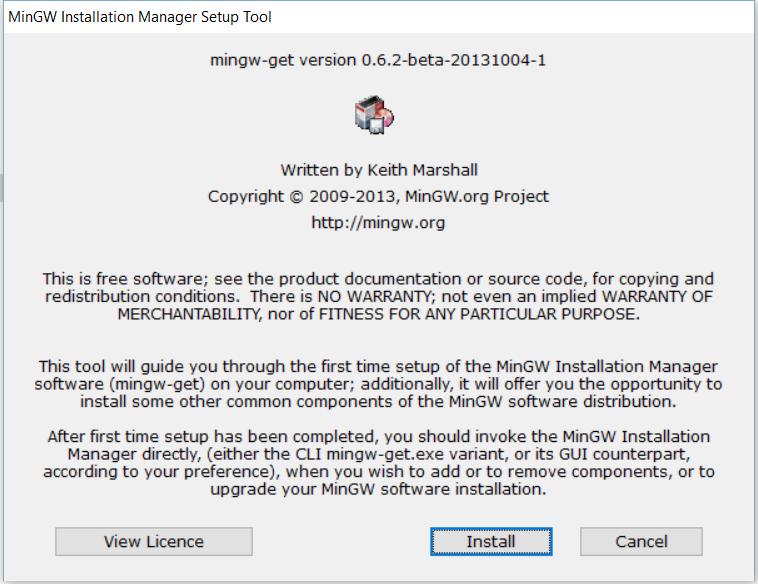
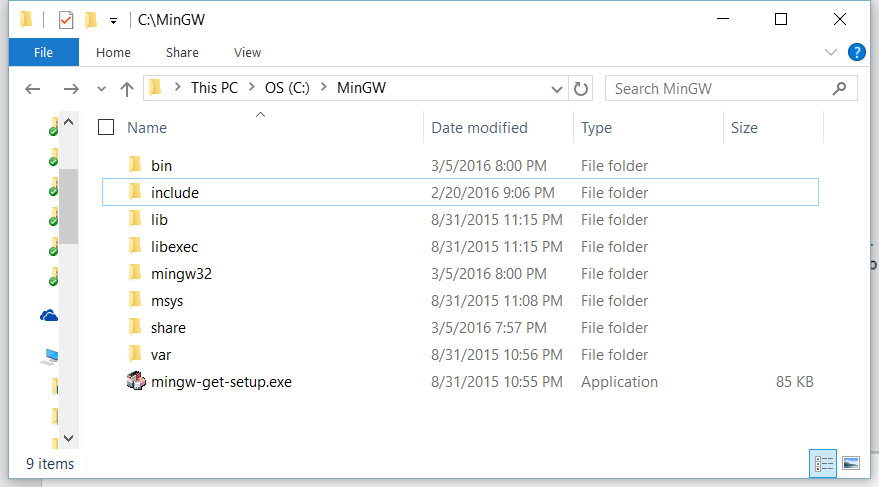
**Projection Segment Project**

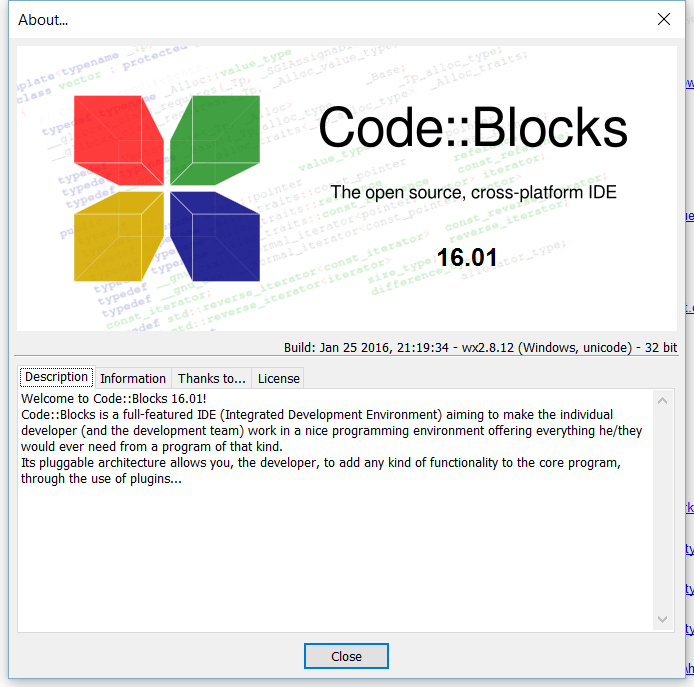
**Developing Environment Configuration**

