布的最新版本是 4.1.0，针对于 C++ 开发者而言官方发布的版本只有 x64 的相应库文件及动态链接库，那么如果需要开发 x86 的软件，就需要自己进行编译 x86 的库文件及动态链接库。此外官方版本中的相应函数只是稳定的可靠的函数，那么有些测试中的函数并没有包括在其中，如果需要使用这些测试中的函数要么需要自己写要么需要将 opencv 的 extra 模块中的内容编译进去，从而调用这些测试中的函数。

综上所述，对 OpenCV 进行编译是十分必要的，我们一般在 Windows 平台下借助于 CMake 及 VS 进行编译，下面做简单介绍。

### 2、准备工作

#### 1) 安装 CMake





本人在 Win10 x64 平台下进行编译，此处采用的 CMake 是 64 位的版本，版本号是 3.11.0。下载地址是 <https://github.com/Kitware/CMake/releases/tag/v3.11.0>

v3.11.0

robertmaynard released this on 1 Dec 2018 · 4561 commits to master since this release

PGP sig by EC8FEF3A7BFB4EDA

#### Assets 15

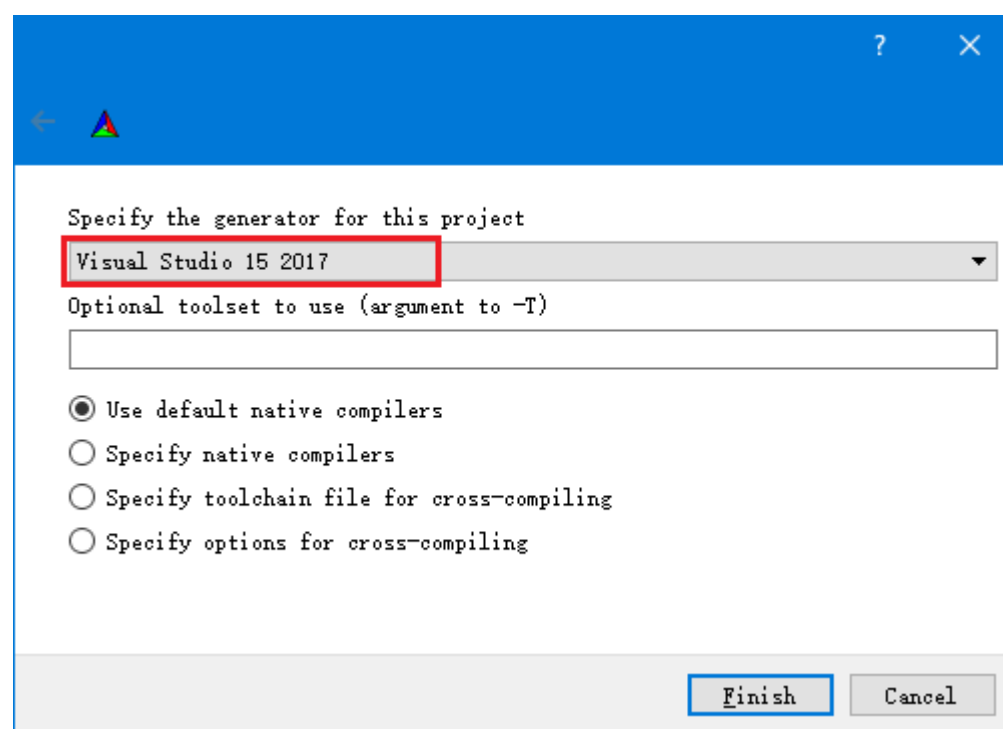
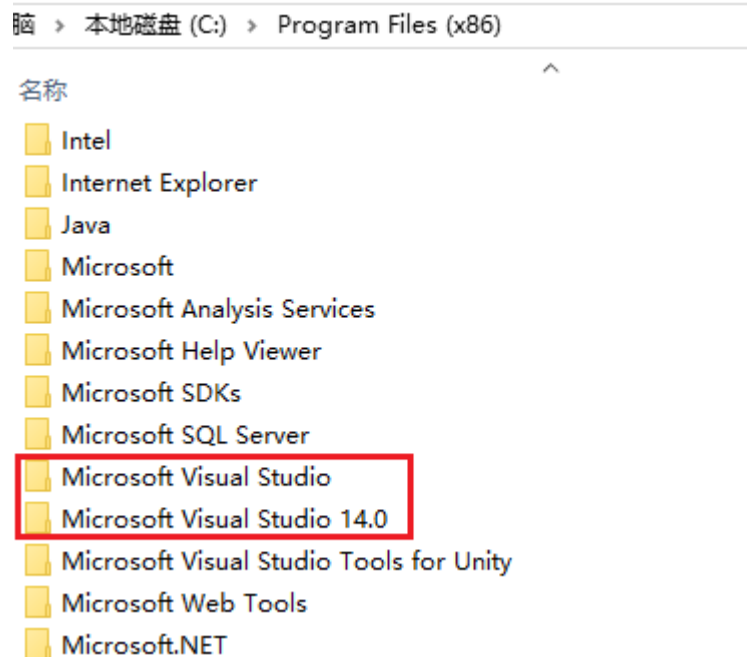
 <a href="#">cmake-3.11.0-Darwin-x86_64.dmg</a>	26.4 MB
 <a href="#">cmake-3.11.0-Darwin-x86_64.tar.gz</a>	26.1 MB
 <a href="#">cmake-3.11.0-Linux-x86_64.sh</a>	32.9 MB
 <a href="#">cmake-3.11.0-Linux-x86_64.tar.gz</a>	32.9 MB
 <a href="#">cmake-3.11.0-SHA-256.txt</a>	1017 Bytes
 <a href="#">cmake-3.11.0-SHA-256.txt.asc</a>	833 Bytes
 <a href="#">cmake-3.11.0-win32-x86.msi</a>	15.6 MB
 <a href="#">cmake-3.11.0-win32-x86.zip</a>	21.6 MB
 <a href="#">cmake-3.11.0-win64-x64.msi</a>	18.5 MB
 <a href="#">cmake-3.11.0-win64-x64.zip</a>	25.8 MB
 <a href="#">cmake-3.11.0.tar.gz</a>	7.58 MB
 <a href="#">cmake-3.11.0.tar.Z</a>	12.3 MB
 <a href="#">cmake-3.11.0.zip</a>	12.7 MB
 <a href="#">Source code (zip)</a>	
 <a href="#">Source code (tar.gz)</a>	

下载之后安装到合适的目录中。

#### 2) 安装 Visual Studio

本人此处采用的是 Visual Studio 2017 Community。注意的是，本人安装之后如下图所示出现一个 Microsoft Visual Studio 文件夹，一个 Microsoft Visual Studio 14.0 文件夹，在很多的教程中 CMake 的 Configure 中选择 Visual Studio 15 2017 或者 Visual Studio 15 2017 Win64，但是本人碰到的情况是如果这样操作就识别不出 VS，

只能选择 Visual Studio 14 2015 或者 Visual Studio 14 2015 Win64，也许是文件夹 Microsoft Visual Studio 14.0 的原因，因此此处建议后面选择的时候根据此处的具体情况进行选择。



