Installation (Windows, MATLAB, OpenCV 3)

Amro edited this page on Mar 19 · 4 revisions

> **IMPORTANT**: mexopencv is developed against specific OpenCV releases, therefore you must be using the correct `opencv` & `opencv_contrib` versions. The latest mexopencv is only compatible with OpenCV 3.2.0.
>
> ```
> - DO NOT use the dev-version (master branch) of opencv/opencv_contrib
> + version 3.2.0 is expected!
> ```

1) OpenCV

The following instructions are meant to compile OpenCV specifically for use in `mexopencv`. OpenCV functionality that is not currently exposed or used by `mexopencv` are disabled (CUDA acceleration, OpenCL with `UMat`, etc...). If you intend to compile OpenCV for general use in your own C++ code, we recommend following the instructions in the OpenCV documentation.

Download OpenCV

First thing is to obtain OpenCV v3.2.0 (the latest stable version). You have a number of options to choose from:

- download the official installer from SourceForge. This package contains both the source code and prebuilt binaries for Windows (x86 and x64 architectures, and select versions of Visual Studio). You should get the file `opencv-3.2.0-*.exe`. We need the contents of the `sources` folder from this package.

  > **Note:** this package doesn't include the extra `opencv_contrib` modules, so you should obtain it using one of the other options listed below.

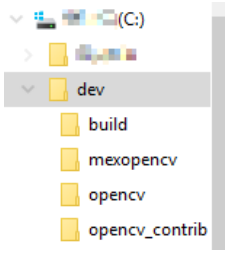
- [recommended] download the source tarballs directly from GitHub. This includes downloading these two archives:
  - the `opencv` sources (tag `3.2.0`): https://github.com/opencv/opencv/archive/3.2.0.zip
  - the `opencv_contrib` sources (tag `3.2.0`): https://github.com/opencv/opencv_contrib/archive/3.2.0.zip
- clone the Git repositories from GitHub. Again there are two parts:
  - the main `opencv` repository: `git clone https://github.com/opencv/opencv.git`
  - the extra `opencv_contrib` repository: `git clone https://github.com/opencv/opencv_contrib.git`

  Once you clone the repositories, you need to switch to the stable `3.2.0` tag in both (`git checkout tags/3.2.0`).

> **Note:** You'll need to have access to `git` either from the command-line using msysgit or Cygwin, or using a graphical user interface like GitHub Desktop, SourceTree, or SmartGit.

Depending on which option you chose above, extract or move the files to some desired location. For example, set it so that you end up with these files in the following paths:

- `C:\dev\opencv\README.md`
- `C:\dev\opencv_contrib\README.md`



Configure OpenCV

This step requires CMake and a supported C++ compiler (Visual Studio 2015 is used in the instructions below, but you could use any other version, VS2010 at the least).

> **Note:** Microsoft offers a full and free version of Visual Studio for personal use: Visual Studio Community 2015 (equivalent in functionality to the Professional Edition).

We begin by generating a VS solution to build the sources. Start the `cmake-gui.exe` tool, and follow these steps:

1. set the source folder as `C:/dev/opencv`
2. set the destination folder as: `C:/dev/build`
3. press *Configure*, and choose `Visual Studio 14 2015 Win64` as compiler
4. under "BUILD" group, deselect the following:

- `BUILD_DOCS`, `BUILD_EXAMPLES`, `BUILD_PACKAGE`, `BUILD_PERF_TESTS`, `BUILD_TESTS`
- `BUILD_opencv_apps`, `BUILD_opencv_cuda*`, `BUILD_opencv_cudev`, `BUILD_opencv_java`, `BUILD_opencv_python*`, `BUILD_opencv_ts`, `BUILD_opencv_viz`, `BUILD_opencv_world`

5. under "CPACK" group, deselect all
6. under "OPENCV" group:

- set `OPENCV_EXTRA_MODULES_PATH` to `C:/dev/opencv_contrib/modules`
- enable `OPENCV_ENABLE_NONFREE`

