

Installation manual for Microsoft Windows

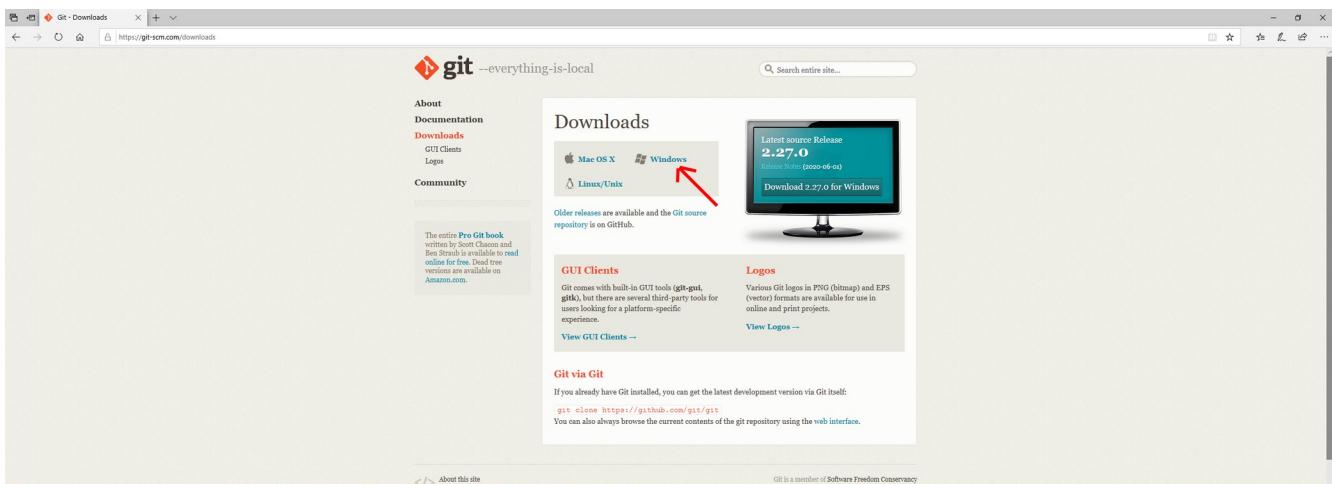
Table of Contents

Git.....	2
Python.....	3
CMake.....	4
mogasens_csv.....	5

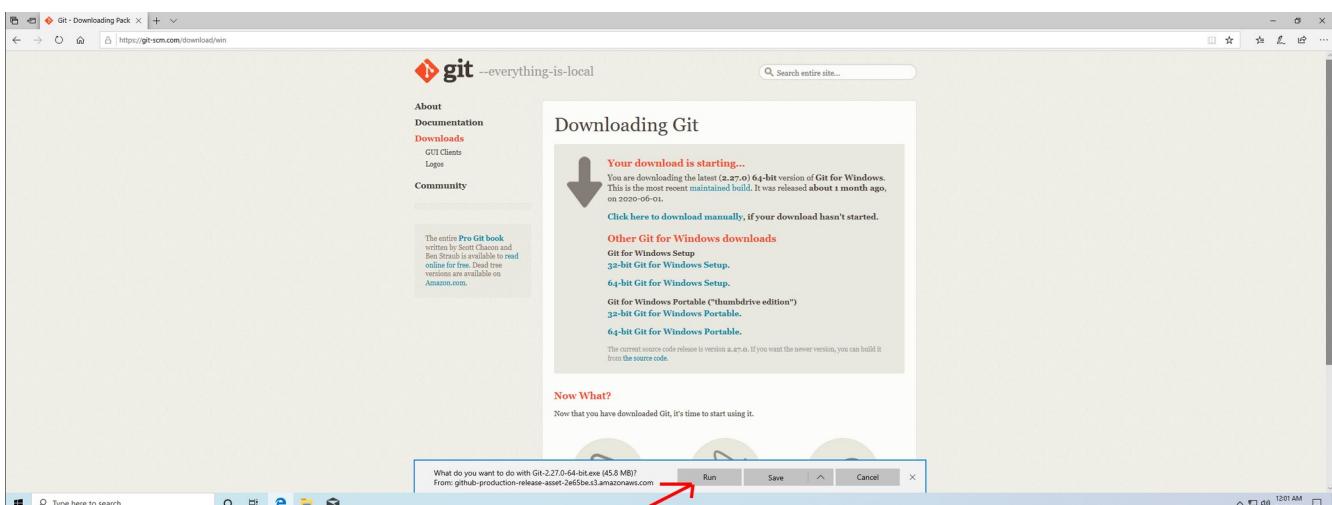
Git



Open git-scm.com in Microsoft Edge and click on “Downloads”.



Click on “Windows”.



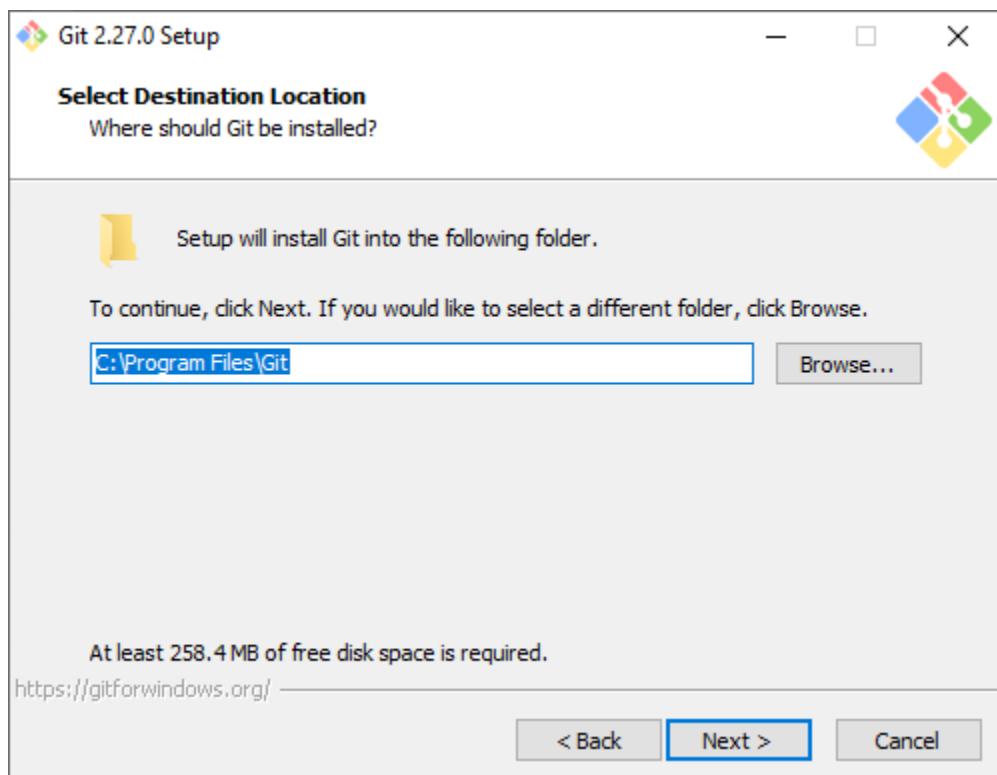
Microsoft Edge will show a prompt. Click on “Run”.



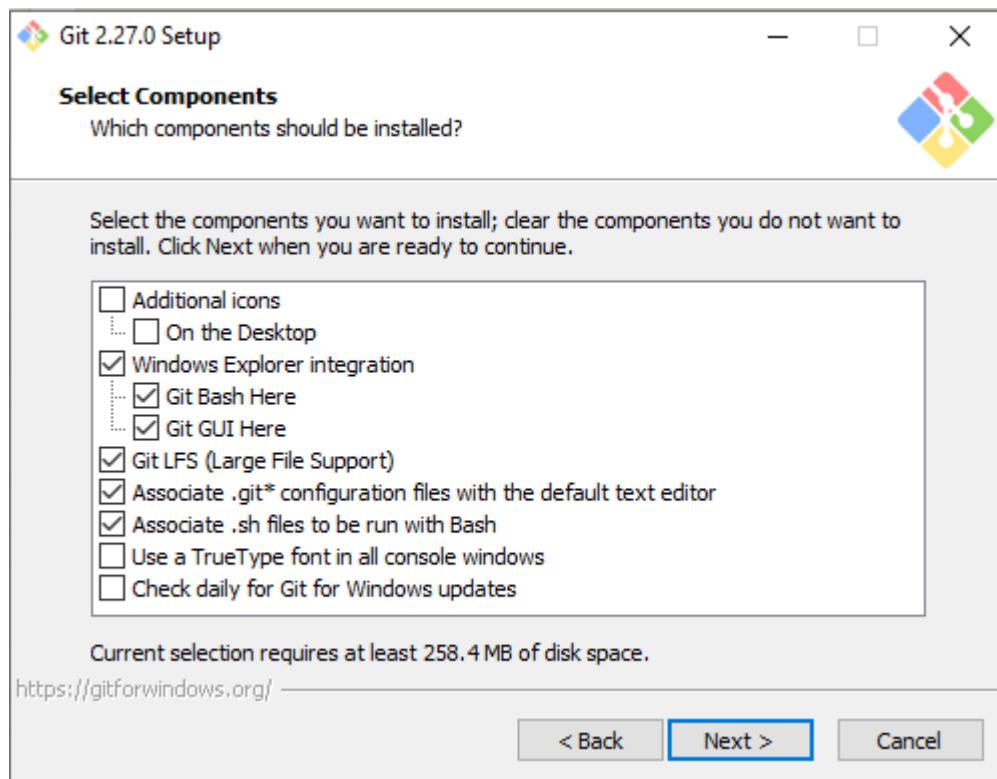
Microsoft Windows will show a UAC prompt, click on “Yes”.



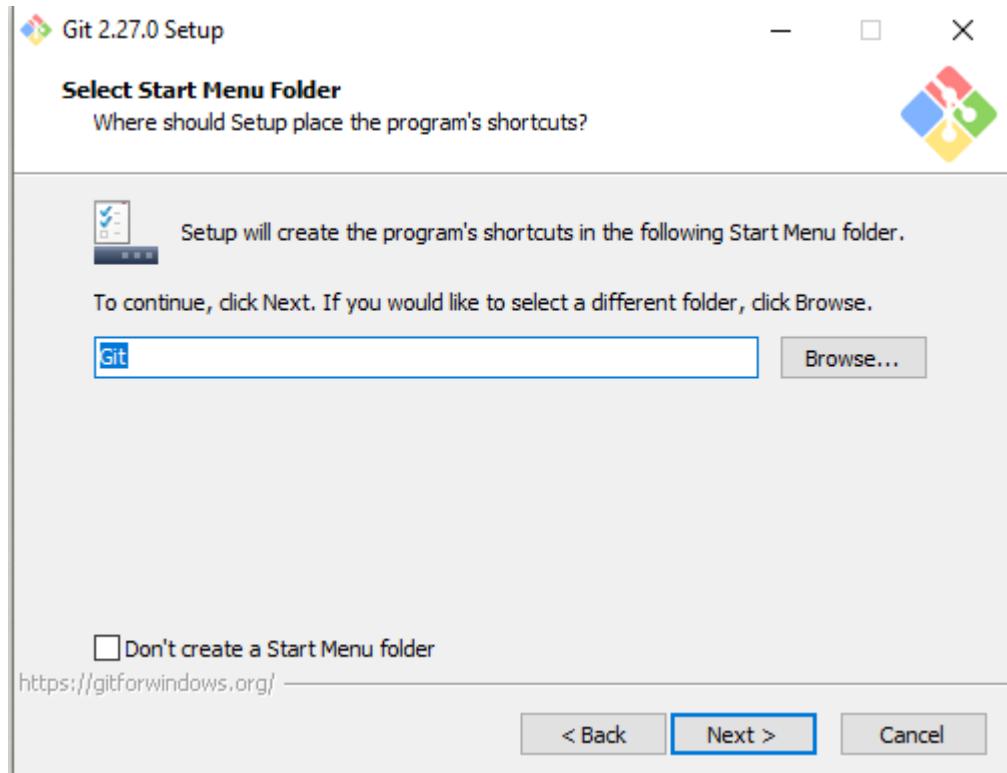
Click “Next”.