Hui Xie April 24 2016

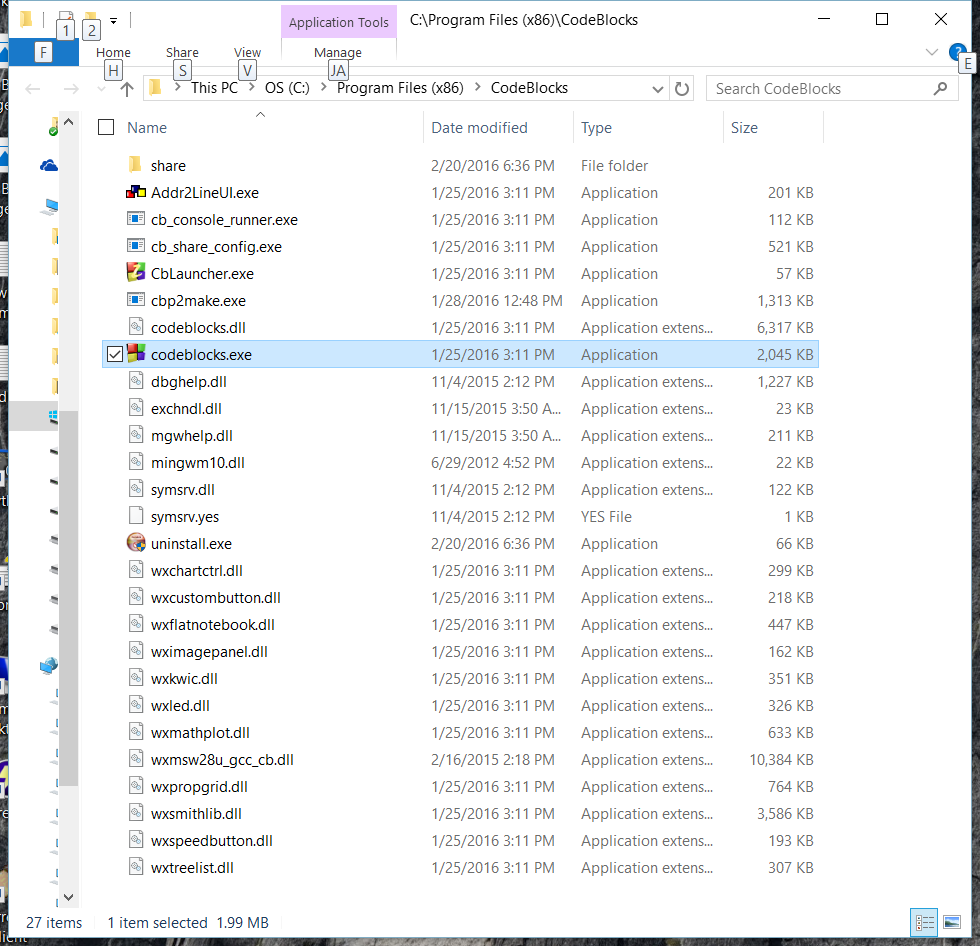
1. **install MinGW, which supplies gcc for Windows.**

After installation, it looks like:



1. ** Install Code::Blocks 16.01**

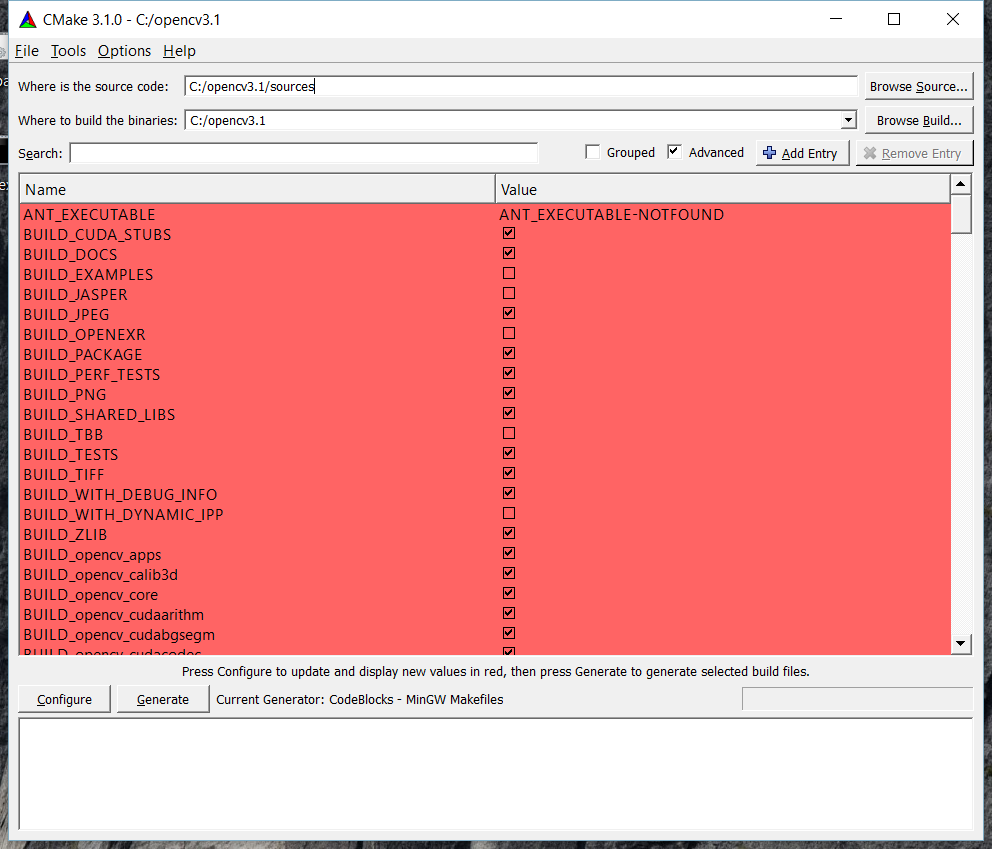
After Installation, it looks like:

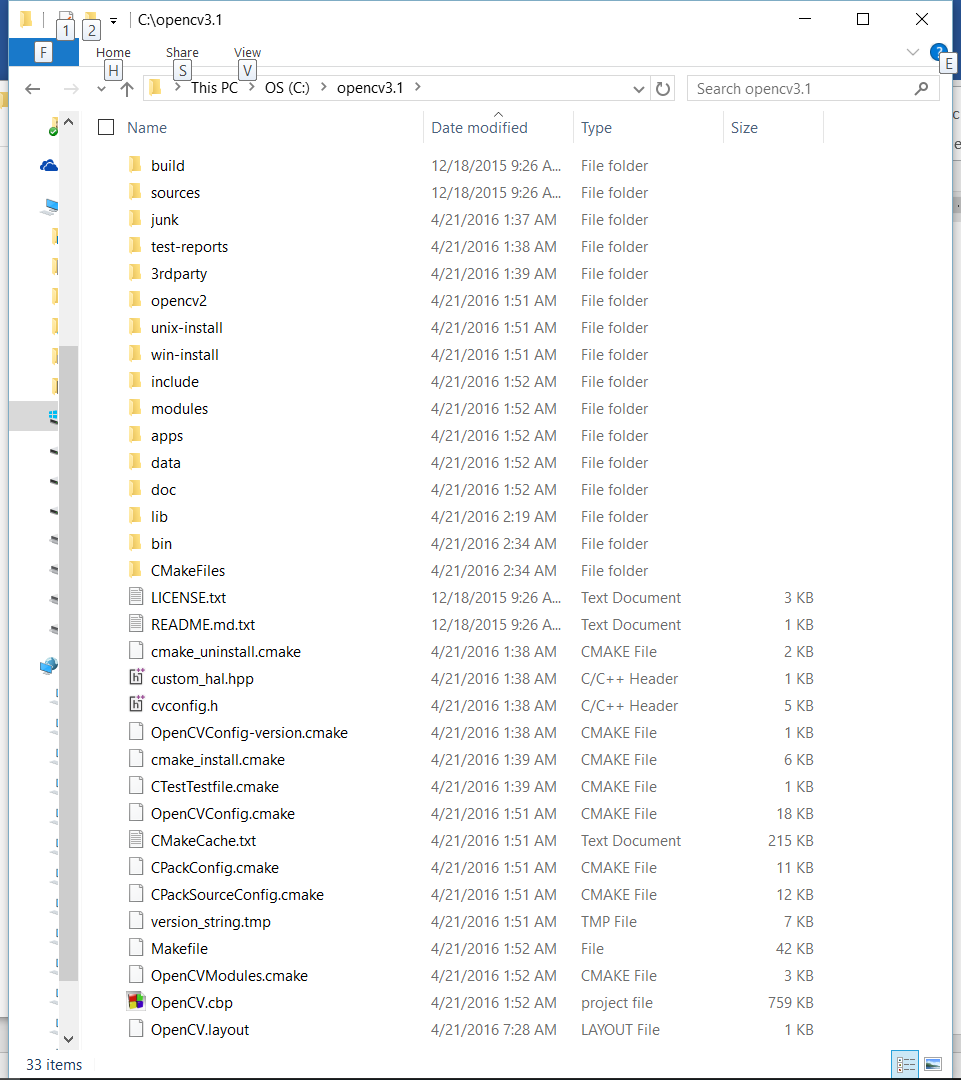


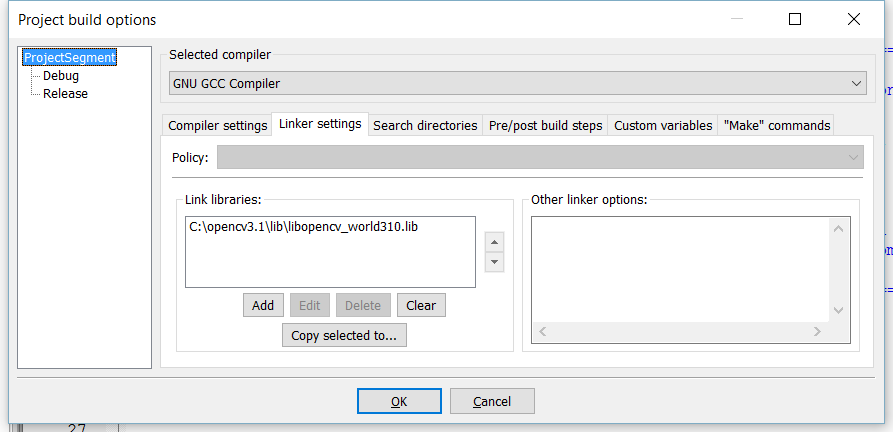
1. **Install OpenCV3.1 for Windows, and you need to use CMake3.1 and Code::Blockes to recompile openCV’s source code, and get the DLL for gcc.**