### 3) 下载 opencv 及 opencv\_contrib 项目的源代码

Opencv 的下载链接是

<https://github.com/opencv/opencv/tree/4.1.0>

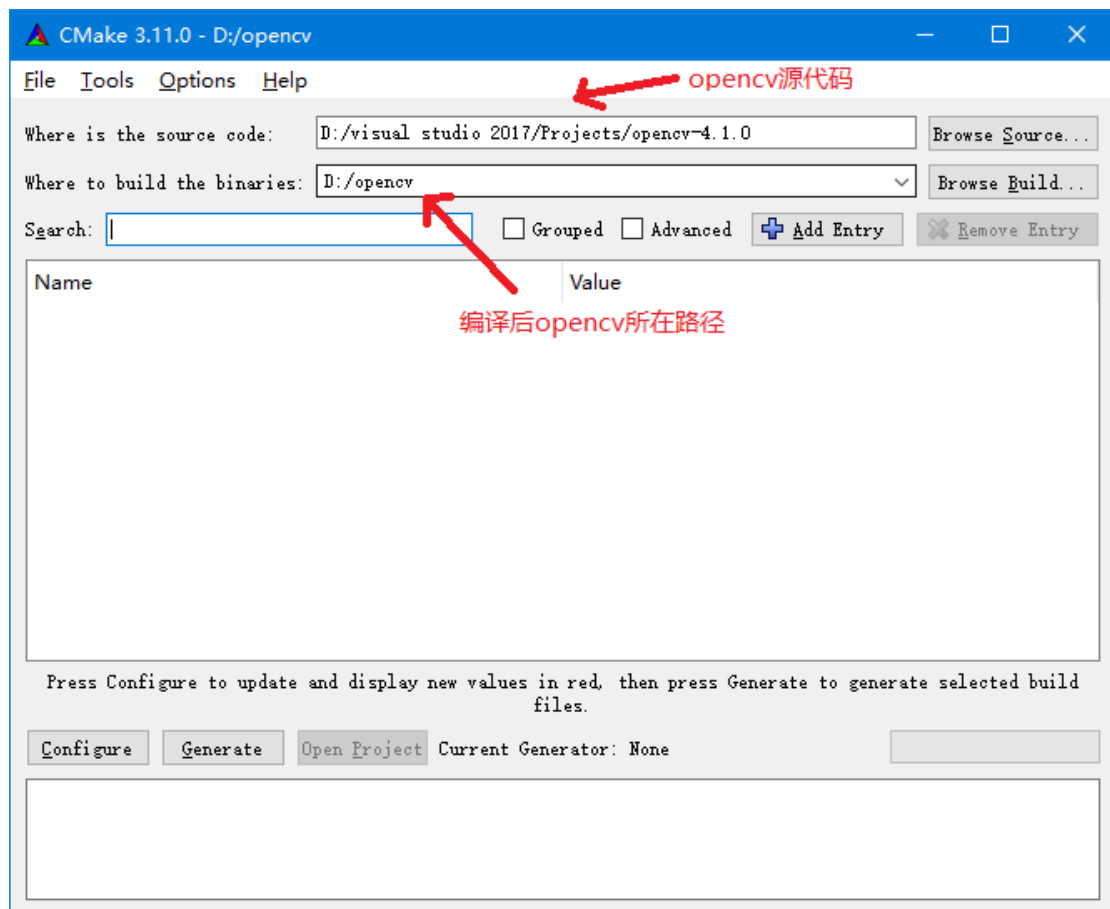
Opencv\_contrib 的下载链接是

[https://github.com/opencv/opencv\\_contrib/tree/4.1.0](https://github.com/opencv/opencv_contrib/tree/4.1.0)

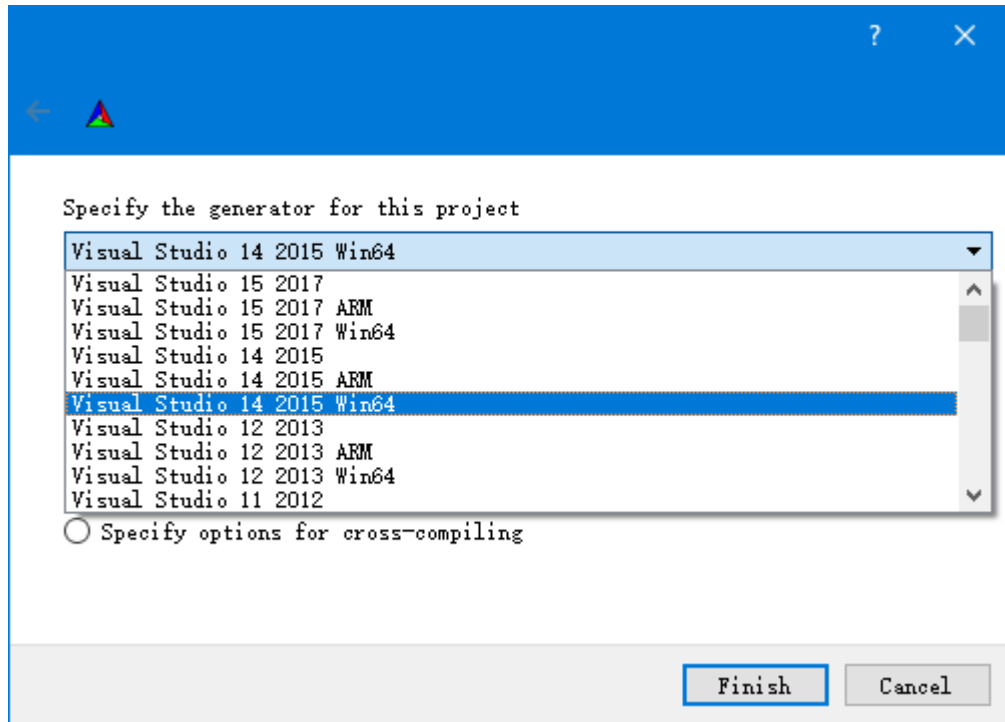
注意此处 opencv 的版本要和 opencv\_extra 的版本一致。

### 3、操作步骤

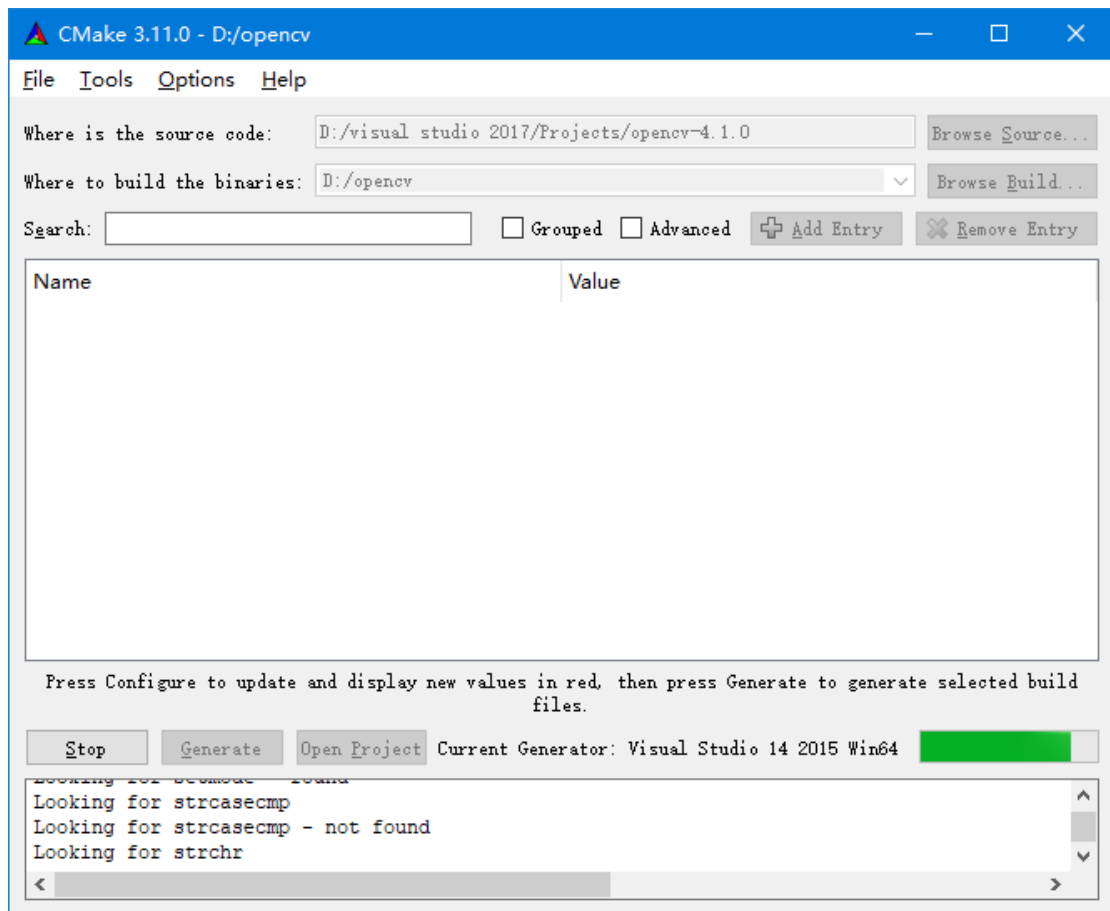
- 1) 将下载的 opencv 和 opencv\_extra 源代码解压到非中文路径下。
- 2) 打开 CMake。
- 3) 在 CMake 界面中导入 opencv 源代码所在路径及 opencv 编译后所在路径，如下图所示：