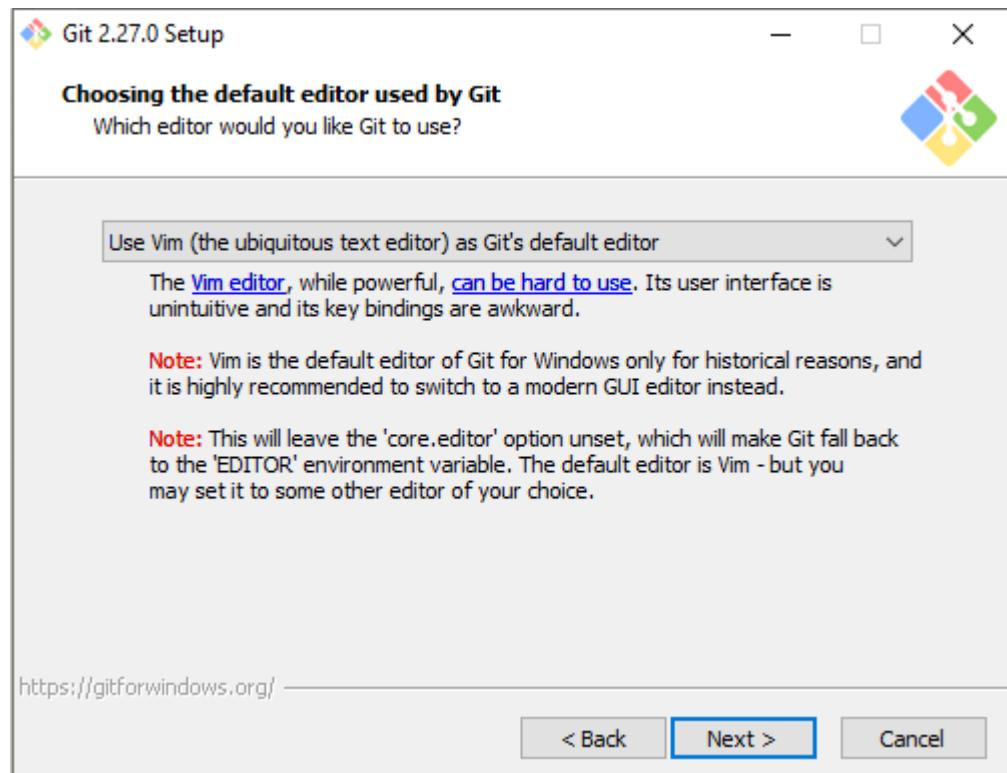
Select an installation directory, then click on “Next”.



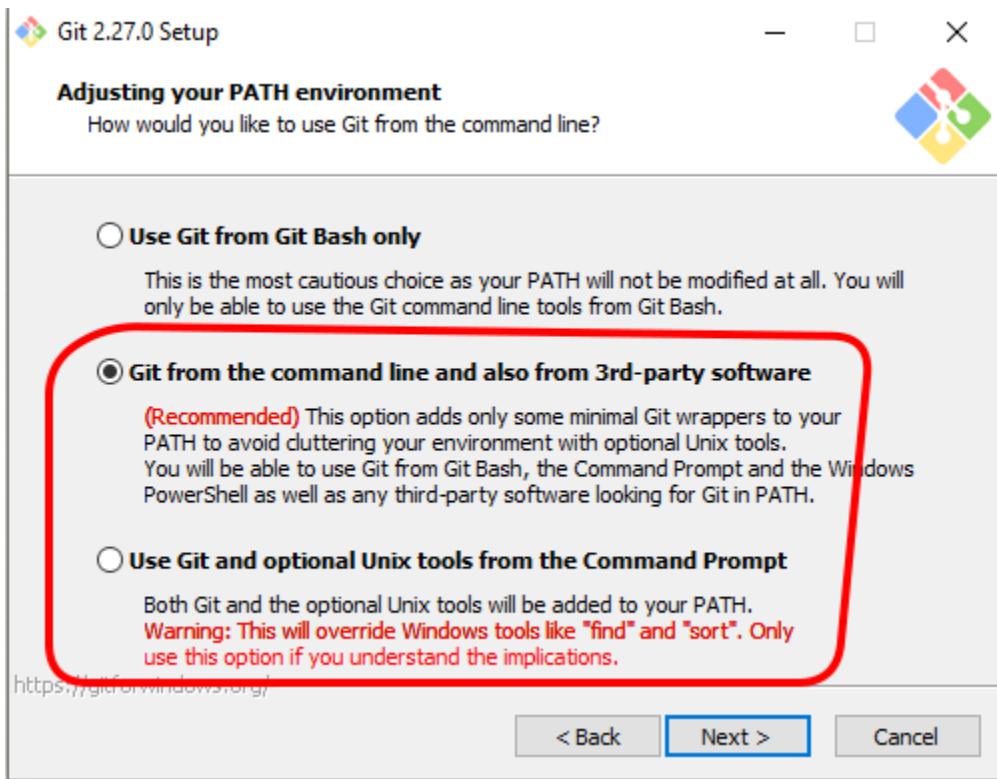
Select whichever components you want, then click on “Next”.



Select a “Start Menu folder”, then click on “Next”.

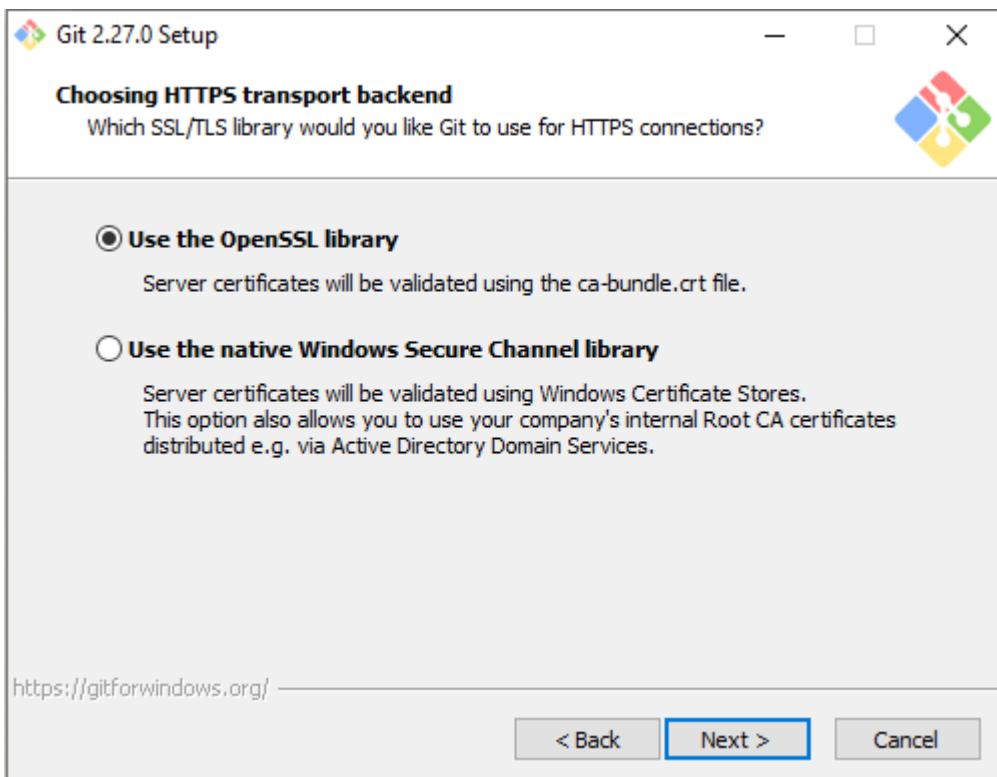


Choose whichever editor you want, then click on “Next”.

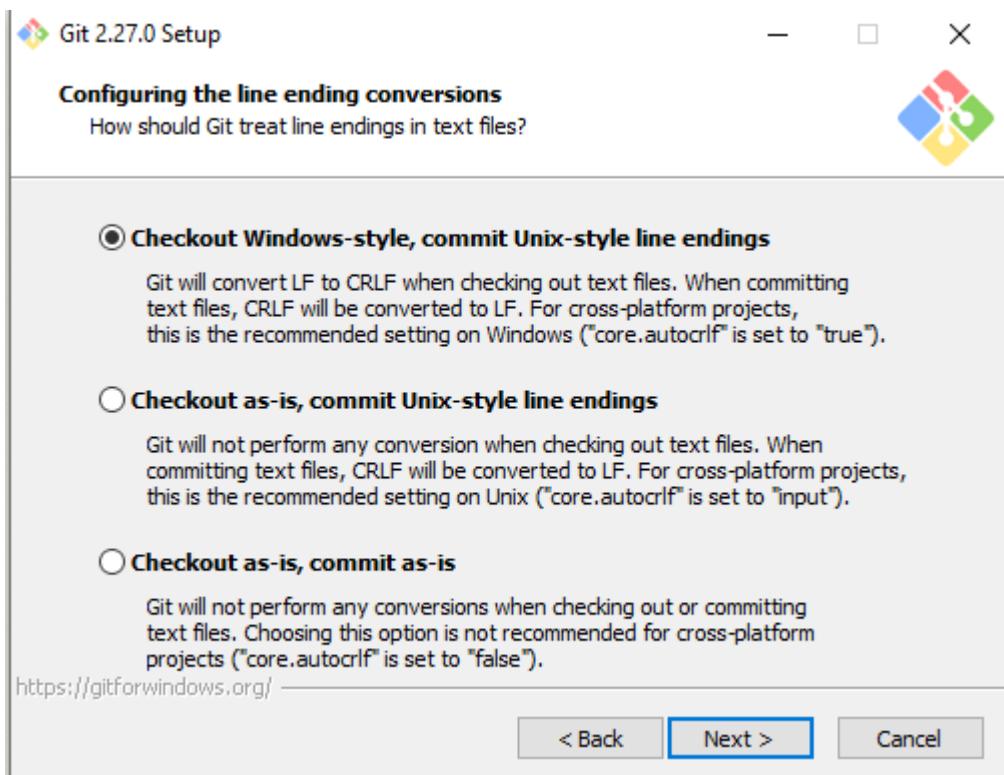


Choose “Git from the command line and also from 3rd-party software” or “Use Git and optional Unix tools from the Command Prompt” so that Git is available from within a Windows Command Prompt.