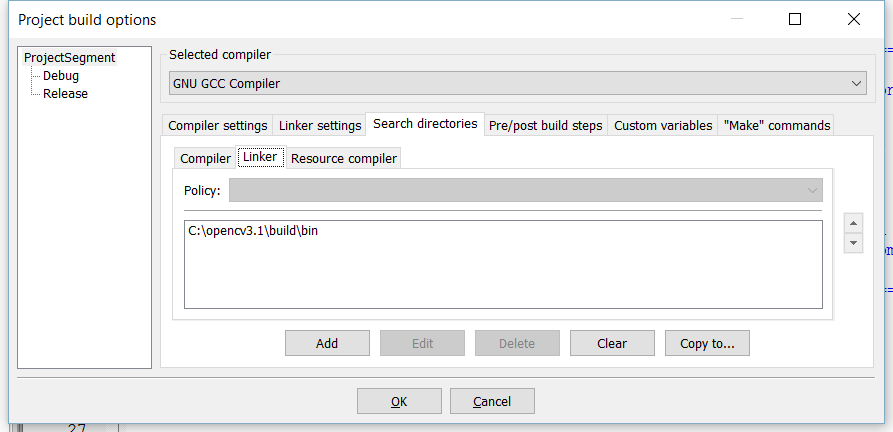
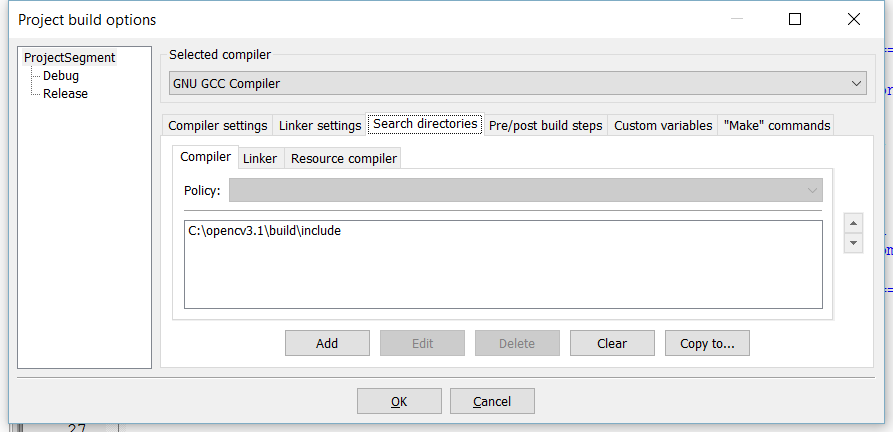
**I directly put the libopencv\_world310.dll which I just compile from above in the directory with our projectSegment.exe file, so it is unnecessary to config opencv path;**