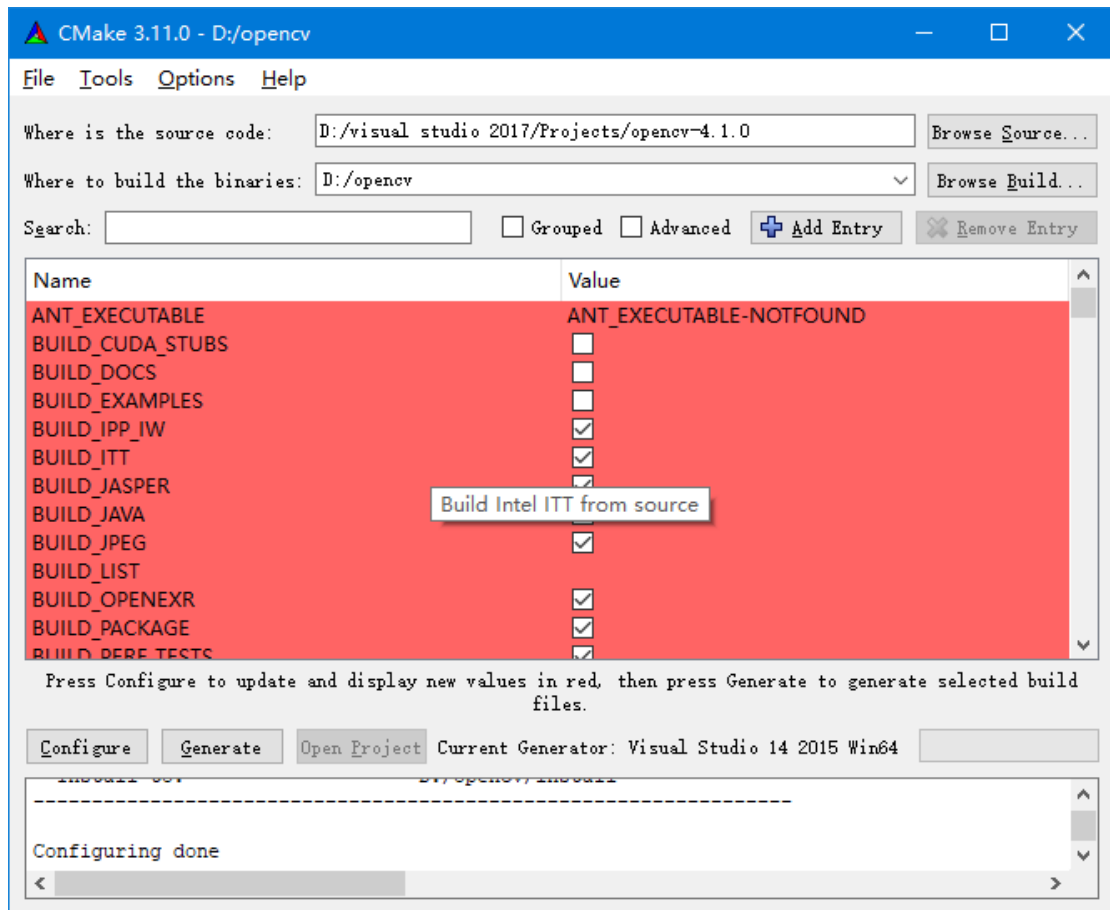
- 4) 点击 Configure 按钮，在弹出的界面中选择编译后的 opencv 库文件及动态链接库的版本, x86 还是 x64, 如果要编译 x86 版本则选择 Visual Studio 14 2015, 如果要编译 x64 版本则选择 Visual Studio 14 2015 Win64, 如下图所示：