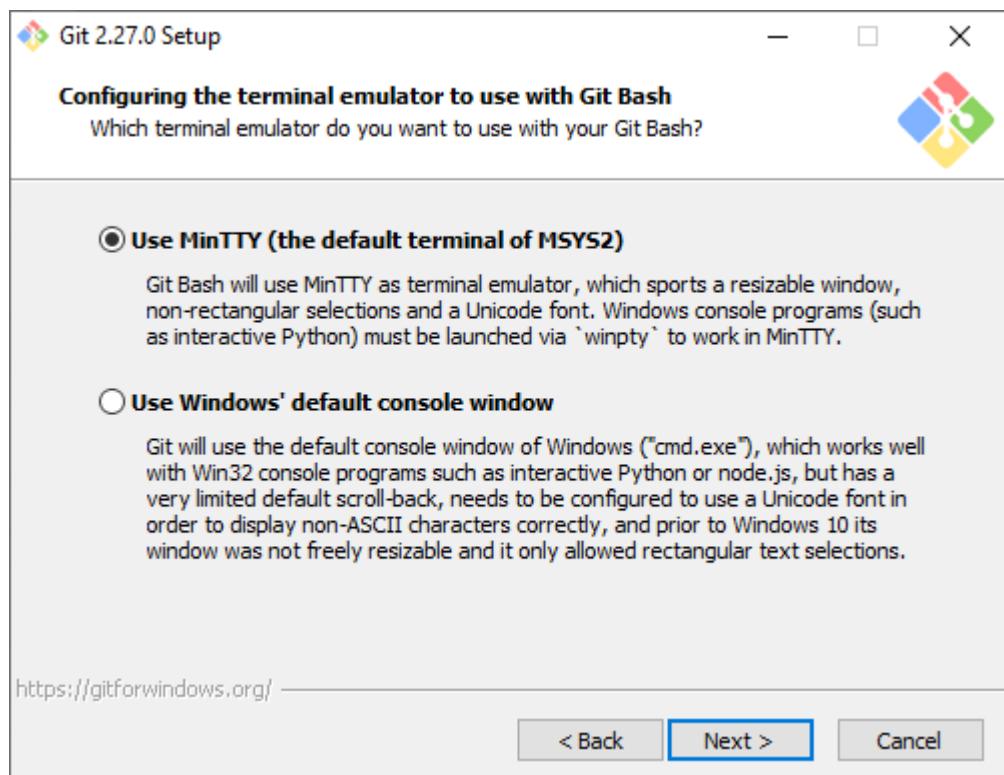
Then click “Next”.



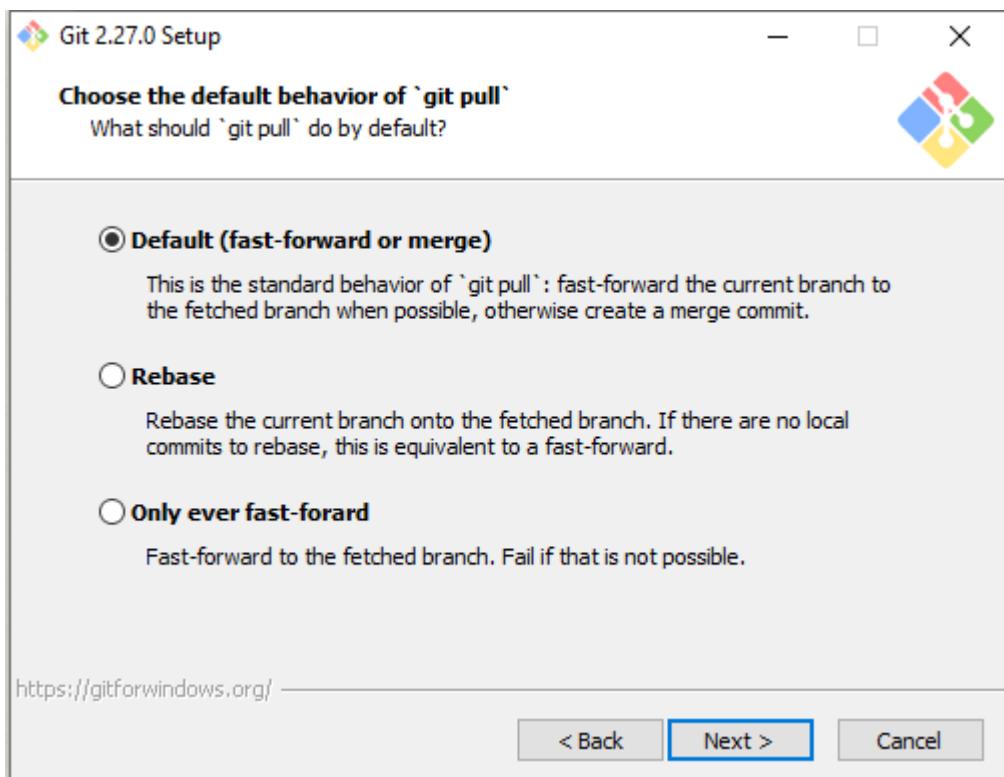
Choose whichever HTTPS transport backend you want, then click on “Next”.



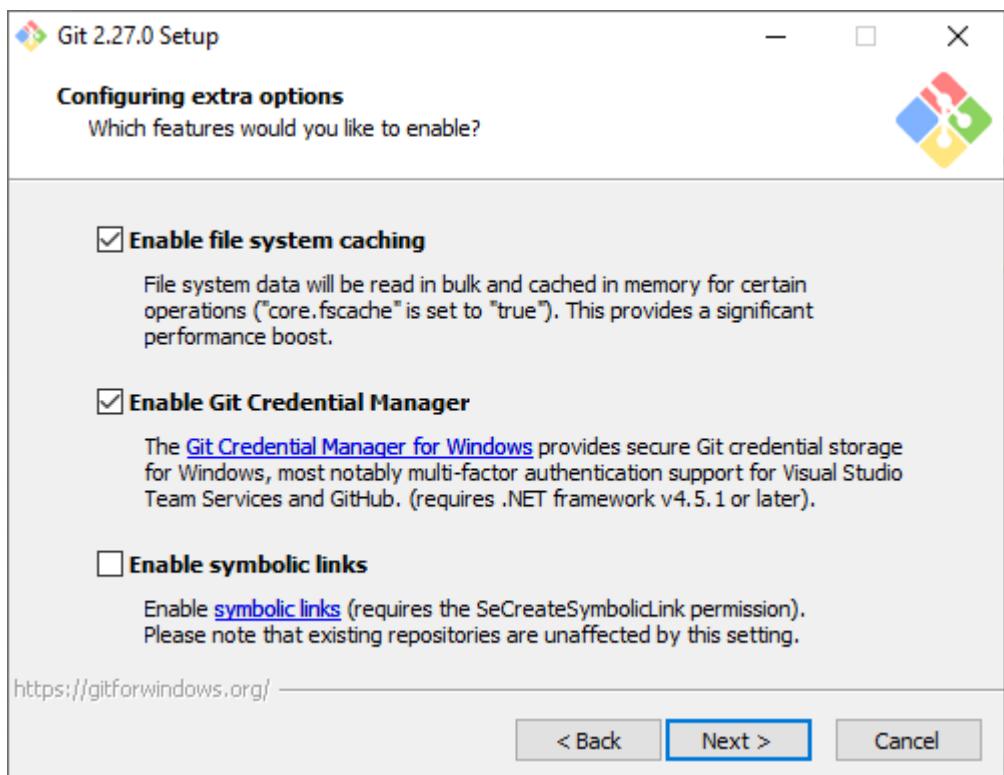
Configure the line ending conversions as you want, then click on “Next”.



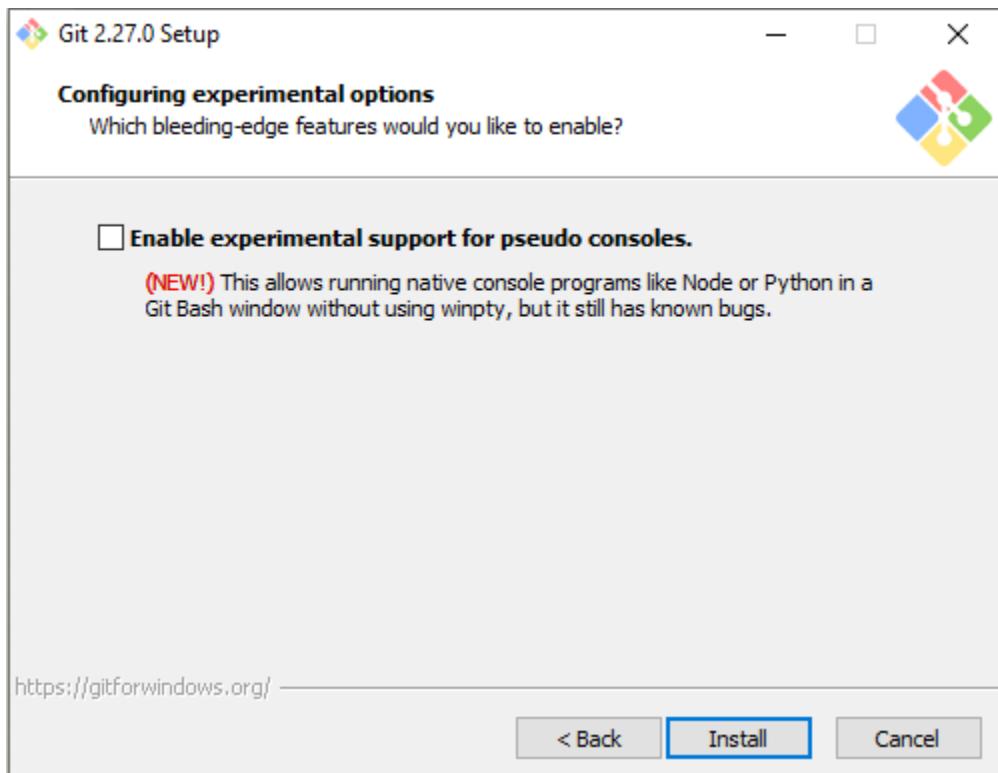
Choose whichever terminal emulator you want, then click “Next”.



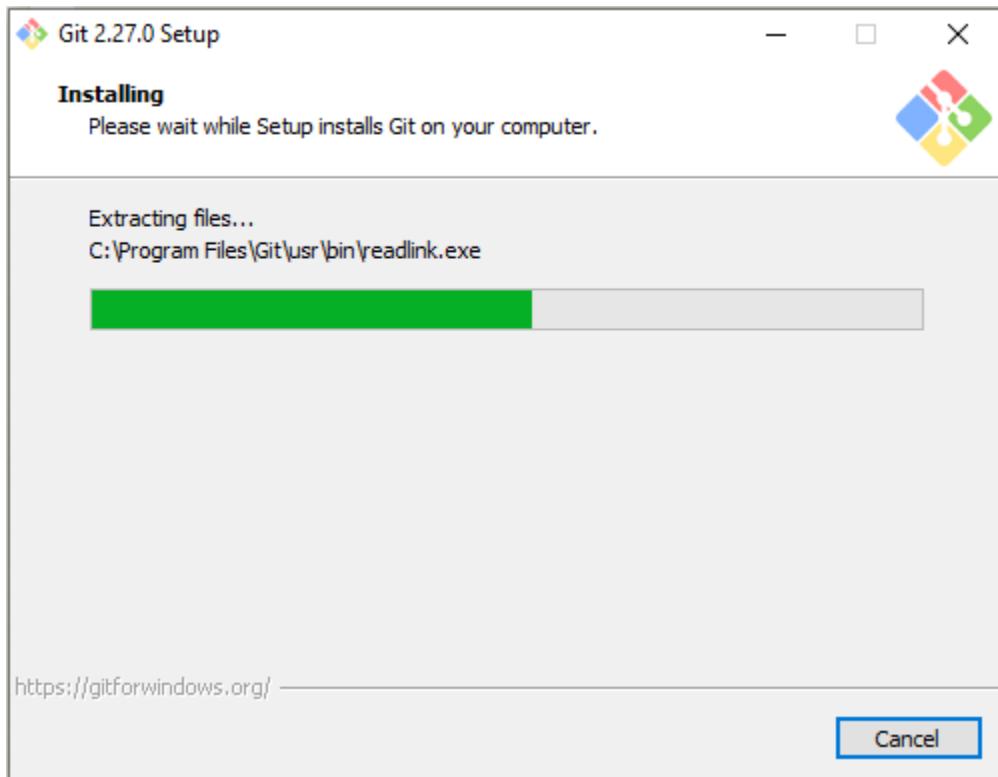
Choose whichever 'git pull' behavior you want, then click "Next".



