# Build-OpenCv-for-Python-with-Extra-Modules-Windows-10-

This is a step by step guide to build OpenCV with Extra Modules for Python (Anaconda) for Windows without errors. Particularly, I will use the freetype module in OpenCV-Contrib.

If yours was anything like mine, you spent hours or even a few days trying to figure out how to get OpenCv running with Freetype2 and harfbuzz. I encountered several errors but found easy ways to go around the issues and complete the install. I hope this step by step guide helps you clear your errors.

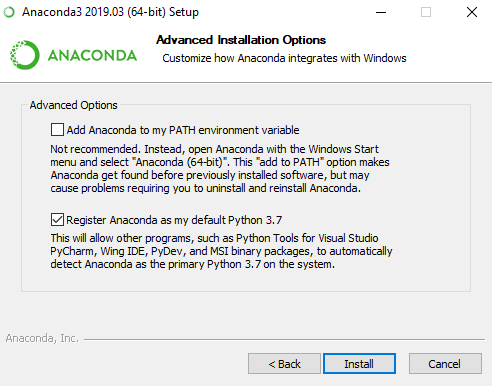
For me, the goal was to use OpenCV to print text in different fonts on images – synthetically generate text. OpenCv has a limited font type hence the reason of including freetype2 to read ttf files and render them to Opencv.

Before I decided to build OpenCv with freetype and Harfbuzz from scratch, I tried installing OpenCv with extra modules using PiP and Conda. For both methods OpenCv installed completely but did not link correctly to freetype and harfbuzz even when I tried PiP installing the OpenCv-Contrib-python as found on <https://pypi.org/project/opencv-contrib-python/>. I tried several variations including installing freetype first then harfbuzz before PiP installing OpenCv-Contrib-Python. In all, OpenCv installed correctly by did not link to freetype/harfbuzz, I kept getting - ModuleNotFoundError: No module named 'cv2'.

**BUILDING OPENCV WITH FREETYPE AND HARFBUZZ ENABLED**

*PREREQUISITES: Setup your machine and Installation tools:*

* Windows 10 (x64)
* OpenCv and OpenCV-Contrib source files – I show where to get this in a later section. I installed OpenCv-4.1.0
* Anaconda3 – installed from [www.anaconda.org](http://www.anaconda.org). Download the latest release. During installation,:
  + Set as the default python environment
  + Do Not add to path – we will do this in a later section when we create a PYTHONPATH environmental variable.



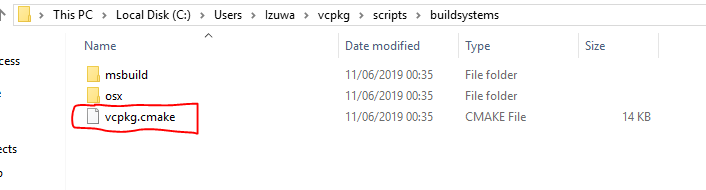
* Python 3.6 – installed during anaconda install
* Cmake 3.15.0 – This is needed to configure OpenCV and generate scripts that will be passed to Visual Studio during build and release.
  + Download the windows installer from <https://cmake.org/download/> . Just double click the downloaded MSI installer and follow normal windows installation process.
* Vcpkg - This is needed to install Freeteype2 and harfbuzz
  + Vcpkg does not need to be installed. To get it working, you’ll need to clone the vcpkg folder from Microsoft github page (instructions below).
  + If you don’t already have git for windows installed, you can get git here - <https://git-scm.com/downloads>.
  + Open up your git bash
  + Enter the following commands:
    1. *git clone* [*https://github.com/Microsoft/vcpkg.git*](https://github.com/Microsoft/vcpkg.git)
    2. The above command copies the needed files to your system. Mine copied to folder located at - C:\Users\<myusername>\vcpkg
    3. Use the *cd* command of git bash to navigate to the vcpkg folder above. For me I had to enter *cd vcpkg*
    4. Run the bootstrap-vcpkg.bat file using. Type (including the leading dot):- *.\bootstrap-vcpkg.bat*
    5. Search for freetype using – *vcpkg search freetype* . This shows you what freetype package is available.
    6. Install freetype using *vcpkg install boost:x86-windows* . Take note of the :x86-windows; for my system since I had x64, my command looked like - *vcpkg install boost:x64-windows*
    7. Use a similar step from 5 – 6 to install harfbuzz .

These vcpkg processes generate a toolchain file which we will use later in our configuring of OpenCv. This file will have the installation details for freetype2 and harfbuzz which OpenCV extra modules needs to ensure freetype is properly linked to OpenCv.