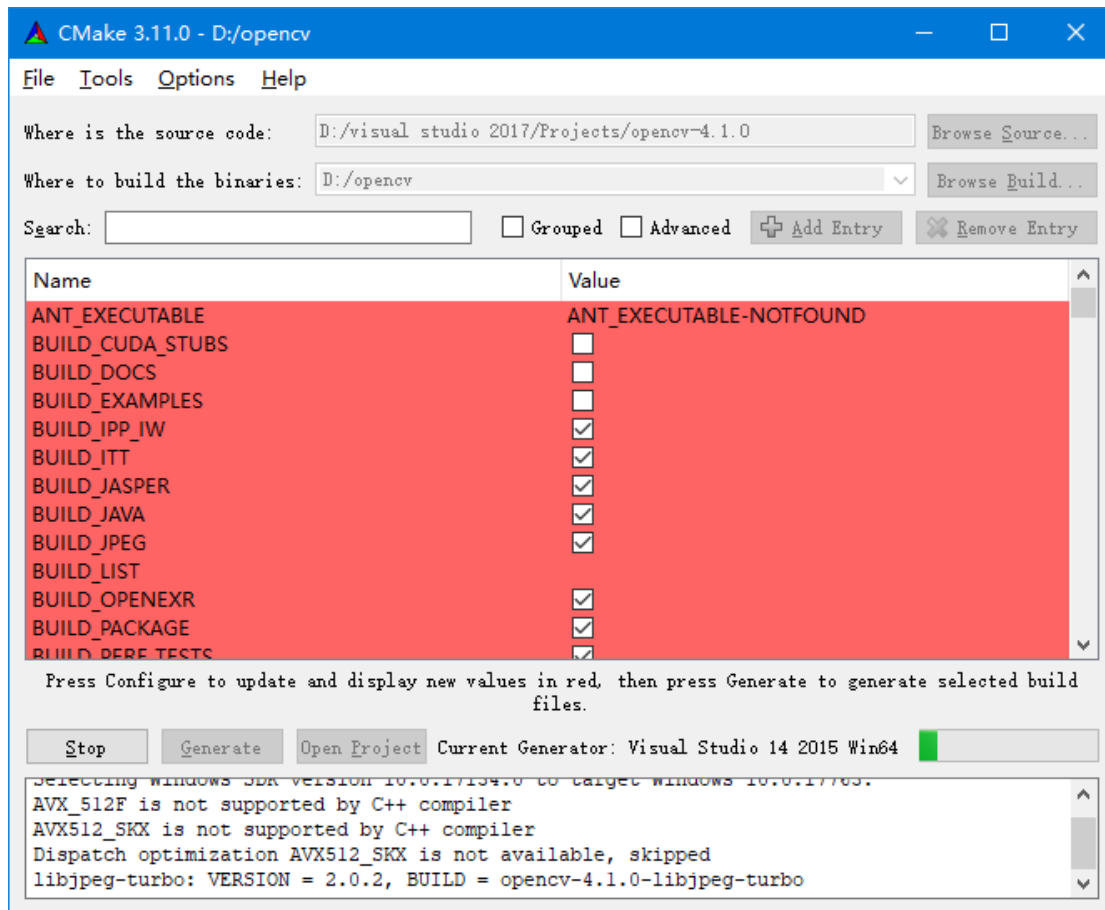
5) 点击 Finish，处理过程如下所示：



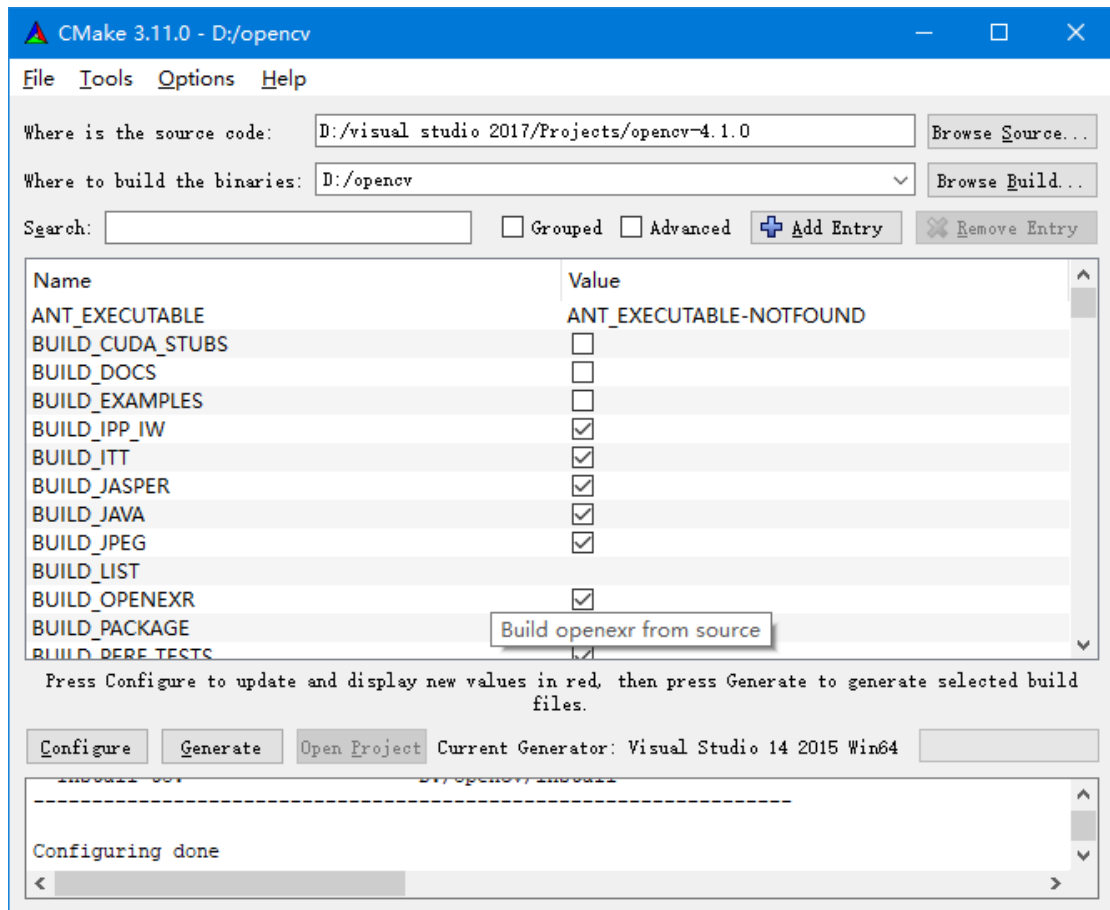
6) 上一步处理之后的结果如下图所示：



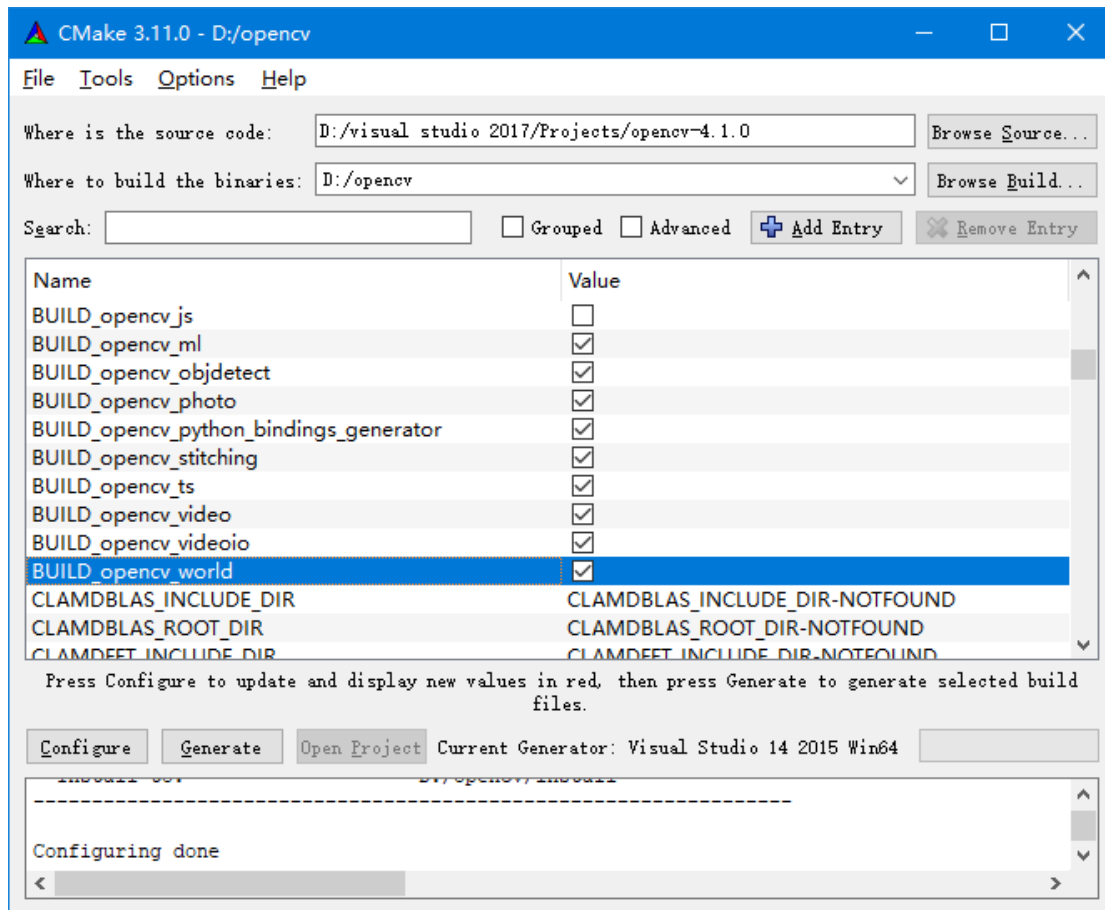
7) 再次点击 Configure 按钮，处理过程如下所示：



8) 上一步处理之后的结果如下图所示:

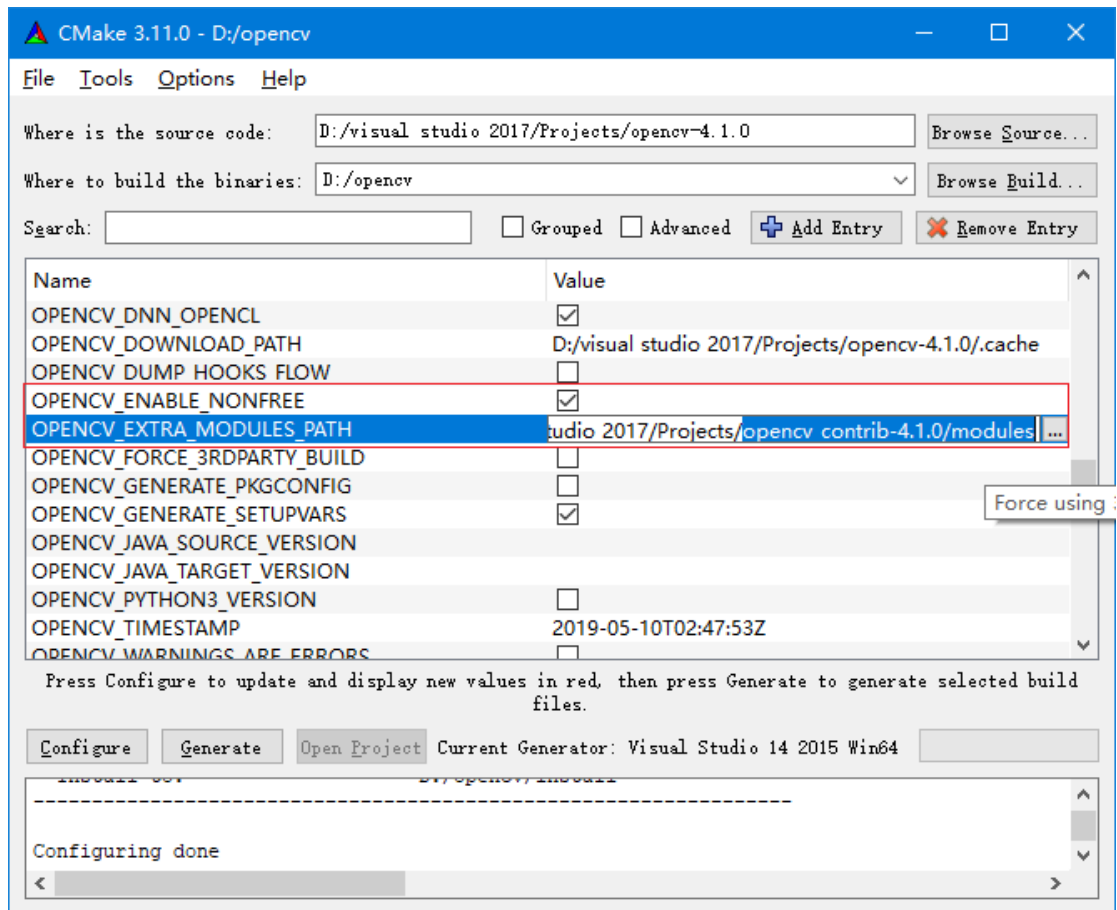


- 9) 将 BUILD\_opencv\_world 打勾，打勾之后生成的文件中所有的库文件及动态链接库都包括在一个标记为 opencv\_world410d.lib 和 opencv\_world410d.dll 文件中，上述是 debug 版本，如果是 release 版本则是 opencv\_world410.lib 和 opencv\_world410.dll。如下图所示：

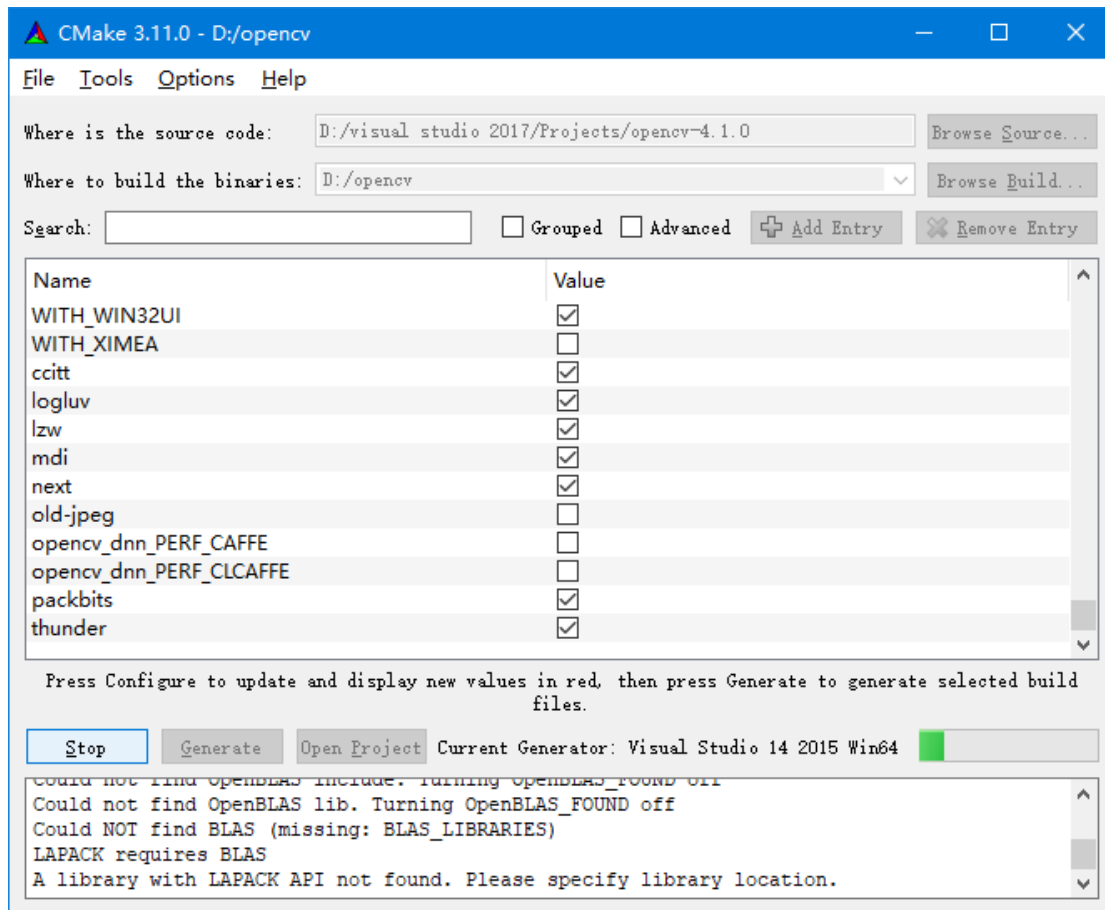


- 10) 将 `OPENCV_ENABLE_NONFREE` 打勾，打勾后就可以将一些非免费的算法编译进去，下面的 `OPENCV_EXTRA_MODULES_PATH` 选择 `opencv_contrib` 源代码中的 `modules` 文件夹所在的路径，如下图所示：

