

Installation Cheat Sheet 1 - OpenCV 3 and C++

Using Windows 10 + Visual Studio 2015 (Community Edition) + precompiled binaries

[Click here to go to the YouTube video for this Cheat Sheet](#)

[GitHub page with all Cheat Sheets and code](#)

[If you found this Cheat Sheet helpful please consider supporting me on Patreon](#)

Note: **Blue** indicates something that will change depending on your version of OpenCV and/or Visual Studio

1) Download and install Visual Studio **2015** Community Edition (yes, it's free, and choosing all default options will work fine)

2) Download the latest version of OpenCV, for example OpenCV **3.2.0** as of when this is being written

3a) Make a folder "C:\OpenCV-**X.X.X**" for your version of OpenCV, ex. "C:\OpenCV-**3.2.0**"

3b) Double-click on the completed OpenCV download executable, then set "Extract to:" to your "C:\OpenCV-**X.X.X**" directory

4a) Add the **bin** directory for your version of OpenCV and Visual Studio to the operating system PATH.

4b) Note: In the OpenCV directories:

"vc**14**" => Visual Studio **2015**

For example if you are using OpenCV **3.2.0** and Visual Studio **2015** add the following to your PATH:

C:\OpenCV-**3.2.0**\opencv\build\x64\vc**14**\bin

4c) Pull up a Command Prompt and verify the bin directory is now in PATH, then reboot

5) From my [MicrocontrollersAndMore GitHub](#) page decide which example you are going to use:

CannyStill.cpp (uses a still image)

CannyWebcam.cpp (uses a webcam)

RedBallTracker.cpp (tracks a red ball, uses a webcam)

If you are going through this for the first time I suggest CannyStill.cpp

6a) Start Visual Studio, choose File -> New -> Project

6b) Choose Visual C++, Empty Project, name as you prefer, ex "CannyStill1", set preferred location, uncheck "Create directory for solution" and "Add to source control", choose OK

7a) Right click in Solution Explorer, choose Add -> New Item

7b) Choose "C++ File", name the C++ file as preferred, ex. "CannyStill1.cpp", choose "Add"

7c) Copy/paste the entire code from your chosen example into the .cpp file

(At this point Visual Studio will underline many of the lines of code with red because we have not yet informed Visual Studio as to the location of OpenCV, subsequent steps will resolve this)

8) If you are using an example with a still image (i.e. CannyStill.cpp), copy any JPEG image into the project directory and rename it "image.jpg". You can use the "image.jpg" from my [MicrocontrollersAndMore GitHub](#) page if you would like to see the same results as in the video (if you are using a webcam example then this step does not apply).

9) In the Visual Studio toolbar, verify that "**Solution Configurations**" is set to "**Debug**", then change "**Solution Platforms**" to "**x64**" (make sure not to skip this step, or 32-bit vs 64-bit errors will be encountered)

10) In VS go to:

Project -> (project name) Properties -> Configuration Properties -> VC++ Directories -> Include Directories
add the **include** directory for your version of OpenCV, ex "C:\OpenCV-3.2.0\opencv\build\include"

11) In VS go to:

Project -> Properties -> Configuration Properties -> VC++ Directories -> Library Directories:
add the **library** directory for your version of OpenCV, ex "C:\OpenCV-3.2.0\opencv\build\x64\vc14\lib"

12) If you currently *do not* have Windows 10 configured to allow viewing / editing of file extensions, go to:
right-click on Start -> Control Panel -> View by: Large icons -> File Explorer Options -> View tab -> uncheck
"Hide extensions for known file types" (if you already have viewing file extensions enabled then skip this step).

13) In File Explorer (not within Visual Studio), navigate to the **lib** directory, ex

C:\OpenCV-3.2.0\opencv\build\x64\vc14\lib

In the lib directory you will find the debug libs (ending with a 'd'), for example if you are using OpenCV 3.2.0 there is only one debug lib:

opencv_world320d.lib

Note the 320 is referring to OpenCV 3.2.0 and this will vary by OpenCV version, also which libraries are present may vary with OpenCV version.

Copy/paste the name of the debug lib(s) in the lib directory into the following location in Visual Studio:

Project -> Properties -> Configuration Properties -> Linker -> Input -> Additional Dependencies

14) Run the program, either without debugging (choose Debug, then the hollow green arrow, or press Ctrl+F5) or with debugging (solid green arrow or press F5)

15) Once you get the **debug** build working, if it is too slow and you would like to try the **release** build, at the top of Visual Studio change "**Solution Configurations**" to "**Release**", then verify "**Solution Platforms**" is still "**x64**" (if not, change it to **x64**), then repeat steps 10 through 13, making the applicable changes for a **release** configuration.

As of OpenCV 3.1.0 or later, when using the precompiled binaries as described in this document, the Include directory (step 10) and the Library directory (step 11) are the same for **debug** or **release**, i.e. use the same Include and Library directory paths as above if using OpenCV 3.1.0 or later. This may change in a future version of

OpenCV so be sure to check this if you are using a version of OpenCV after **3.2.0**.

For the .lib files themselves (step 13), when configuring for a **release** build, make sure to use the .lib file(s) without the 'd' just before the .lib, i.e. if you are using OpenCV **3.2.0**, use the following .lib:

opencv_world**320**.lib

Once you have the **release** build configured, you can switch back and forth between the **debug** build and the **release** build simply by changing "Solution Configurations" back and forth between "Debug" and "Release".