

# How to install and run OpenCV 3.0.0 using Microsoft Visual Studio 2015

## Visual Studio

You can use Visual Studio 2015 Express for Desktop for free, get it from

<https://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx>

You can also use “Community”, which has more stuff. Or, get the Visual Studio 2015 Enterprise from the Microsoft “Dreamspark” program through CSM.

For Community or Enterprise, when you go to create a C++ project, it makes you install C++ a couple of additional things.

## OpenCV

Get OpenCV. I put it in `C:\OpenCV-3.0.0`. It has two folders:

`Build` – this contains the includes; also the prebuilt libs and dlls that work with a particular version of Visual Studio. If you are using that version of Visual Studio, you don’t need to compile OpenCV.

`Sources` – source code for OpenCV; also contains sample code and data.

## CMake

If you need to work with another version of Visual Studio, or need Qt or some other package like Vtk, continue on. This [video](#) is helpful.

Get Cmake. In in `C:\OpenCV-3.0.0`, make a folder called “mybuild”.

Start Cmake-gui. For “where is the source code”, put `C:/OpenCV-3.0.0/sources`. For where to build binaries, put `C:/OpenCV-3.0.0/mybuild`.

Click “configure”. For what “generator” to use, you can use “Visual Studio 14 2015” (this is for 32 bit) or “Visual Studio 14 2015 Win64” (this is for 64 bit). I’m guessing the latter is better/faster? I had to uncheck “With VTK”; I don’t know why this was checked in the first place. Click “configure” again until all entries are white, not red.

Click “generate”. This will create a Visual Studio “sln” file in `C:/OpenCV-3.0.0/mybuild`. Close cmake.

## Compiling OpenCV

In C:/OpenCV-3.0.0/mybuild, double click on "OpenCV.sln". Now, if you have chosen the generator "Visual Studio 14 2015 Win64" in cmake, your visual studio project should be set to "x64" as the configuration at the top. It is probably set to "Debug" as well; that's ok.

After it is done "parsing", go to "ALL\_BUILD" in the "Solutions Explorer" window. Right click on that and choose "Build". This will create all the "lib" and "dll" files for debugging. On my Surface (Windows 8.1) I get a popup message saying "Microsoft® C/C++ Optimizing Compiler has stopped working". I close that window and it still seems to compile ok. There are two "failures" but it seems to be a problem with building some test program.

Change "Debug" to "Release", and recompile (by right clicking on "ALL\_BUILD"). This will create all the "lib" and "dll" files for the release versions.

I then right clicked on "INSTALL" under "CMakeTargets" for both Debug and Release. This combines all the lib and dll files into a single "lib" and a single "bin" folder. Mine went into: C:\OpenCV-3.0.0\mybuild\install\x64\vc14\lib and C:\OpenCV-3.0.0\mybuild\install\x64\vc14\bin.

I then moved C:\OpenCV-3.0.0\mybuild\install to C:\OpenCV-3.0.0\install.

I then added C:\OpenCV-3.0.0\install\x64\vc14\bin to the system PATH.

## Other issues

### PDB files

These are "symbol files" that help in debugging.

- Cmake creates a bunch of pdb files in the "bin" directories, but INSTALL doesn't copy them over to the new bin directory. But I don't think you need these? It looks like you can step through OpenCV code just fine without them.
- If you run your program and it runs ok, but you get lots of messages in the "output" window that it can't find "pdb" files – you can ignore this if you want because it is looking for windows pdb files. However, if you don't like this you can do this:

Go to Tools->Options->Debugging->Symbols and check the box for "Microsoft Symbol Servers".

Then it will download those symbol files when you debug your program. This may be slow the first time, but after that it won't need to download them.

You can always run your program faster by doing Debug->Start Without Debugging, instead of Debug->Start Debugging.