One of the required modules called “SFM” is not included in the default version of openCV and needs to be compiled from source. This module also depends on other libraries: eigen3,ceres, google’s gflags and glog libraries.

1. Create a virtual environment using (where myenv is the name of the new envirment:

```
python -m venv myenv
```

2. To activate it

```
source myenv/bin/activate (mac/linux)
.\myenv\Scripts\Activate.ps1 (windows)
```

Instead of compiling all the libraries manually there is a better way by using VCPKG which is a package manager for C++.

1. Install VCPKG from https://github.com/microsoft/vcpkg
2. Follow official guide: https://learn.microsoft.com/en-us/vcpkg/get_started/get-started?pivots=shell-powershell
3. Include VCPKG in Path

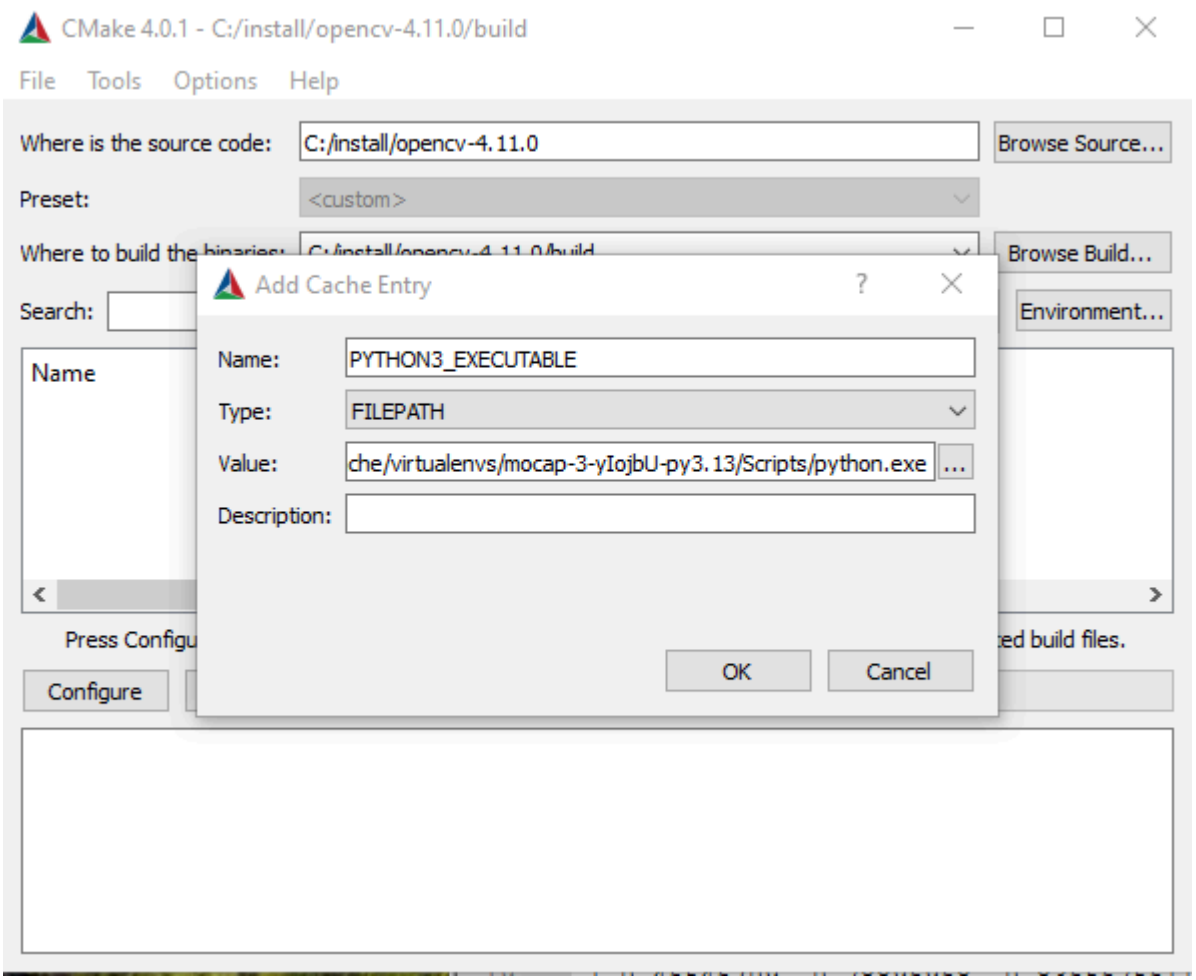
```
$env:VCPKG_ROOT = "C:\path\to\vcpkg"
$env:PATH = "$env:VCPKG_ROOT;$env:PATH"
```

```
vcpkg install vtk:x64-windows eigen3:x64-windows gflags:x64-windows glog:x64-windows ceres:x64-windows hdf5:x64-windows ffmpeg:x64-windows
```