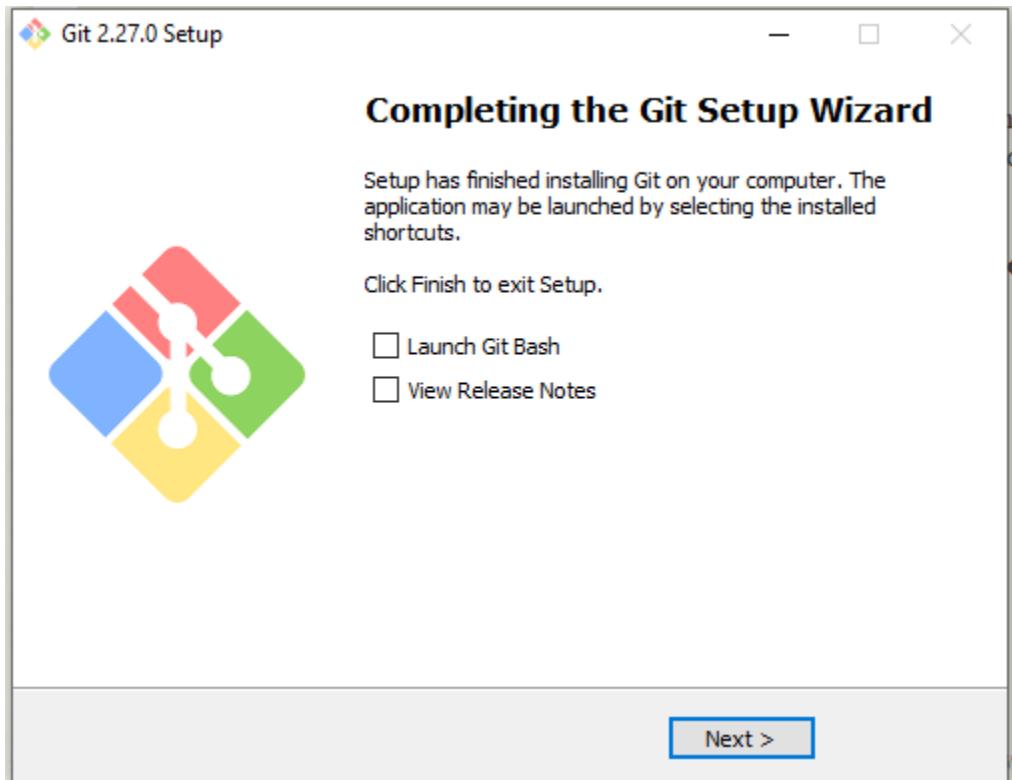
Choose whichever extra options you want, then click on "Next".



Choose whichever experimental options you want, then click “Install”.

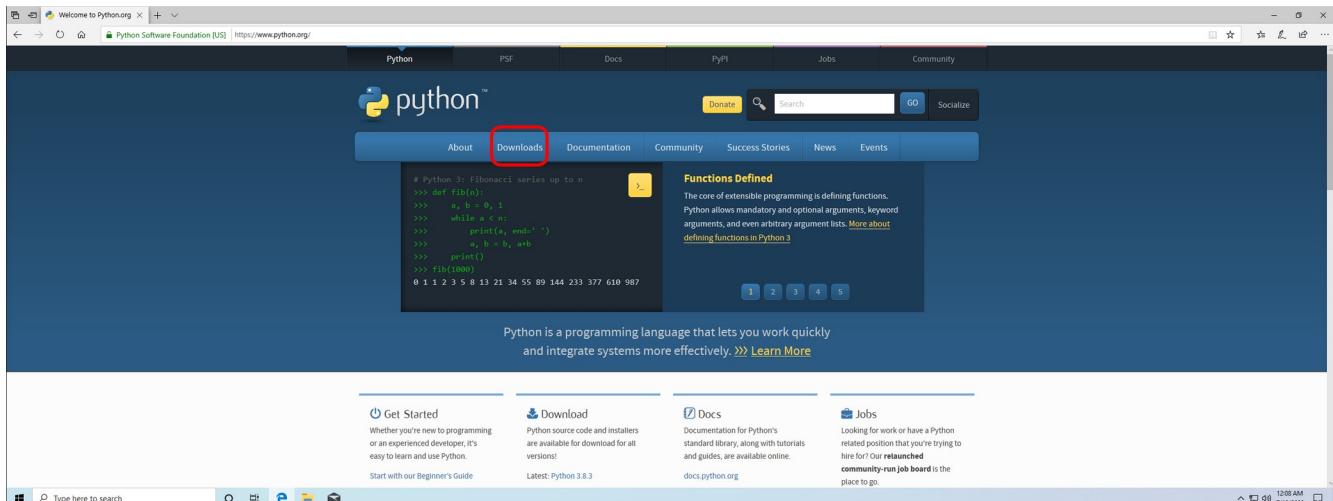


Finally Git will start installing.

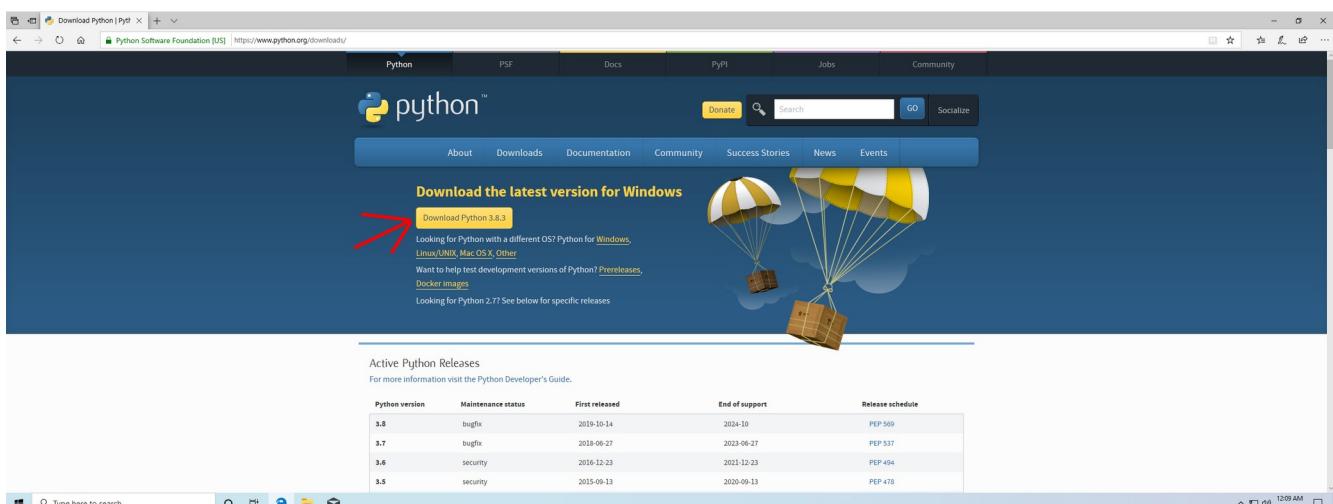


When it's done installing, click “Next” again.

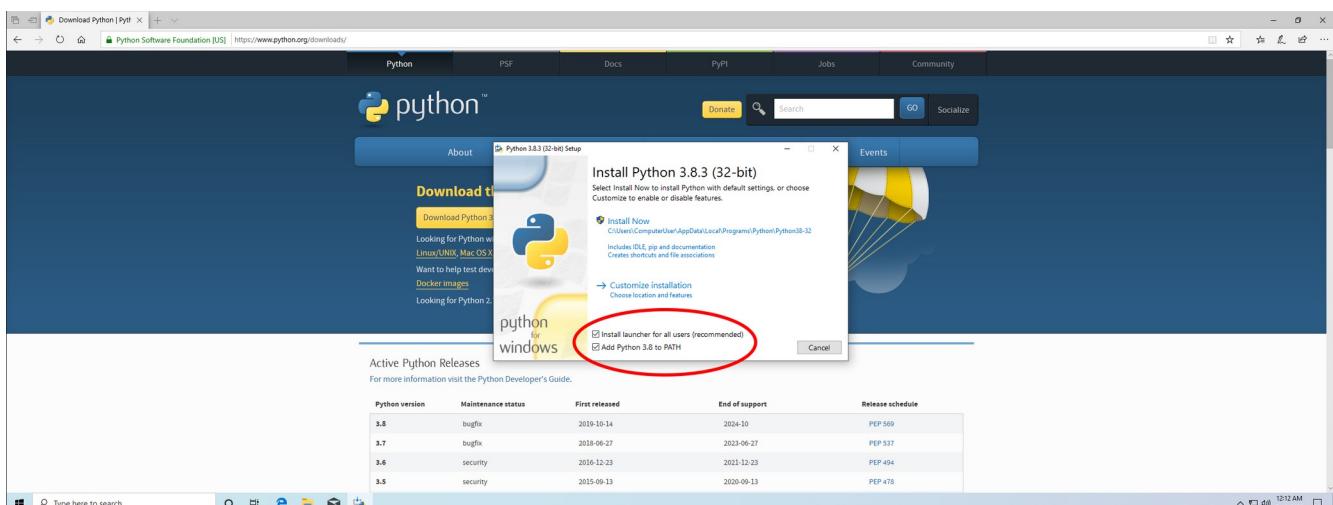
Python



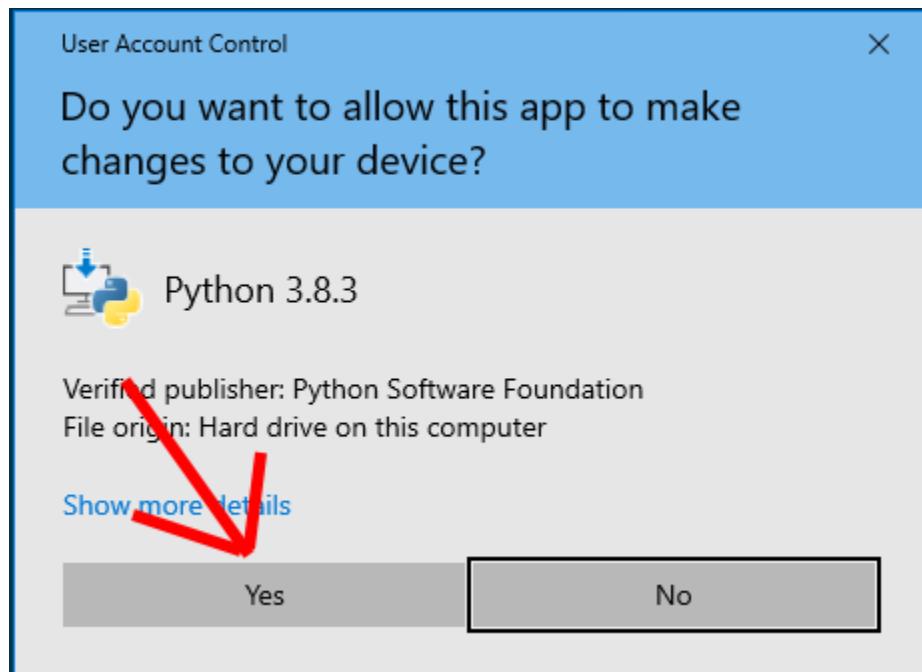
Open python.org in Microsoft Edge and click on “Downloads”.