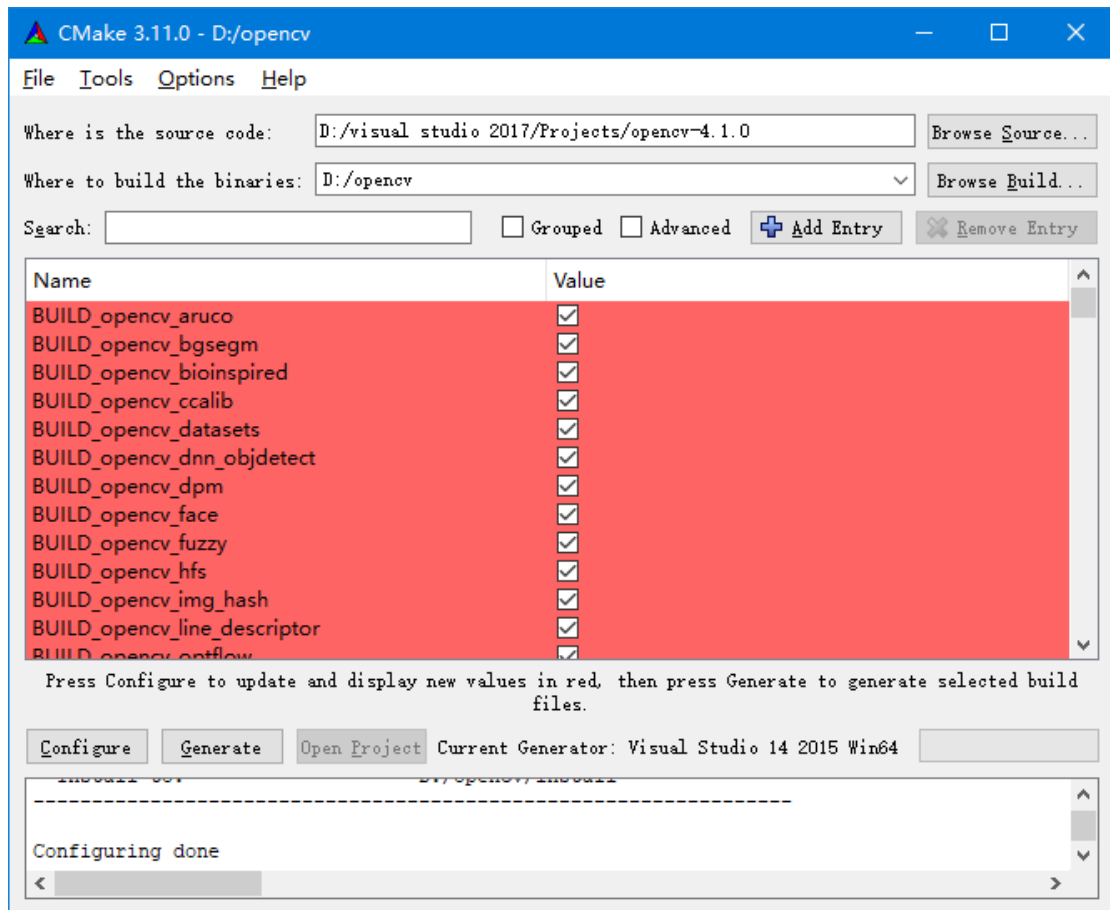




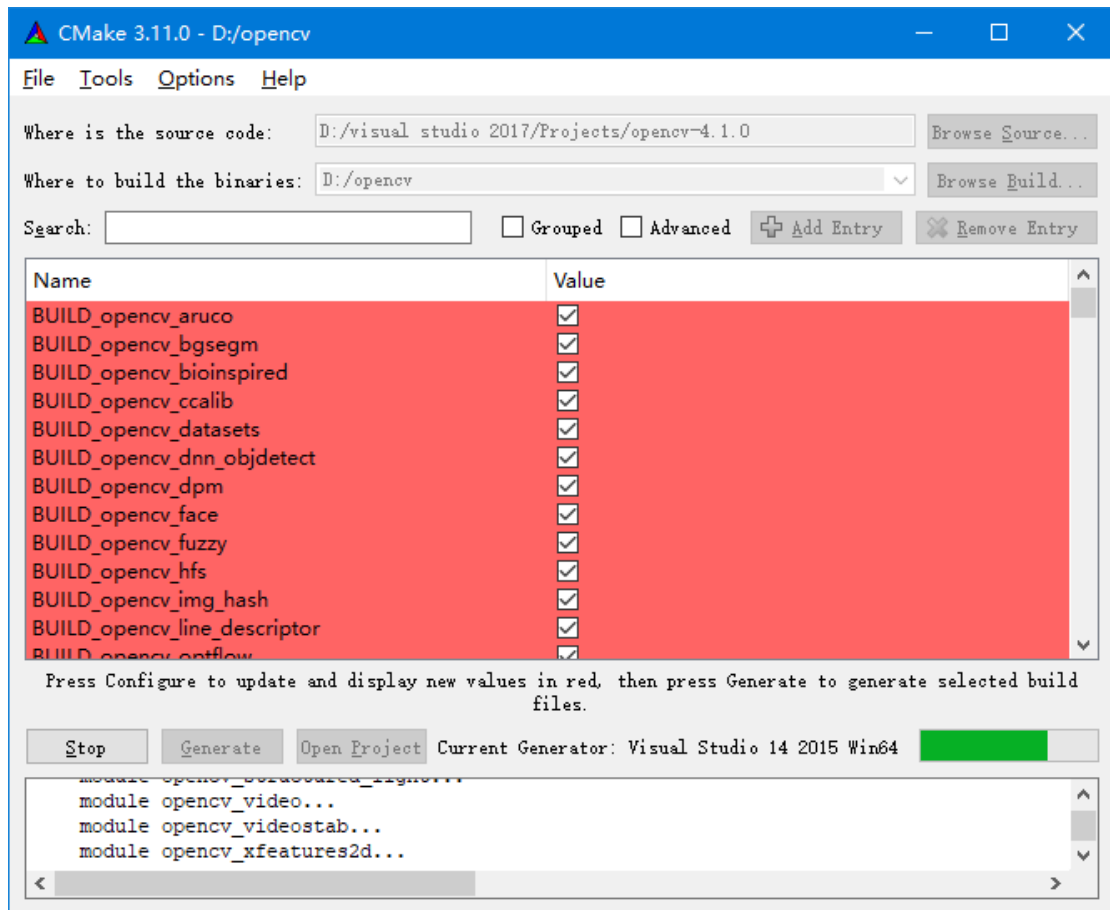
11) 上述步骤操作完成之后点击 Configure 按钮，处理过程如下图所示：