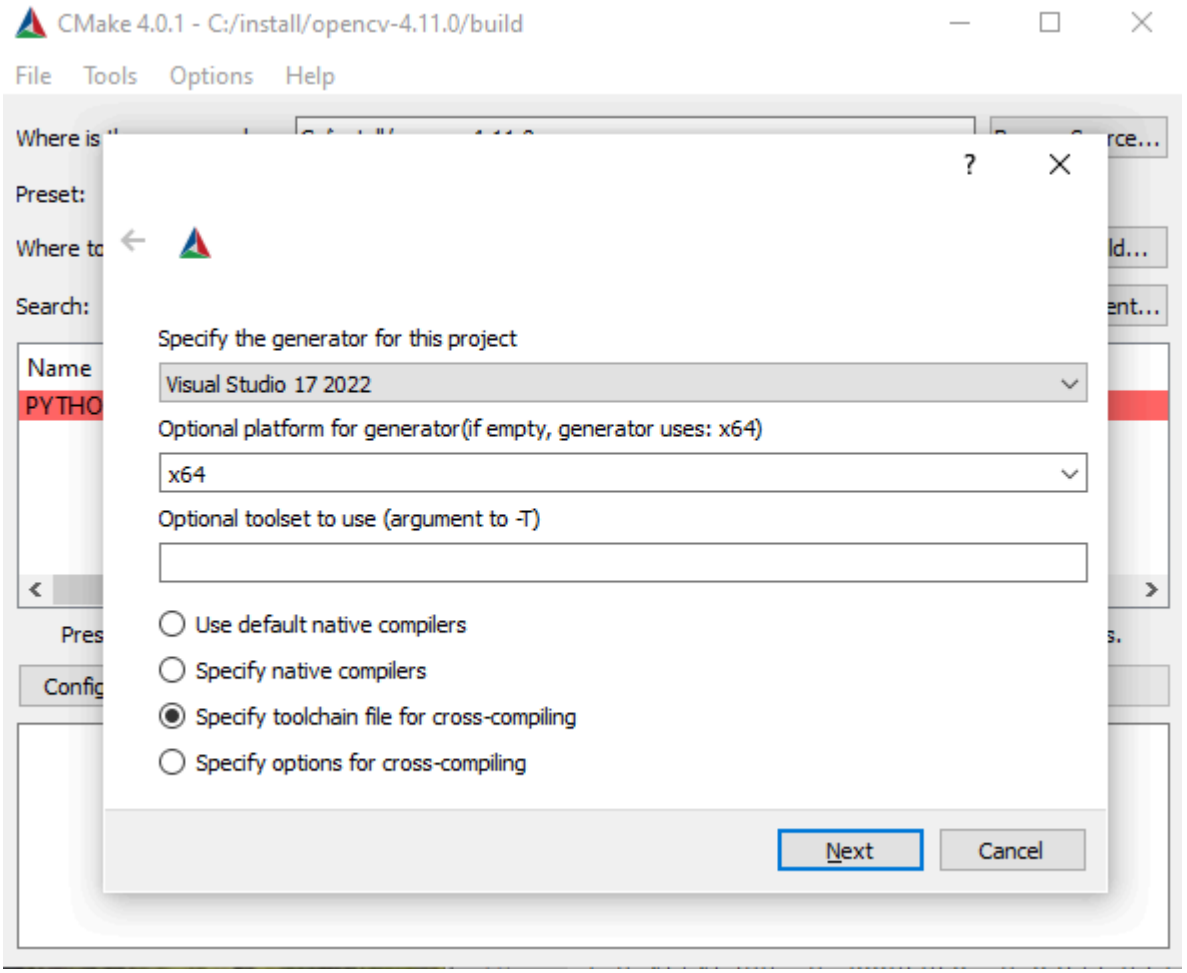
```
git clone https://github.com/opencv/opencv_contrib.git
git clone https://github.com/opencv/opencv.git
```