Take note of the location of this script file – *vcpkg.cmake*; it will be in the vcpkg install directory. For me it was located at –

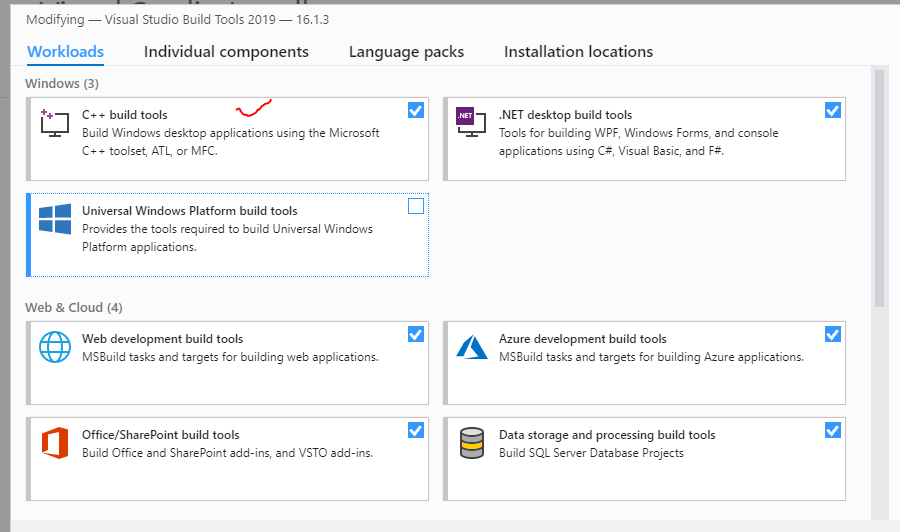
*~\scripts\buildsystems\vcpkg.cmake*

(~ refers to the path of the vcpkg folder on my system)



The above steps are available in more details and other OS types on - <https://docs.microsoft.com/en-us/cpp/build/vcpkg?view=vs-2019> .

* Visual Studio 2019 – this will be used for building and releasing OpenCv. You could also use Visual studio 2017. Be sure to include the C + + build tools. You can download visual studio from - <https://visualstudio.microsoft.com/downloads/> .



**CONFIGURE OPENCV SOURCE FOR BUILD WITH CMAKE:**

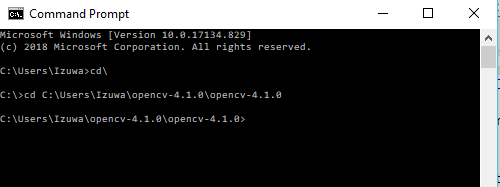
In my installation, I noticed that the CMAKE-GUI didn’t enable most of the flags I wanted. So I used both the command line and GUI.

Using the command line forces configuration for python which I found didn’t happen using the GUI alone. I basically run the command on the command line and use the GUI to double check that all the parameters are enabled correctly. Doing this eliminated errors I had previously gotten like – *freetype found: NO, importerror: opencv loader: missing configuration file: [‘config.py’]. check opencv installation, cv2 not found, ModuleNotFoundError: No module named ‘cv2’, in <module> import cv2 ImportError: DLL load failed: The specified module could not be found.*

STEPS:

In carrying out this steps in CMAKE, if you encounter any errors or make a mistake, go back to the build folder and delete its content so that you have a fresh, clean build.

1. Download the source files for OpenCv and OpenCV-Contrib. You can find the latest release here:
   1. OpenCv – <https://github.com/opencv/opencv/releases>
   2. OpenCv-Contrib - <https://github.com/opencv/opencv_contrib/releases>.
2. Extract the files to a suitable folder
3. Create an empty folder called build in the opencv root folder i.e. the folder containing the other folders like include, doc, data, apps etc.
4. Open command prompt.
5. Use the cd command to navigate to the root folder; in my case - *cd C:\Users\Izuwa\opencv-4.1.0\opencv-4.1.0*



1. Before running the final script to configure OpenCV, you will need to modify the CMakelists.txt file for freetype. Without this modification, CMAKE didn’t recognize freetype and harfbuzz as installed. UnaNancyOwen made this script available in his github repository here - <https://gist.github.com/UnaNancyOwen/14c72a3f10a46d41c359ab6ea307a1d2#file-readme-md>

Open the CMakeLists.txt on your system at ~/opencv\_contrib/modules/freetype/CMakeLists.txt

Delete the content, copy the new content from the github link above, paste it in and save the file.

1. Run the following CMAKE Command (I put them in individual lines so you can read each of the command; ideally you should run it all together like the screen shot below). If there are spaces in your file path, enclose the entire file path in parenthesis.