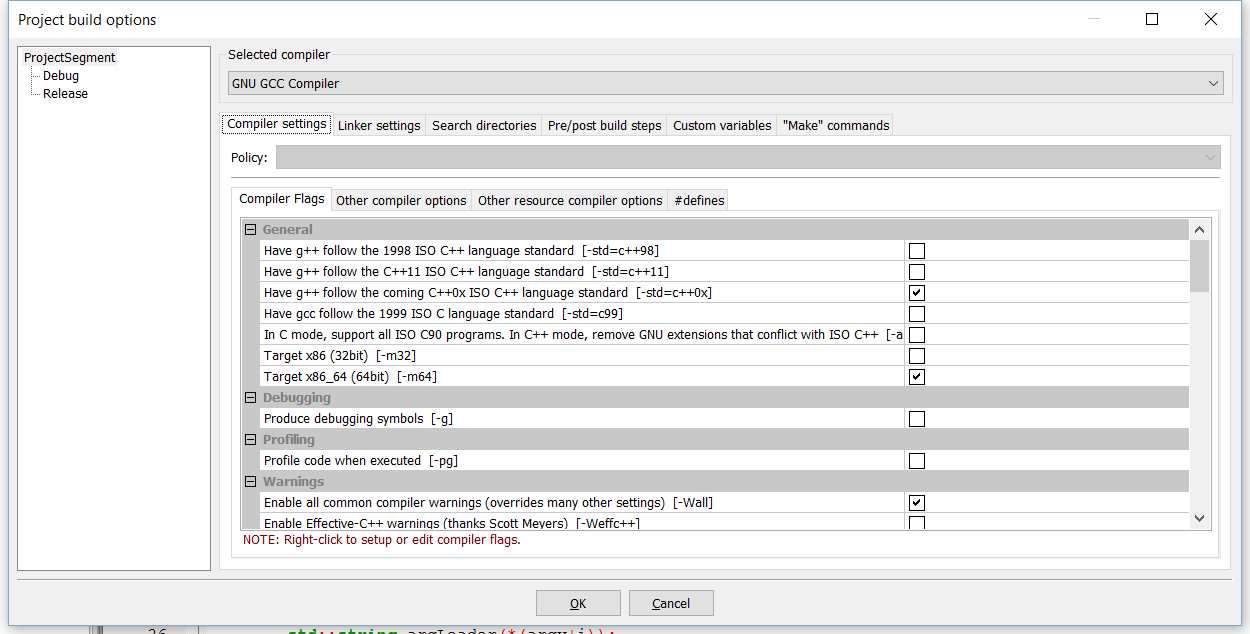
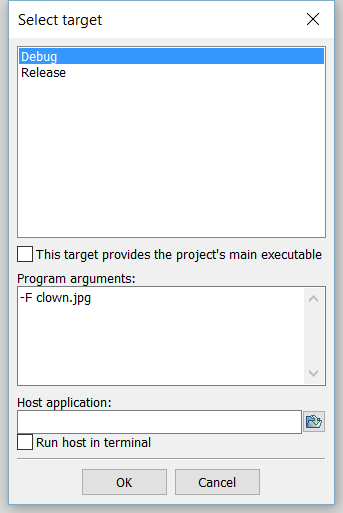
**If you don’t have Cmake3.1 in your PC, your need to install one.**





1. **Code::Blocks’s project config:**



The right Select Target is for debug config for project/set project parameter:

1. **Gcc compile script if using gcc CLI**

-------------- Build: Release in ProjectSegment (compiler: GNU GCC Compiler)---------------

g++.exe -Wall -std=c++0x -m64 -fexceptions -O2 -std=c++11 -m32 -IC:\opencv3.1\build\include -c "E:\VC Projects\ProjectSegment\ImageAnalyzer.cpp" -o obj\Release\ImageAnalyzer.o

g++.exe -Wall -std=c++0x -m64 -fexceptions -O2 -std=c++11 -m32 -IC:\opencv3.1\build\include -c "E:\VC Projects\ProjectSegment\LabelManager.cpp" -o obj\Release\LabelManager.o

g++.exe -Wall -std=c++0x -m64 -fexceptions -O2 -std=c++11 -m32 -IC:\opencv3.1\build\include -c "E:\VC Projects\ProjectSegment\main.cpp" -o obj\Release\main.o

g++.exe -LC:\opencv3.1\build\bin -o bin\Release\ProjectSegment.exe obj\Release\ImageAnalyzer.o obj\Release\LabelManager.o obj\Release\main.o -m64 -s -m32 C:\opencv3.1\lib\libopencv\_world310.lib

Output file is bin\Release\ProjectSegment.exe with size 68.50 KB

1. **Now you can compile our ProjectSegment.cbp project file. You also can run it in the Linux OS.**