Start cmakegui and specify the source and build directories:

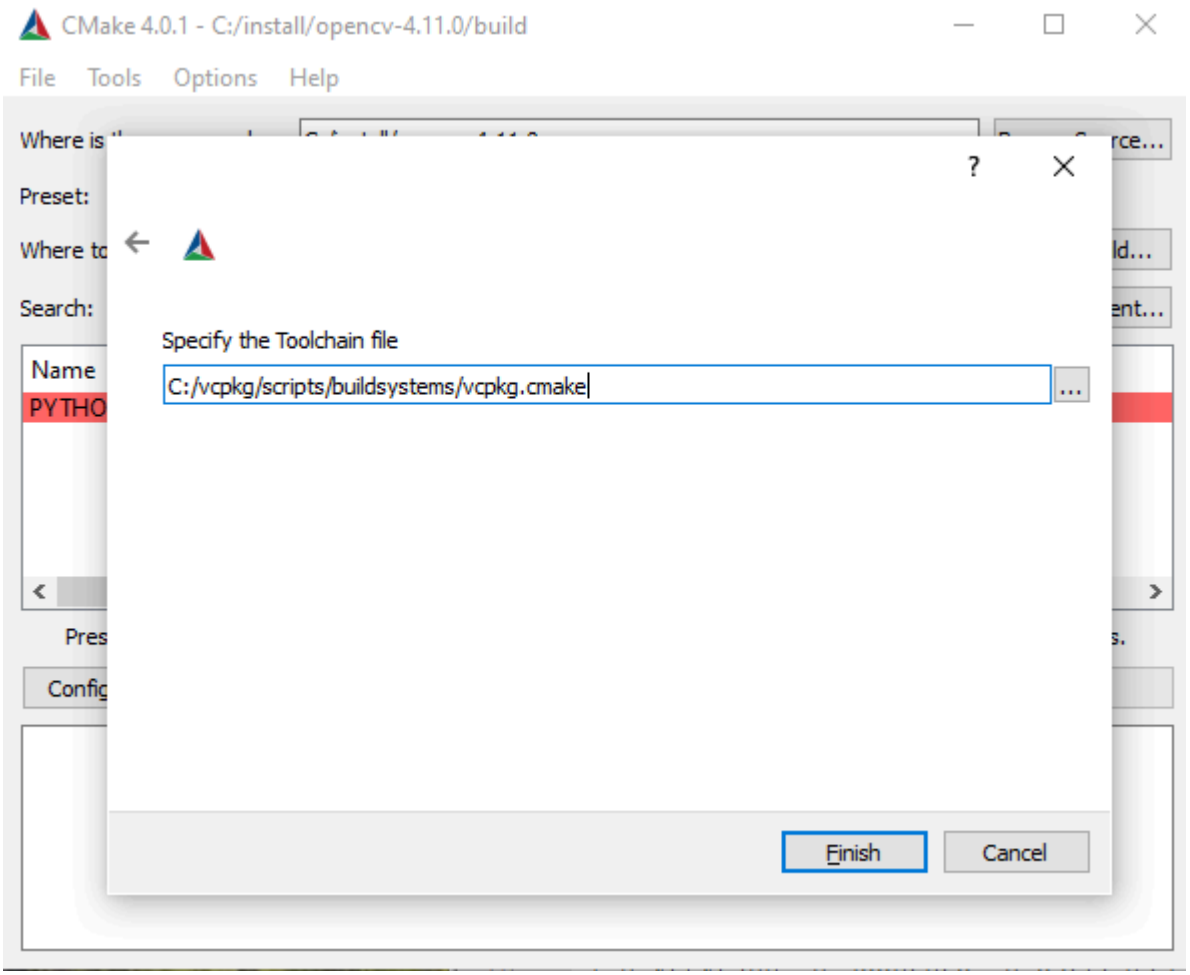
Click on Add Entry



Click on configure and set “specify toolcahin file for cross-compiling