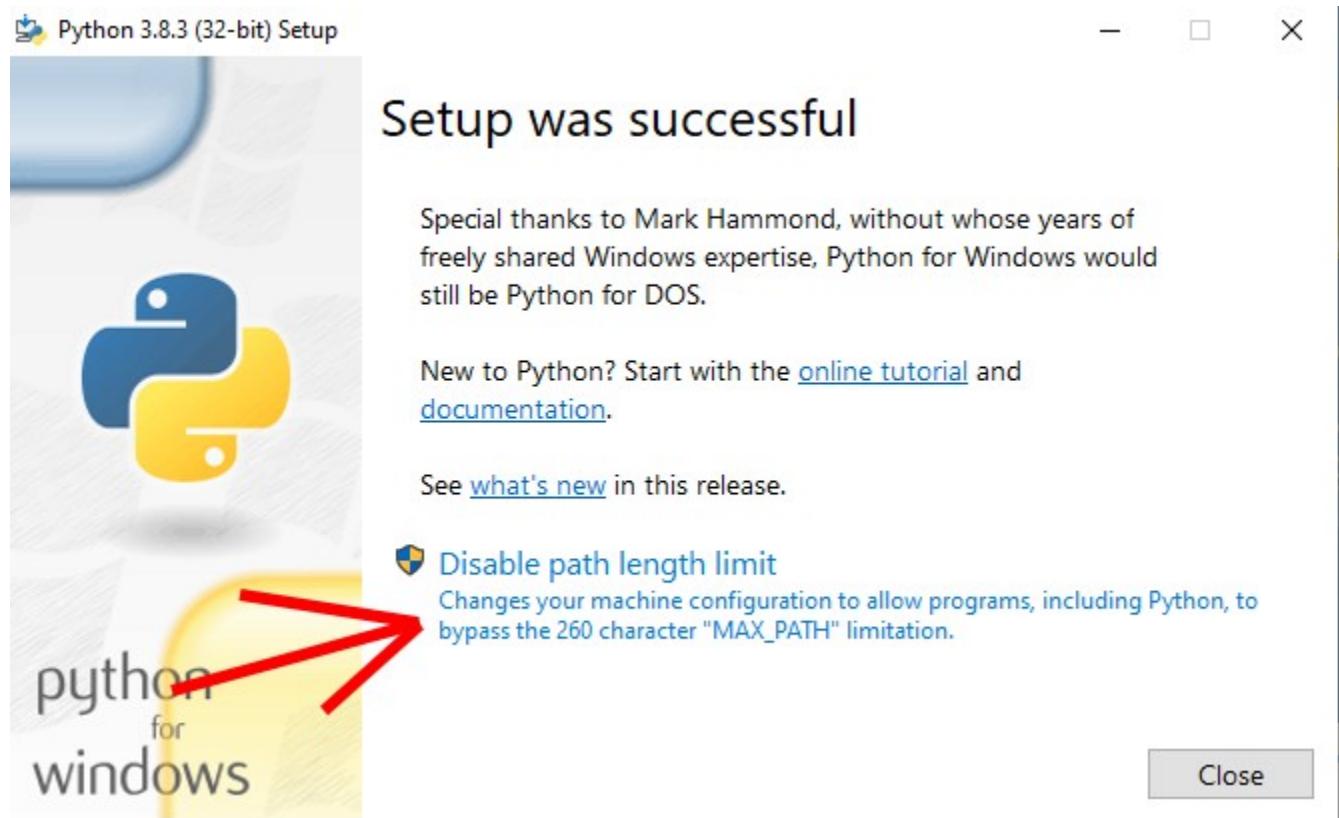
Click the “Download Python 3.8.3”-Button (or any newer version).



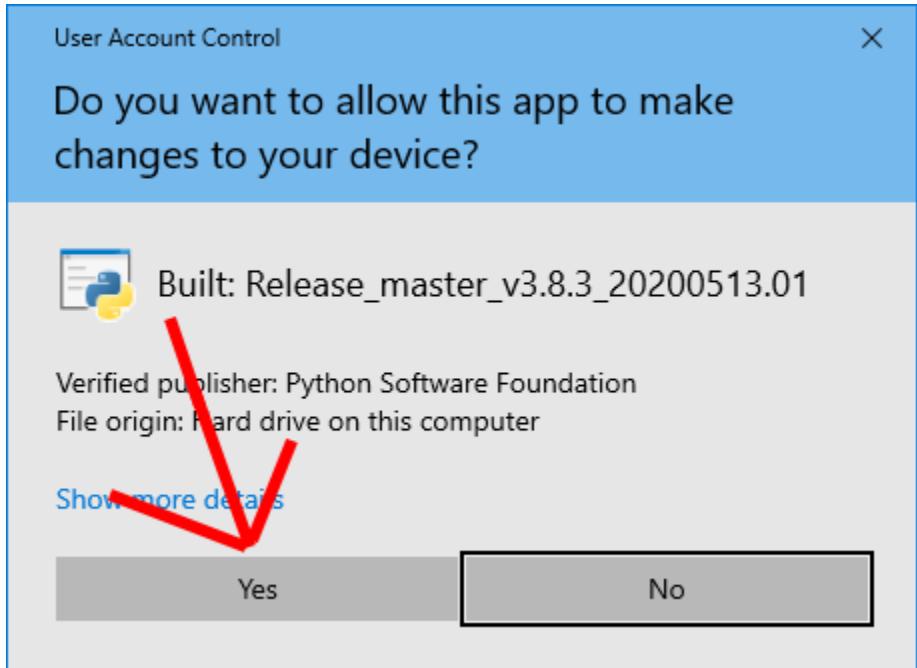
The installer will open. Don't forget to check “Add Python 3.8 to PATH”. Then click “Install Now”.



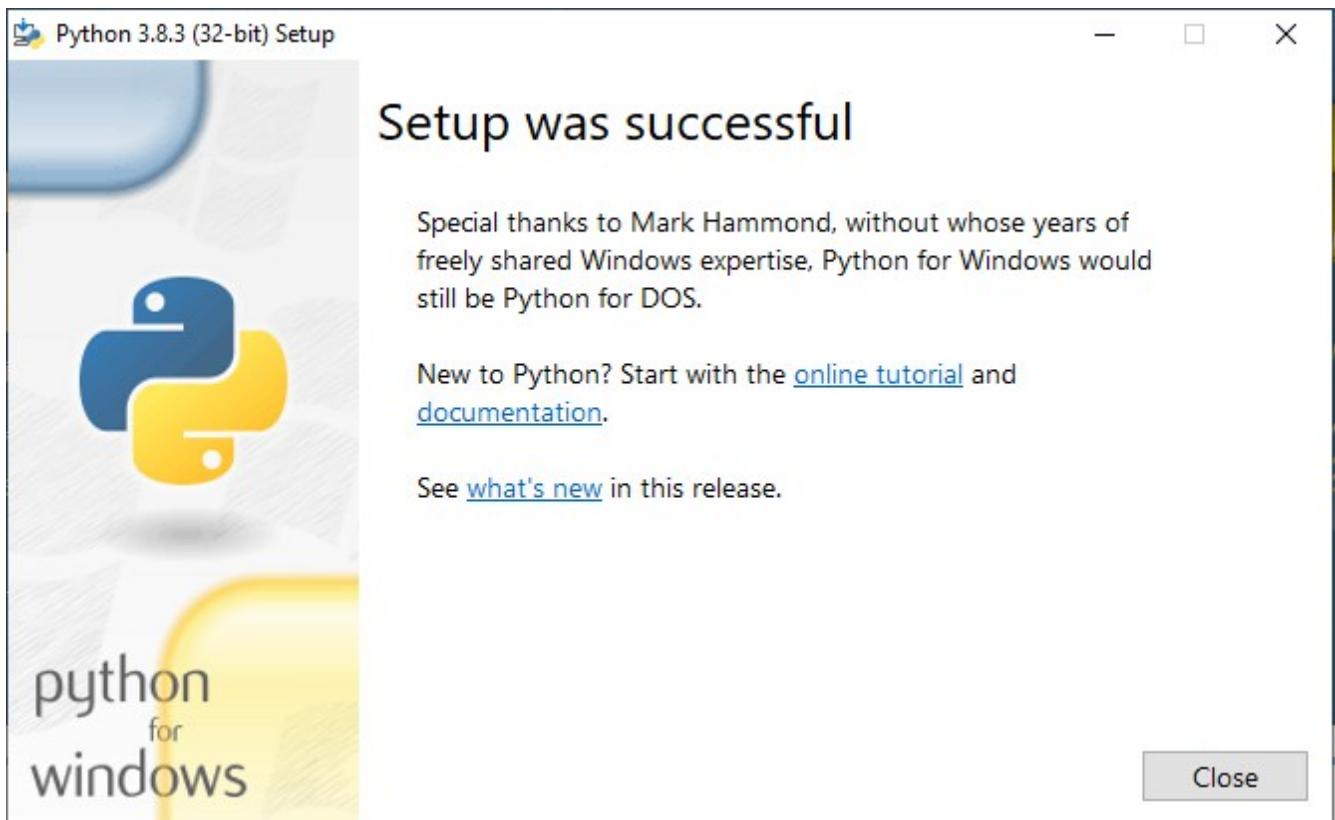
Microsoft Windows will show a UAC prompt. Click “Yes”.



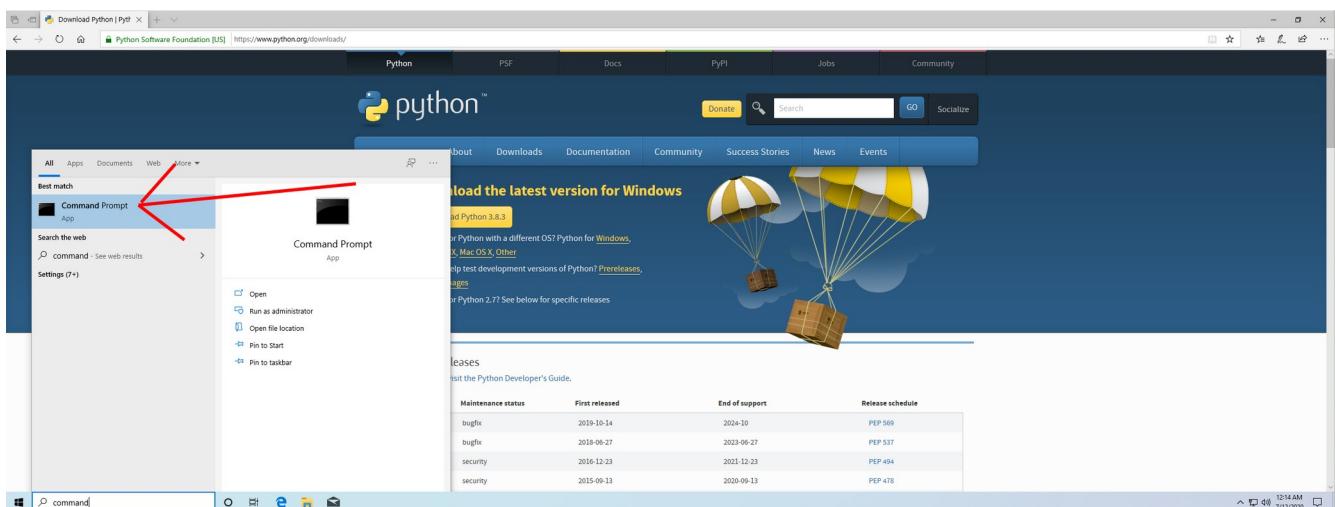
After Python is installed, it is recommended to click “Disable path length limit” to not run into problems with long automatically generated filesystem paths.