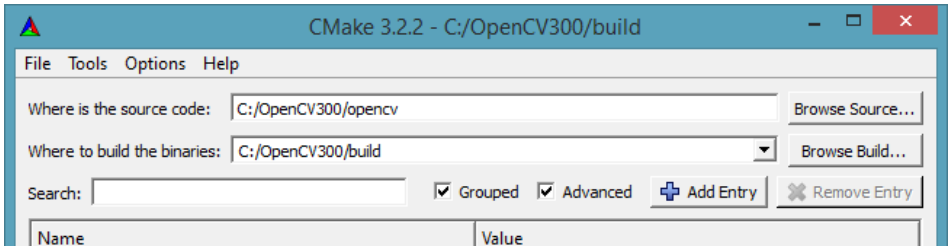
7. under "WITH" group, deselect:

- `WITH_CUDA`, `WITH_CUFFT`, `WITH_EIGEN`, `WITH_MATLAB`, `WITH_VTK`

8. press *Configure* again
9. under the newly added "BUILD" group (displayed in red), deselect the following modules if possible:

- `BUILD_opencv_contrib_world`, `BUILD_opencv_cvv`, `BUILD_opencv_freetype`, `BUILD_opencv_hdf`, `BUILD_opencv_matlab`, `BUILD_opencv_sfm`

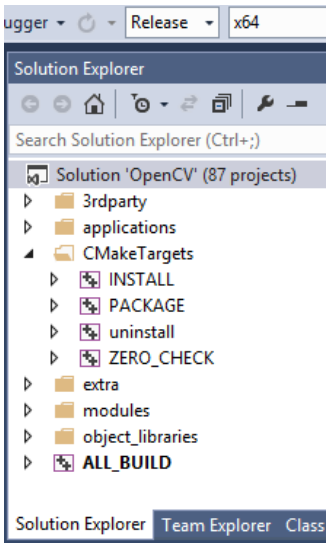
10. press *Configure* yet another time
11. press *Generate*
12. close CMake



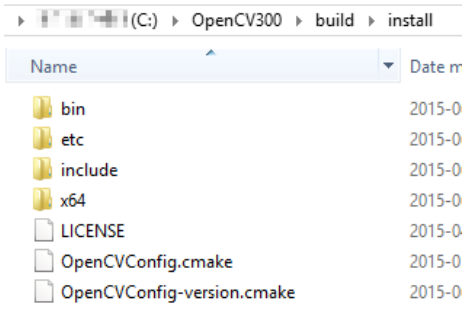
Compile OpenCV

Next step is to actually build OpenCV. Open the created solution file in Visual Studio: `C:\dev\build\OpenCV.sln`, and do the following:

1. switch configuration to "Release" mode (or "Debug" if you like)
2. build the solution ("ALL_BUILD" target), this will take a few minutes
3. select the "INSTALL" project in the solution explorer, and build it
4. close Visual Studio

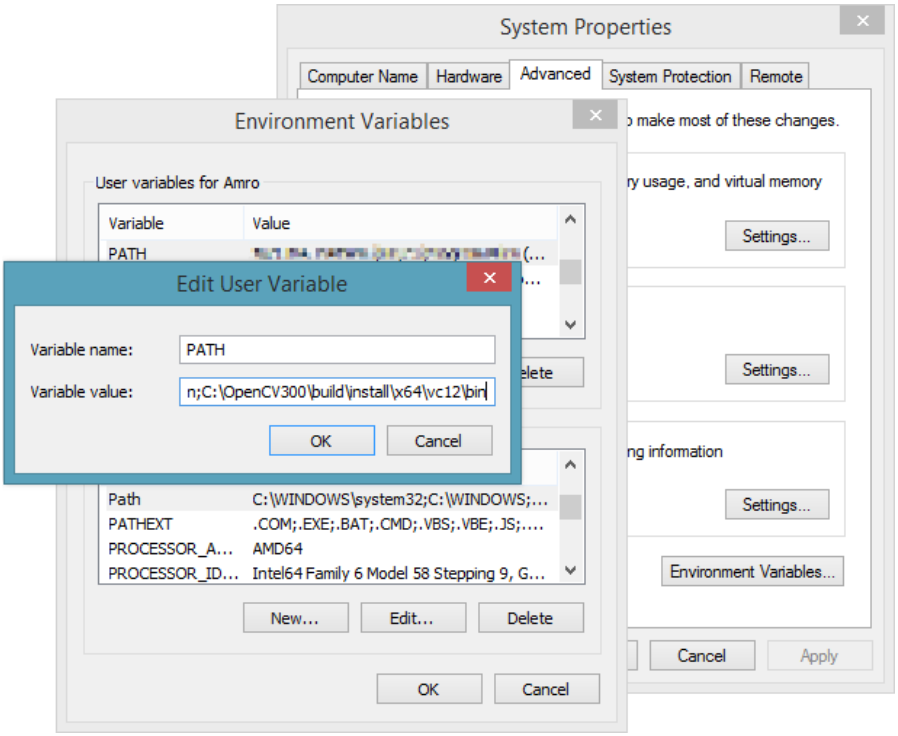


The resulting folder should be located at: `C:\dev\build\install`.