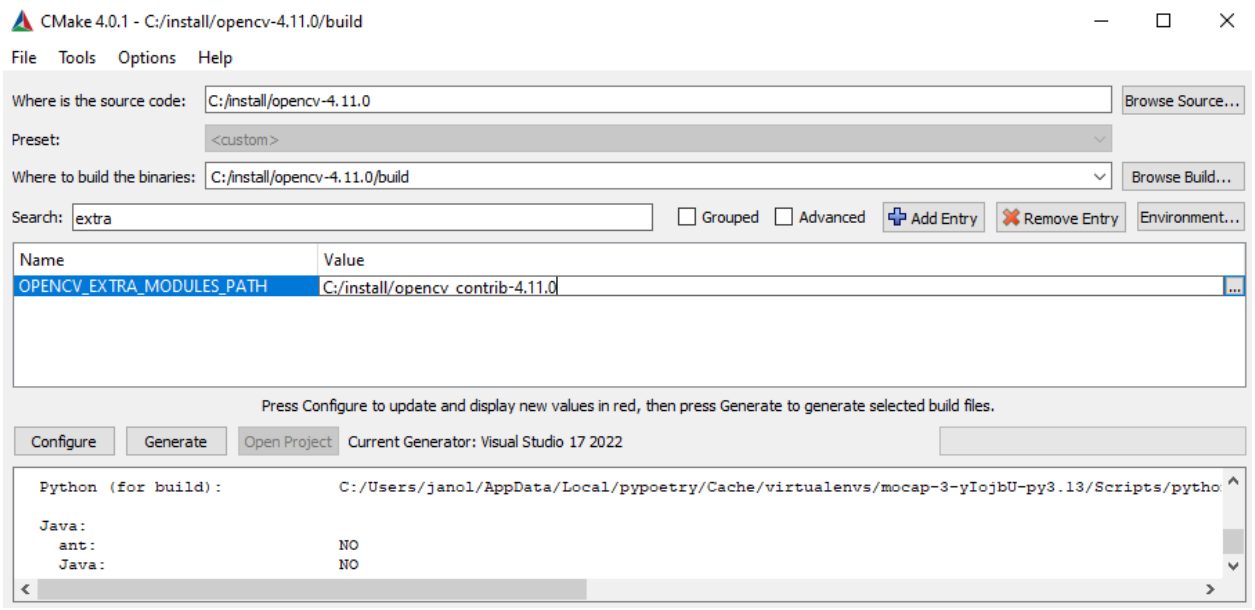


Find the toolchain in the vcpk installed directory



`Click on finish and wait`

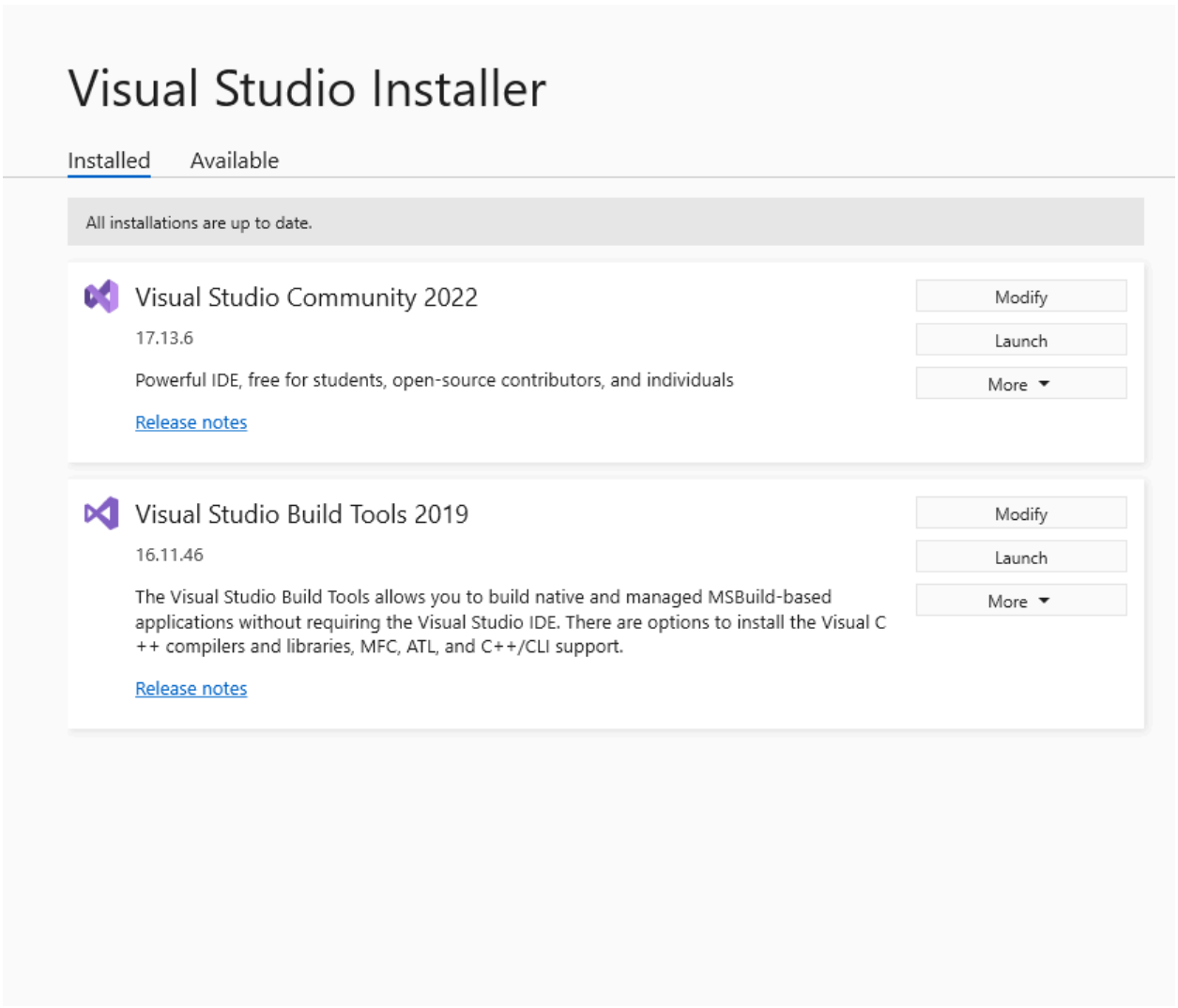
Set OPENCV_ENABLE_NONFREE
Set OPENCV_FORCE_PYTHON_LIBS
Set OPENCV_PYTHON3_VERSION
Set OPENCV_EXTRA_MODULES_PATH to the path where the “opencv-contrib” was saved