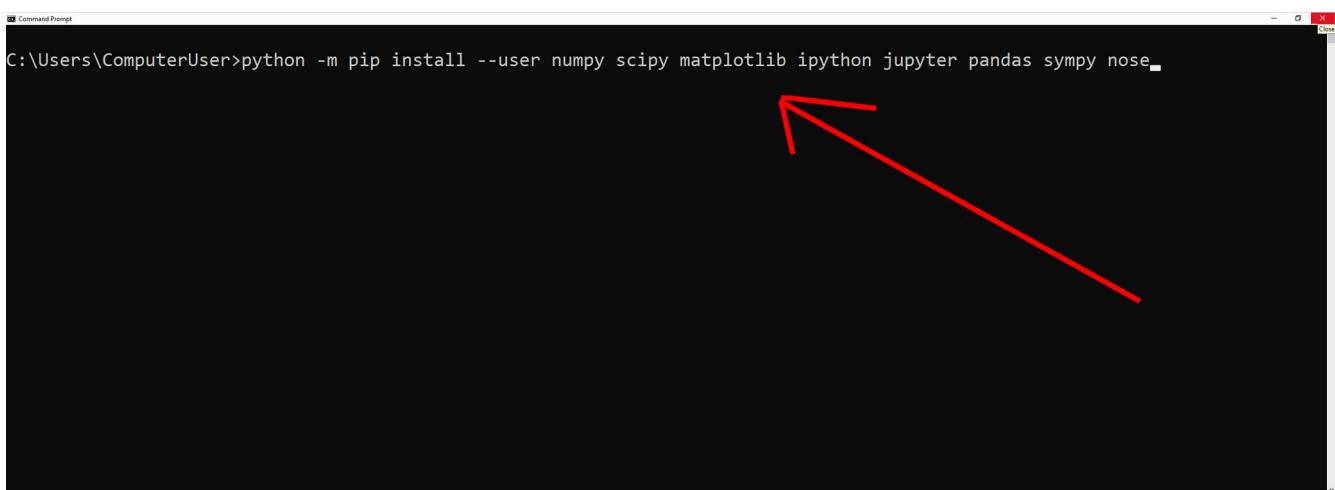
Microsoft Windows will show yet another UAC prompt, again click on “Yes”.



Python is installed, click “Close”.

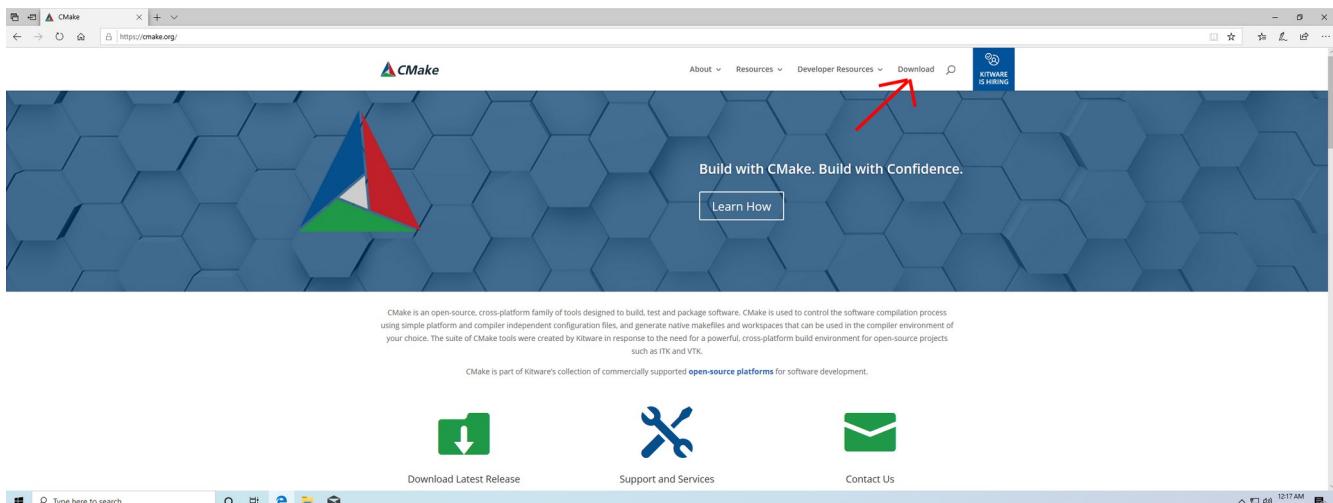


Search for “Command Prompt” in the bottom left search field. Then click on “Command Prompt”.

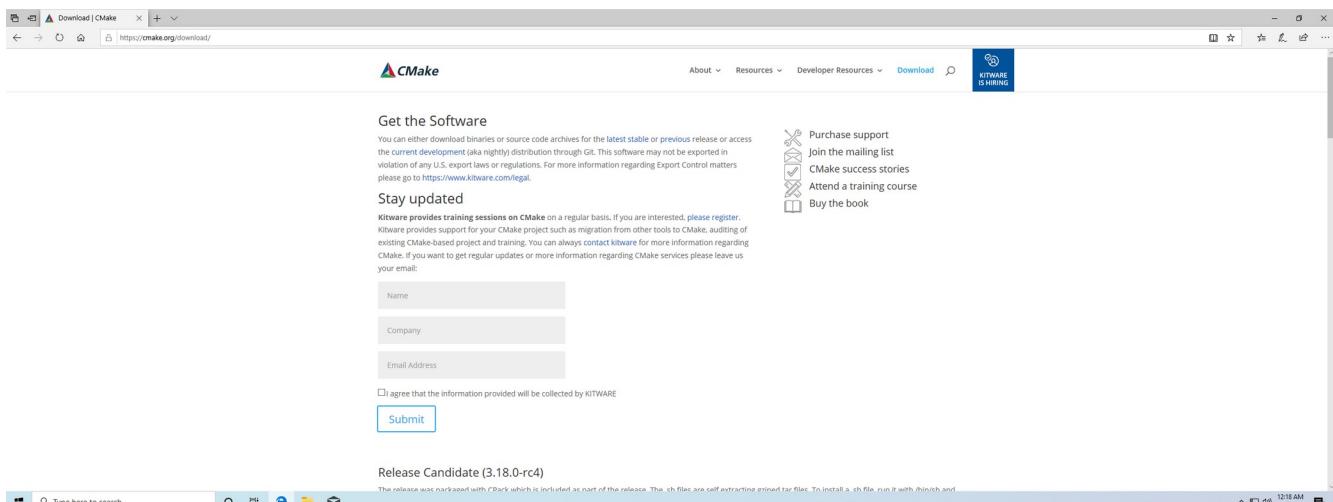


In the Command Prompt enter the following command: python -m pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose

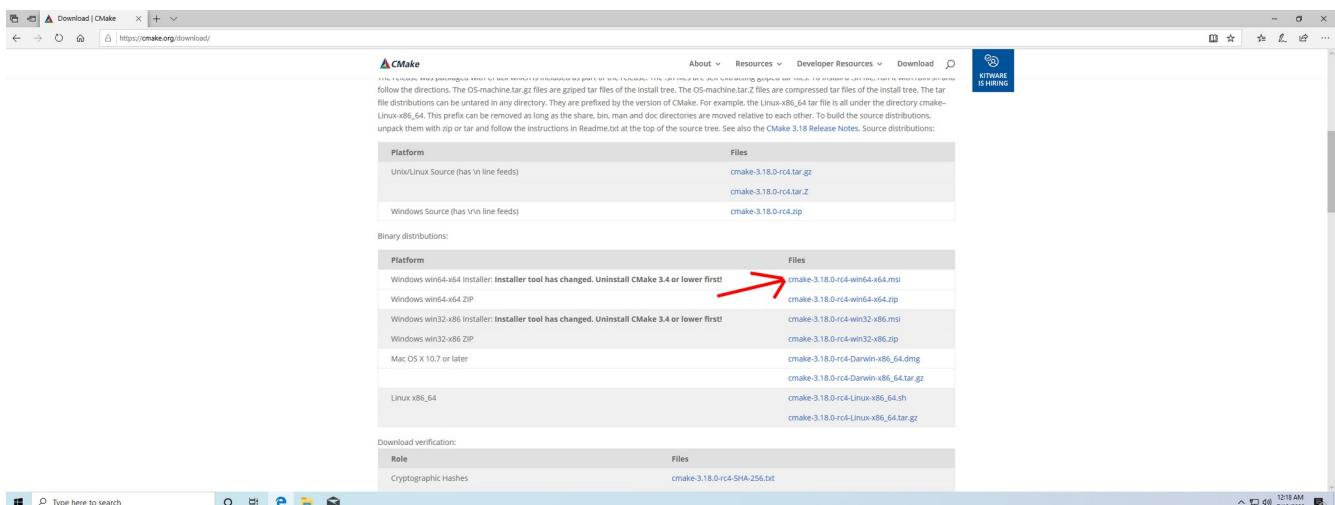
CMake



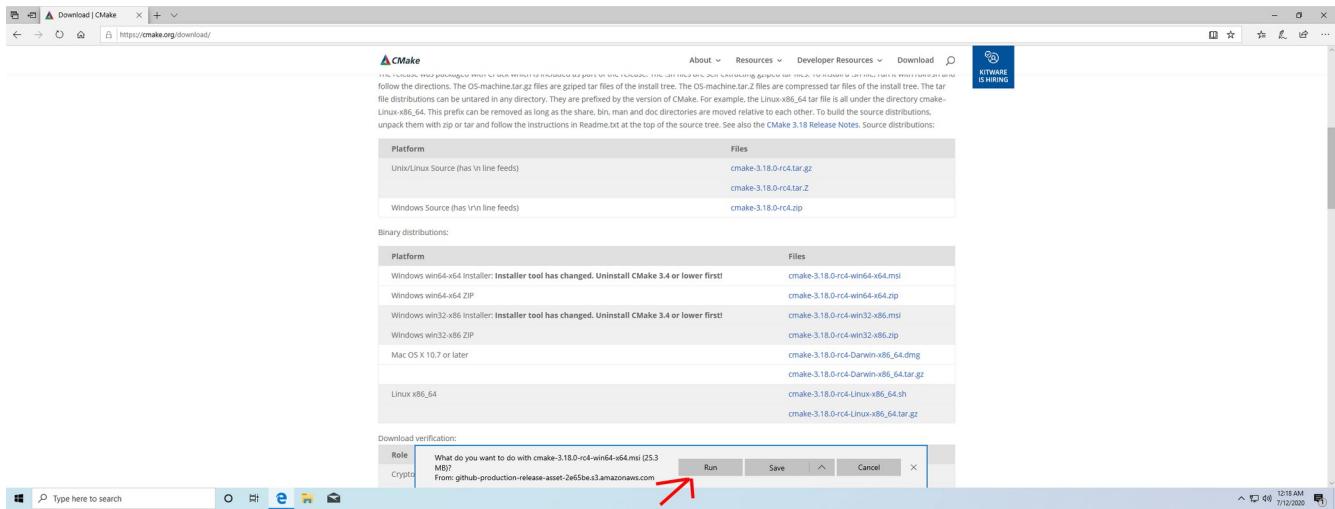
Open cmake.org in Microsoft Edge. Then click on “Download”.