Finally you need to add the OpenCV binaries (the folder with OpenCV DLLs) to the `PATH` environment variable. Following these instructions, the directory to add is:

- `C:\dev\build\install\x64\vc14\bin`



2) mexopencv

In this final step, download `mexopencv` project (either clone the git repo, or download as ZIP file). Say you saved it to have this path:

- `C:\dev\mexopencv\README.markdown`

Next start MATLAB, and run the following:

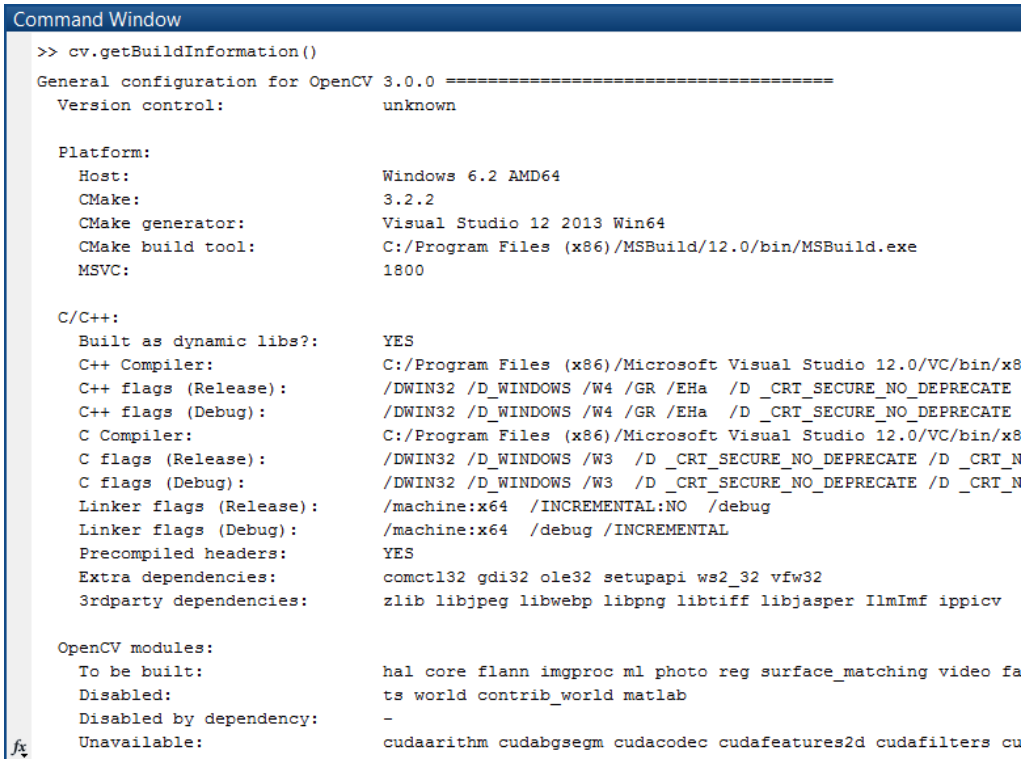
```
>> cd('C:\dev\mexopencv')
>> addpath('C:\dev\mexopencv')
>> addpath('C:\dev\mexopencv\opencv_contrib')
>> mexopencv.make('opencv_path','C:\dev\build\install', 'opencv_contrib',true)
```

(It should take several minutes to finish compilation).

Assuming everything went well, you can finally verify the generated MEX-files by running this in MATLAB:

```
>> cv.getBuildInformation()
```

You should see something like:



Congratulation, now you have OpenCV working in MATLAB!

Help: Stack Overflow (MATLAB, OpenCV) | MATLAB Answers | OpenCV Answers