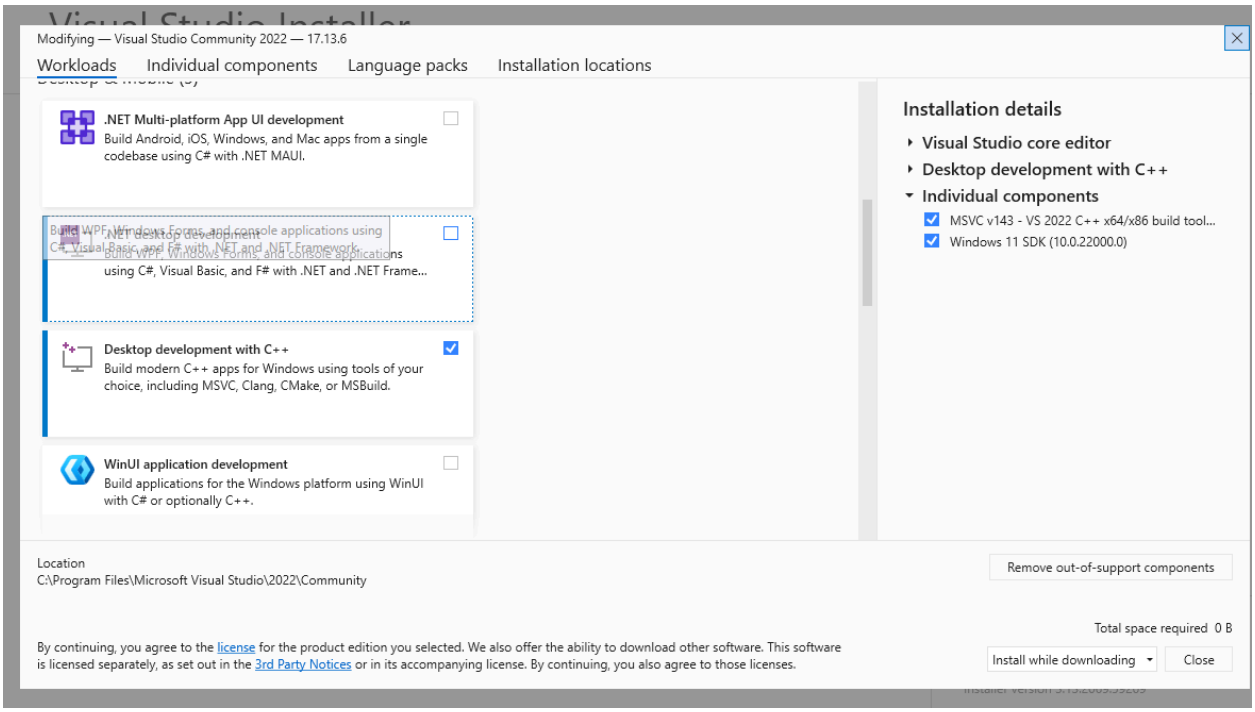


The “...” button can be used to select the path using explorer.

Click on Configure and than Generate

Finally open the project in Visual Studio 2022
Build all and build INSTALL


Visual Studio Installer
Installed
Available
All installations are up to date.
Visual Studio Community 2022
17.13.6
Powerful IDE, free for students, open-source contributors, and individuals
Release notes
Modify
Launch
More
Visual Studio Build Tools 2019
16.11.46
The Visual Studio Build Tools allows you to build native and managed MSBuild-based applications without requiring the Visual Studio IDE. There are options to install the Visual C ++ compilers and libraries, MFC, ATL, and C++/CLI support.
Release notes
Modify
Launch
More


Modifying — Visual Studio Community 2022 — 17.13.6
Workloads
Individual components
Language packs
Installation locations
.NET Multi-platform App UI development
Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.
.NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...
Desktop development with C++
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
WinUI application development
Build applications for the Windows platform using WinUI with C# or optionally C++.
Installation details
Visual Studio core editor
Desktop development with C++
Individual components
MSVC v143 - VS 2022 C++ x64/x86 build tool...
Windows 11 SDK (10.0.22000.0)
Location
C:\Program Files\Microsoft Visual Studio\2022\Community
Remove out-of-support components
Total space required 0 B
By continuing, you agree to the license for the product edition you selected. We also offer the ability to download other software. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.
Install while downloading
Close