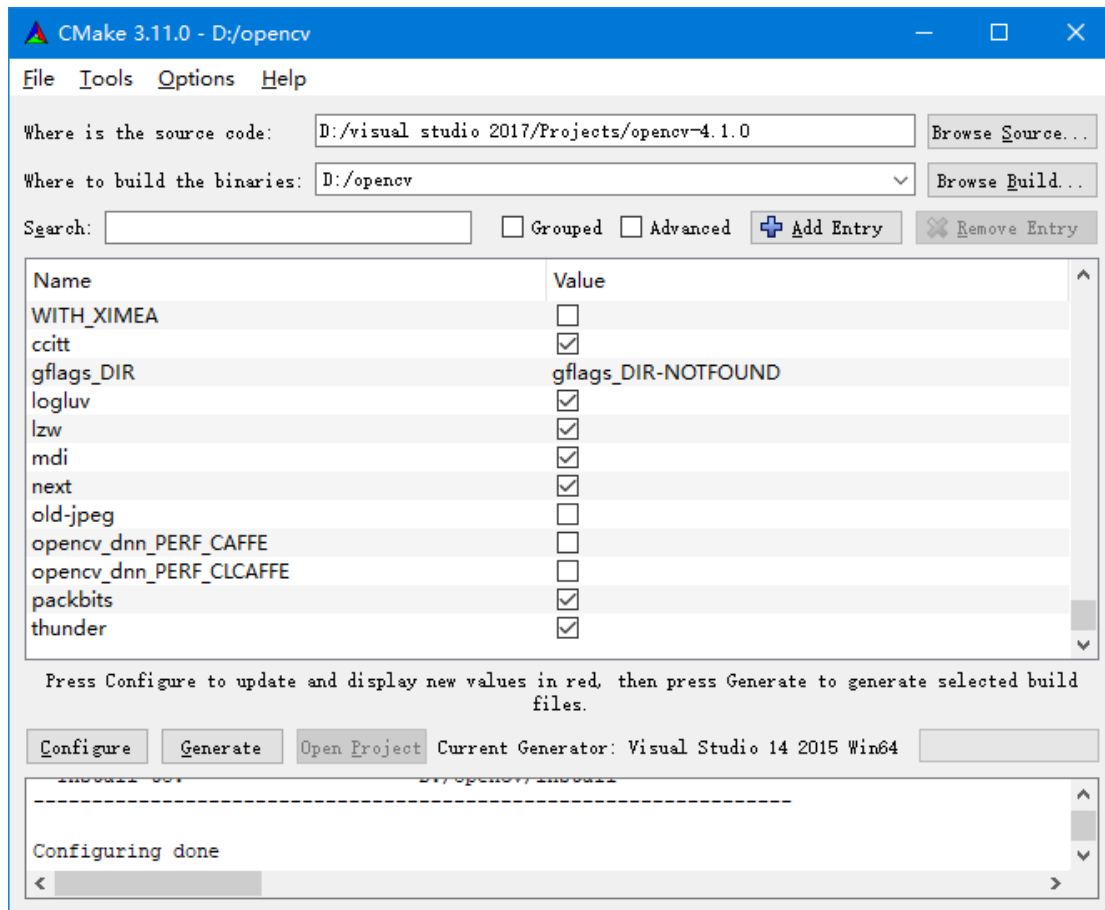
12) 处理之后的结果如下图所示:



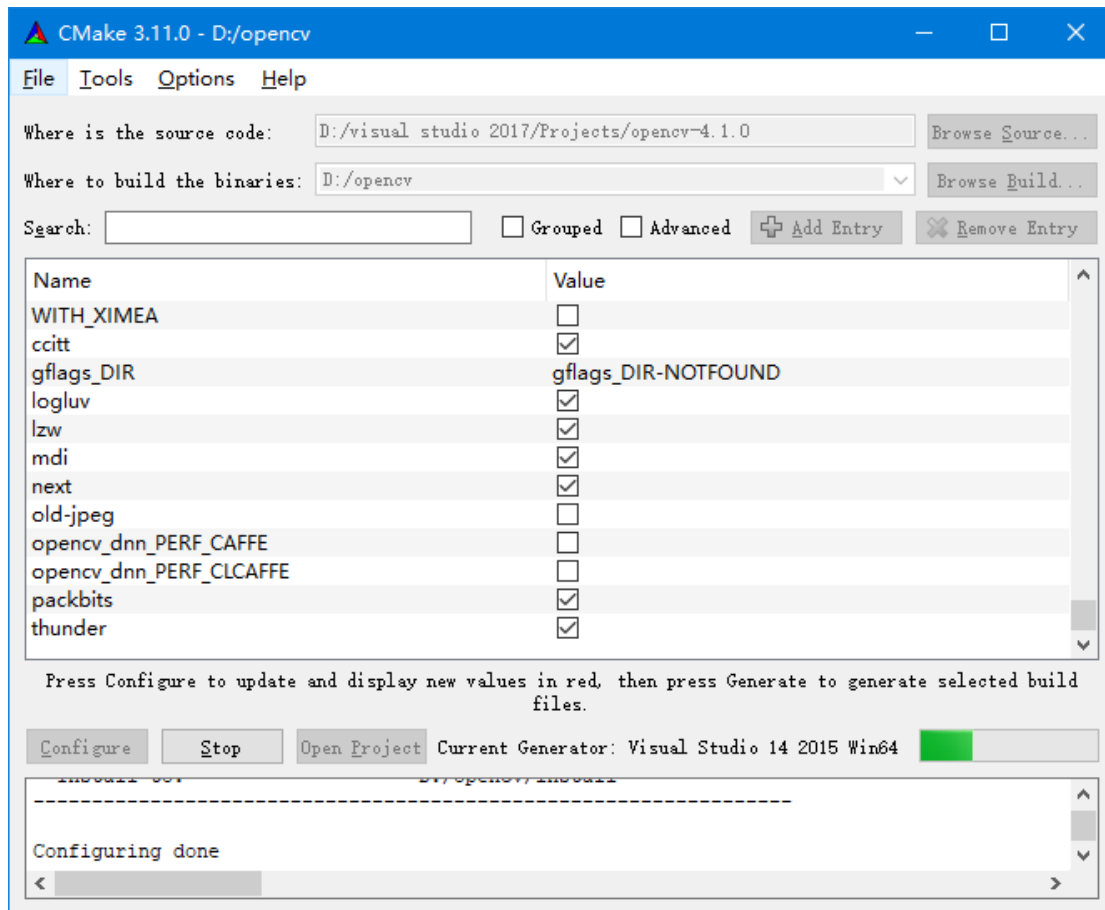
13) 上一步的操作完成之后，点击 Configure，处理过程如下图所示：



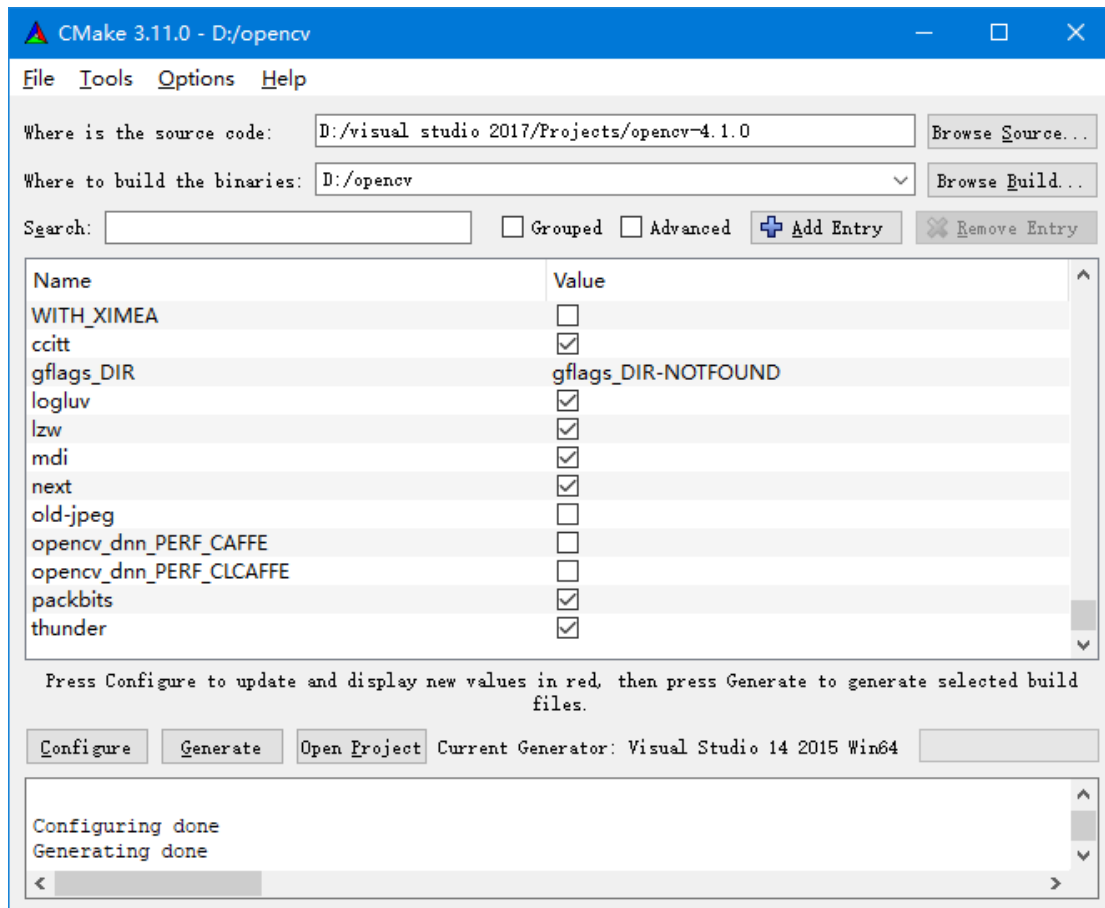
14) 处理完成之后的结果如下图所示:



15) 点击 **Generate** 按钮，生成解决方案，处理进度图如下所示：



16) 处理完毕如下图所示:



17) 很多教程在此步骤会直接点击 Open Project 或者通过 VS 打开项目进行编译，事实上这里有更便捷的办法，并且若生成 Python 库必须这样做（本人此步操作失败，依然没有生成 Python 库）。（也可以直接跳到 18）进行操作，依然可以生成所需要的文件）

在生成项目的文件夹（如 opencv）空白处，按住 shift 的同时点击鼠标右键，点击“在此处打开 powershell 窗口”，键入如下命令：

Release 模式：

```
cmake.exe --build . --config Release --target INSTALL
```

Debug 模式： debug 模式需要去掉 INSTALL\_PYTHON\_EXAMPLES、BUILD\_opencv\_python3 和 BUILD\_opencv\_python2 这三项的勾选，经测试若勾选会编译失败。

```
cmake.exe --build . --config Debug --target INSTALL
```

等待编译完成即可。

18) 点击 Open Project 按钮或者到 opencv 文件夹中找到 opencv.sln 并打开，于是就打开了生成的解决方案。（如果依照 17）的方法进行操作下面的步骤可以不进行操作）

19) 通过切换 Debug 和 Release 分别编译成 Debug 版本或者 Release 版本，后面的 x64 不能切换，需要在 CMake 中的 configure 中重新生成，如下图所示：



20) 当打开了生成的 opencv 项目，可以选择想要生成的库文件的 Debug 版本或 Release 版本

默认是 Debug 版本

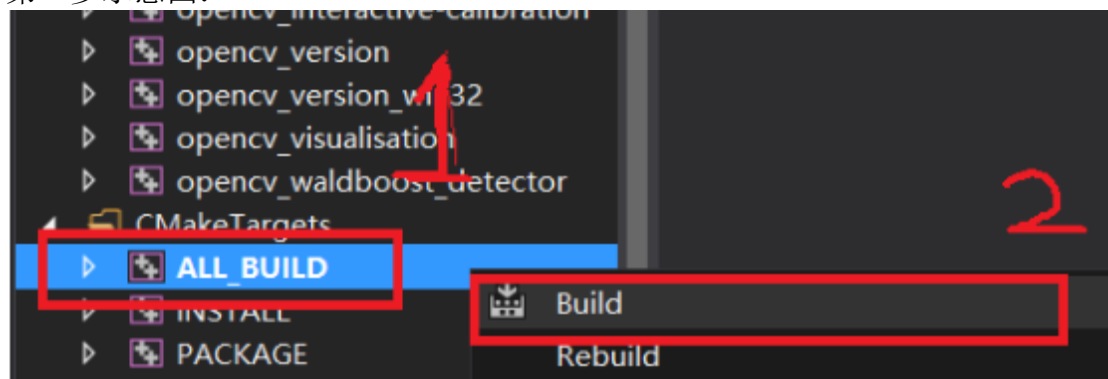
第一步：选择 CMakeTargets 中的 ALL\_BUILD，右键 ->生成，

第二步：生成结束如果没有失败的，再选择 INSTALL→右键→仅当前项目→只生成 INSTALL。

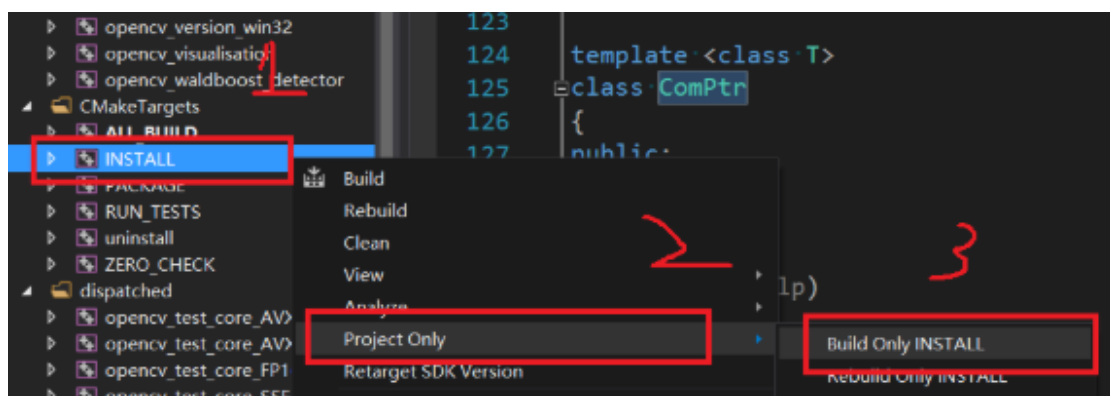
如果需要 release 版本，选择 release 版后重复第一步和第二步。

如下图所示：

第一步示意图：



第二步示意图



4、当在 VS2017 完成了生成 INSTALL，那么 OpenCV 的库就已经生成好了，在我们的 opencv 文件夹下的 install 文件夹中，包括了 bin、etc、include、x64、x86 文件夹和一些文件，这就和官网提供的 OpenCV 包里 build 文件夹下一样的结构了，只是没有 java 和 python 文件，将 install 拷贝到其他地方改个名，如 OpenCV410 就可以正常使用啦。



参考资料

<https://www.cnblogs.com/Brandon0807/p/10403872.html>

<https://www.cnblogs.com/lzhu/p/8198654.html>

[https://blog.csdn.net/Gordon\\_Wei/article/details/85775328](https://blog.csdn.net/Gordon_Wei/article/details/85775328)

<https://blog.csdn.net/howlclat/article/details/78783701>

<https://blog.csdn.net/LOVELESSYI/article/details/79946859>

[https://blog.csdn.net/poem\\_qianmo/article/details/21974023](https://blog.csdn.net/poem_qianmo/article/details/21974023)

[https://blog.csdn.net/qq\\_36088602/article/details/83754403](https://blog.csdn.net/qq_36088602/article/details/83754403)

<https://blog.csdn.net/qq385105501/article/details/73770826>

<https://blog.csdn.net/suswulongyuan/article/details/86544180>

<https://blog.csdn.net/u012900686/article/details/69442406?locationNum=13&fps=1>