*cmake -G "Visual Studio 16 2019"*

*-B C:\Users\Izuwa\opencv-4.1.0\opencv-4.1.0\build*

*-D BUILD\_NEW\_PYTHON\_SUPPORT=ON*

*-D BUILD\_PYTHON\_SUPPORT=ON*

*-D BUILD\_opencv\_python3=yes*

*-D PYTHON\_DEFAULT\_EXECUTABLE=C:\Users\Izuwa\Anaconda3\python.exe*

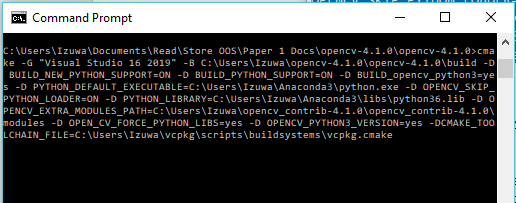
*-D OPENCV\_SKIP\_PYTHON\_LOADER=ON*

*-D PYTHON\_LIBRARY=C:\Users\Izuwa\Anaconda3\libs\python36.lib*

*-D OPENCV\_EXTRA\_MODULES\_PATH=C:\Users\Izuwa\opencv\_contrib-4.1.0\opencv\_contrib-4.1.0\modules*

*-D OPEN\_CV\_FORCE\_PYTHON\_LIBS=yes*

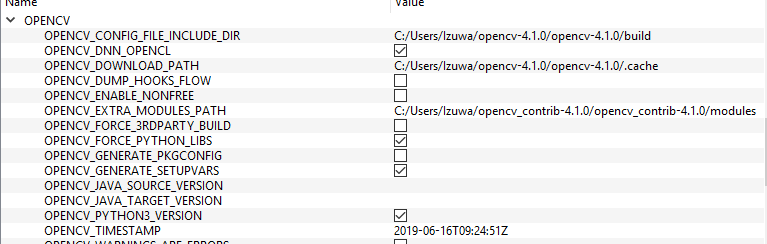
*-DCMAKE\_TOOLCHAIN\_FILE=C:\Users\Izuwa\vcpkg\scripts\buildsystems\vcpkg.cmake*



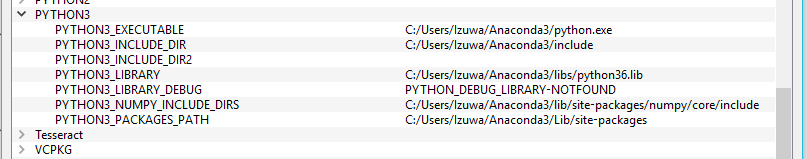
Some important parts of the script:

* -B takes a parameter of the path of the empty build directory you created.
* PYTHON\_DEFAULT\_EXECUTABLE – the path to the python.exe. in my case since Anaconda3 is my default python environment, python.exe is in the root folder for Anaconda3
* PYTHON\_LIBRARY – in my case I have python 3.6 installed. The python36.lib is located in the libs folder from the root anaconda installation folder.
* OPENCV\_EXTRA\_MODULES\_PATH – is the location of the “modules” folder from the extracted OpenCV-Contrib folder. In my case it was - C:\Users\Izuwa\opencv\_contrib-4.1.0\opencv\_contrib-4.1.0\modules
* Finally, -DCMAKE\_TOOLCHAIN\_FILE needs the path of the vcpkg.cmake script generated after installing freetype2 and harfbuzz. If you didn’t take note of this path previously, refer to the section above on using vcpkg to install freetype2 and harfbuzz for details of the location of this file.

1. After this completes in command line. Open the CMake GUI, browse for the folder of the source code folder (root folder) and build folder and then click File-Reload Cache. Also ensure the grouped and Advanced check box are checked (this makes it easier to read). Then check the following:
   1. Click on Build, make sure BUILD\_opencv\_python\_bindings\_generator is checked.
   2. Click on ENABLE, make sure in addition to others selected ENABLE\_PYLINT is also selected
   3. Under INSTALL, you can choose to INSTALL\_PYTHON\_EXAMPLES – this will take a longer time to build
   4. CLICK on OPENCV, make sure:
      1. OPENCV\_EXTRA\_MODULES\_PATH has the right path if not, click the entry field and input the right path
      2. OPENCV\_PYTHON3\_VERSION is checked
      3. OPENCV\_FORCE\_PYTHON\_LIBS is checked



* 1. Click on PYTHON3 and make sure:
     1. PYTHON3\_EXECUTABLE has the right path
     2. PYTHON3\_INCLUDE\_DIR has the right path
     3. PYTHON3\_LIBRARY has the right lib file link
     4. PYTHON3\_NUMPY\_INCLUDE\_DIRS has the right link (numpy is required; if you have not already installed it, you can open anaconda prompt and type - *conda install -c conda-forge numpy*)
     5. PYTHON3\_PACKAGES\_PATH has the right link