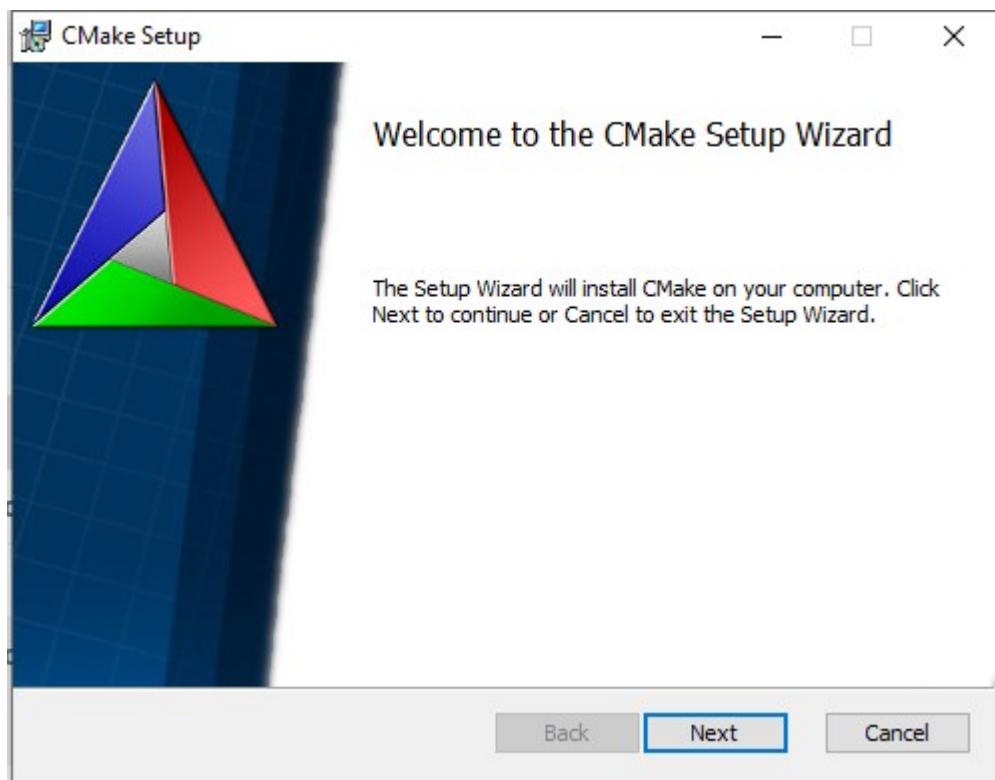
The Download page looks like this, scroll down to the installers.



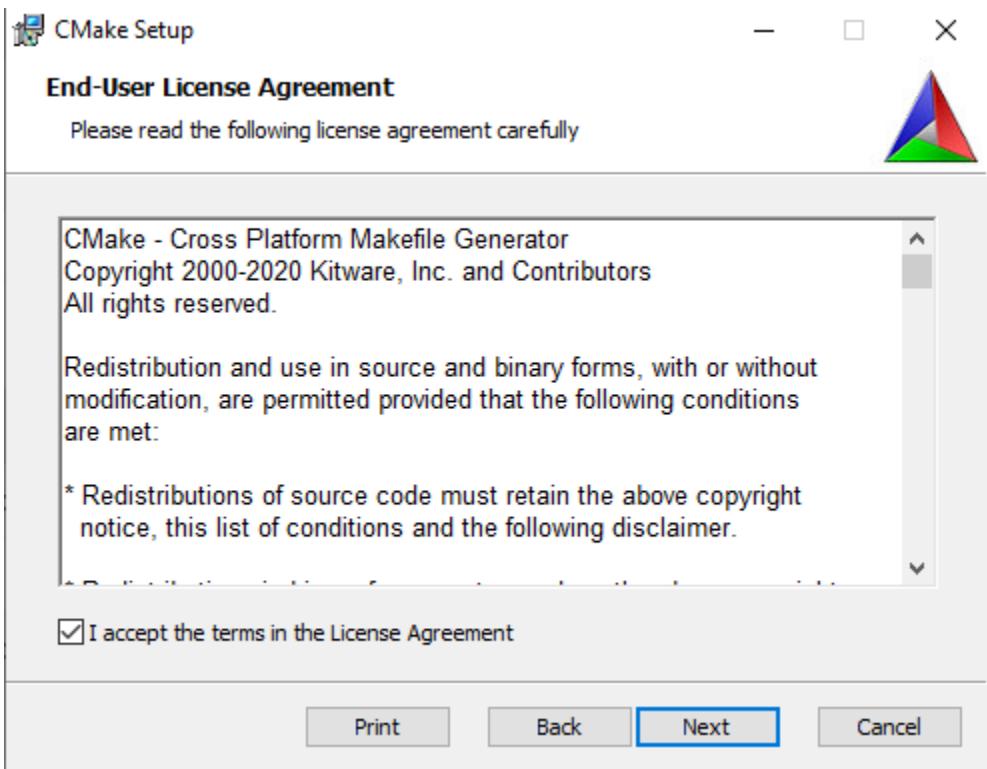
Click on “cmake-3.18.0-rc4-win64-x64.msi”. Any newer version is also fine.



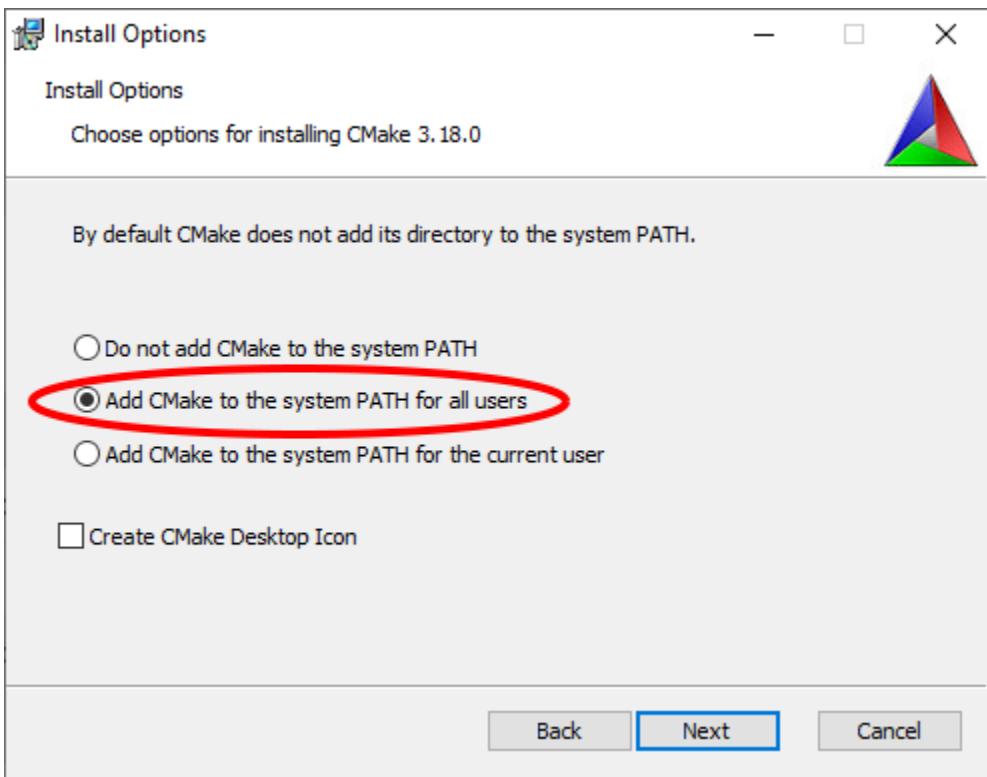
Microsoft Edge will show a prompt, click on “Run”.



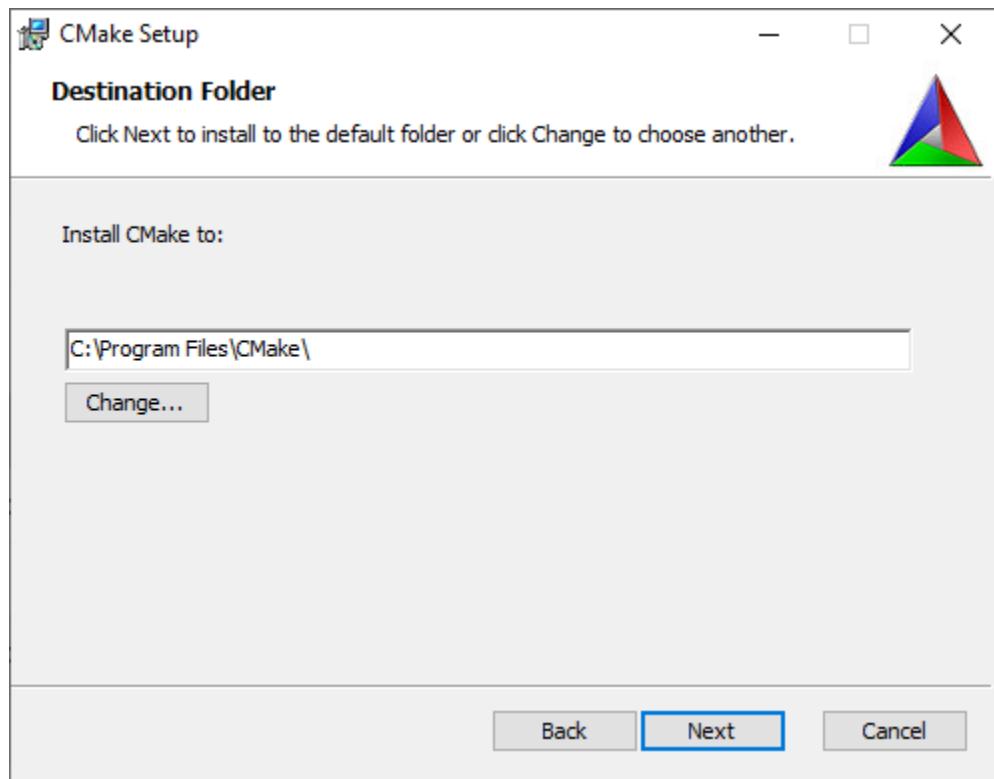
The installer will open, click on “Next”.



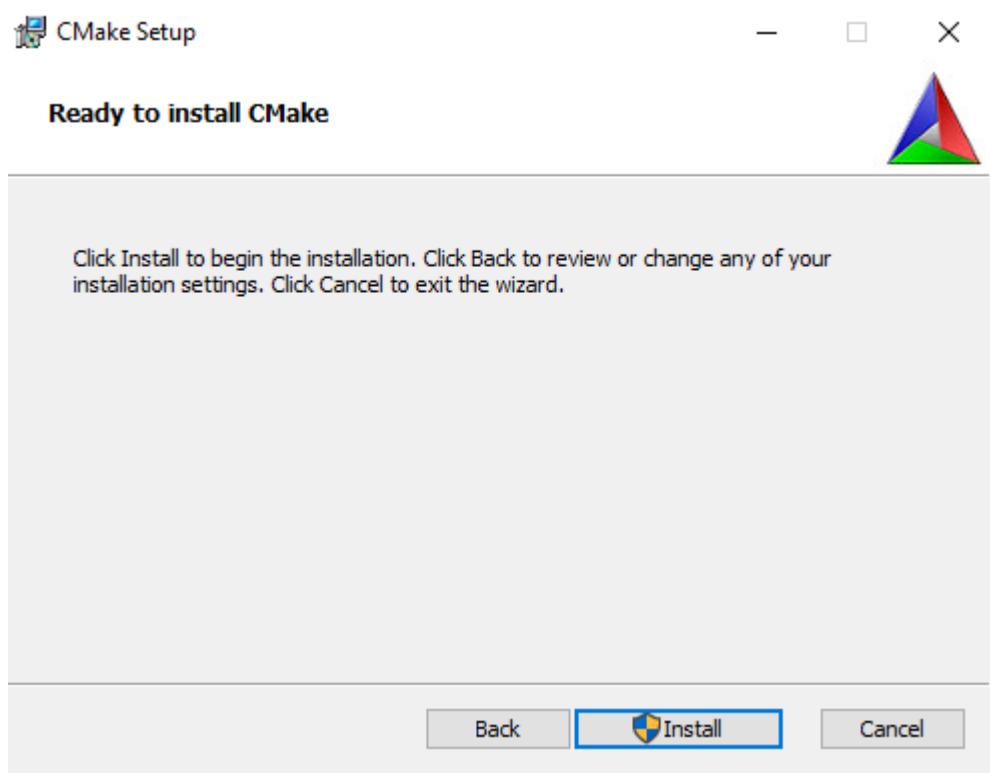
Check the "I accept the terms in the License Agreement" checkbox, then click on "Next".