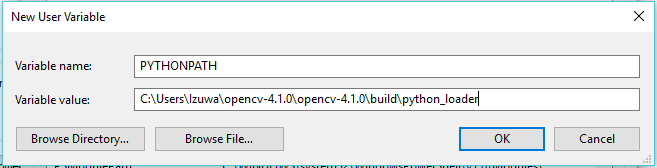


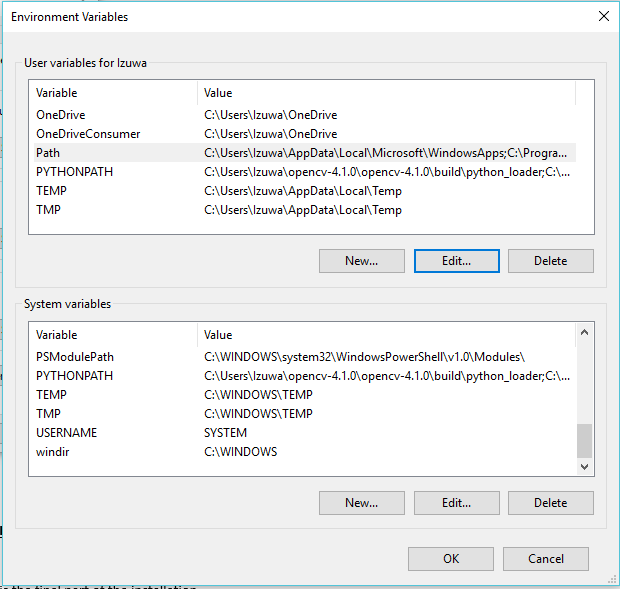
* 1. Finally click Configure (once) and then generate

At this point, you are ready to build the project and release it with Visual Studio.

1. NOTE: As part of the output written out as CMAKE configures the build, CMAKE produces the following - *OpenCV Python: during development append to PYTHONPATH: C:/Users/Izuwa/opencv-4.1.0/opencv-4.1.0/build/python\_loader*

You will need to create an environment Variable called PYTHONPATH and add this folder path specified by CMAKE. In my installation, I added the default python path (C:\Users\<localName>\Anaconda3\ and also C:\Users\<localName>\Anaconda3\Scripts. I made the OpenCV Python path first on the list like the screen shot below:





**BUILD AND RELEASE OPENCV WITH VISUAL STUDIO:**

This is the final part of the installation.

Steps:

1. Open the build folder in the OpenCv directory
2. You should find a Visual Studio .sln file – (OpenCV.Sln) – double click it to open with Visual Studio
3. On the right pane, right click INSTALL and click build. Ensure that the solution configuration is Debug and the solution platform corresponds to that of your system (in my case x64). This build will take a while – let it run.
4. Finally, change the solution configuration to release and right click INSTALL and click build.

**TESTING THAT OPENCV IS INSTALLED:**

At this point, you are ready to test that openCv is running on anaconda.

Remember to complete this step:

NOTE: As part of the output written out as CMAKE configures the build, CMAKE produces the following - *OpenCV Python: during development append to PYTHONPATH: C:/Users/Izuwa/opencv-4.1.0/opencv-4.1.0/build/python\_loader*

You will need to create an environment Variable called PYTHONPATH and add this folder path specified by CMAKE. In my installation, I added the default python path (C:\Users\<localName>\Anaconda3\ and also C:\Users\<localName>\Anaconda3\Scripts. I made the OpenCV Python path first on the list like the screen shot below:

To test that the installation was successful,

Open spyder, jupyter or python prompt and type

*Import cv2*

*Cv2.\_\_version\_\_*

If both run successfully, you have completed the installation of OpenCV with freetype2

**SOME COMMON ERRORS ENCOUNTERED:**

Some of the common errors encountered during installation are listed below. They are caused by either improper installation, the python loader for opencv is not seen by the python environment, or CMAKE didn’t configure the build to include python environment.

AttributeError: module 'cv2' has no attribute 'freetype'

importerror: opencv loader: missing configuration file: ['config.py']. check opencv installation.

cv2 not found

modulenotfounderror no module named 'cv2' windows

importerror: dll load failed: the specified module could not be found.

**SOME USEFUL REFERENCES:**

<https://vcpkg.readthedocs.io/en/latest/examples/installing-and-using-packages/>

<https://docs.microsoft.com/en-us/cpp/build/vcpkg?view=vs-2019>

<https://stackoverflow.com/questions/38051220/when-build-opencv-could-not-find-the-file-cv2-so/39651947>

<https://stackoverflow.com/questions/15790501/why-cv2-so-missing-after-opencv-installed>

<https://answers.opencv.org/question/173247/how-to-build-opencv-with-freetype-and-harfbuzz/>

<https://docs.opencv.org/4.1.0/d9/dfa/classcv_1_1freetype_1_1FreeType2.html#aba641f774c47a70eaeb76bf7aa865915>

<https://github.com/opencv/opencv/issues/13202>

<https://fireant.github.io/misc/2017/01/28/ttf-opencv.html>