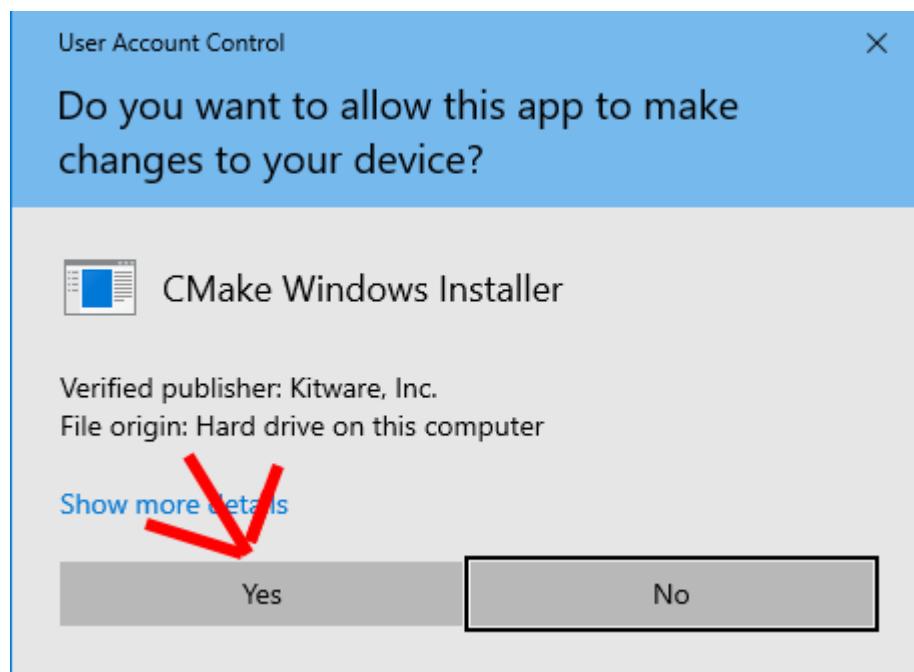
Set the radio button to add CMake to the system PATH, then click "Next".



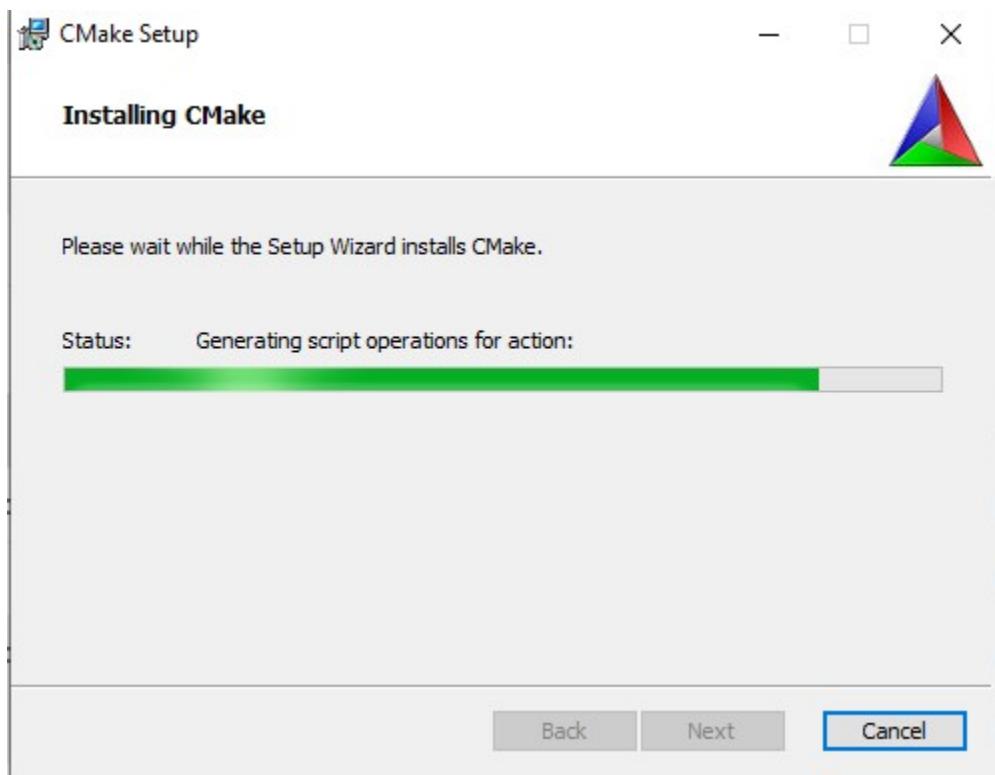
Choose an installation directory, then click “Next”.



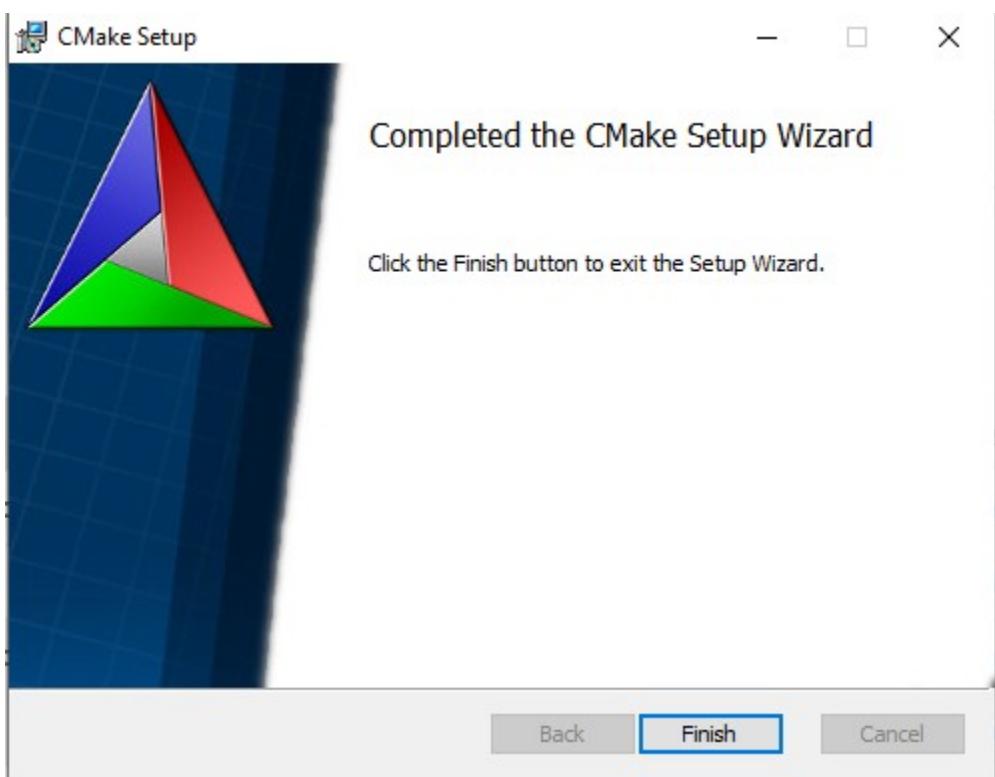
Click "Install".



Microsoft Windows will show a UAC prompt, click "Yes".

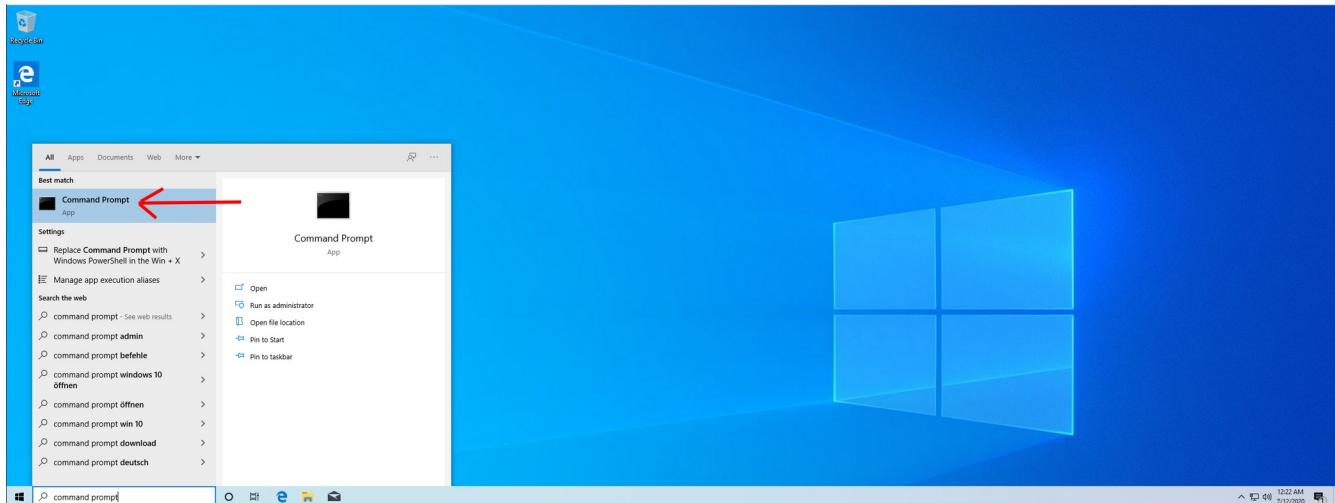


Installation will begin.



Once installation is finished, click on "Finish".

mogasens_csv



Type “Command Prompt” in the search field in the bottom left corner. Then click on “Command Prompt”.

```
Microsoft Windows [Version 10.0.19041.264]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ComputerUser>cd Desktop
```

A screenshot of a Command Prompt window. The window title is "Command Prompt". The text inside shows the Windows version, copyright information, and a command being entered: "C:\Users\ComputerUser>cd Desktop". A red arrow points to the "cd Desktop" command.

Navigate to the Desktop (or any other directory) using the following command: cd Desktop

```
Microsoft Windows [Version 10.0.19041.264]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ComputerUser>cd Desktop

C:\Users\ComputerUser\Desktop>git clone --recursive https://github.com/CppPhil/mogasens_csv.git
```

Clone the Git repository using the following command: git clone --recursive
https://github.com/CppPhil/mogasens_csv.git

```
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 5563 (delta 8), reused 13 (delta 2), pack-reused 5530
Receiving objects: 100% (5563/5563), 1.81 MiB | 651.00 KiB/s, done.
Resolving deltas: 100% (3650/3650), done.
Cloning into 'C:/Users/ComputerUser/Desktop/mogasens_csv/external/scnlib/test/doctest'...
remote: Enumerating objects: 199, done.
remote: Counting objects: 100% (199/199), done.
remote: Compressing objects: 100% (129/129), done.
remote: Total 10293 (delta 81), reused 151 (delta 66), pack-reused 10094
Receiving objects: 100% (10293/10293), 5.83 MiB | 4.61 MiB/s, done.
Resolving deltas: 100% (6623/6623), done.
Submodule path 'external/scnlib/benchmark/google-benchmark': checked out '7ee72863fdb1ccb2af5a011250b56af3f49b7511'
Submodule path 'external/scnlib/test/doctest': checked out 'f13a00cc27ed3c1ec4f755572ab7556c4cb01716'
Submodule path 'external/tl': checked out '3331072339e72f36351edab8d53e824b31ada429'

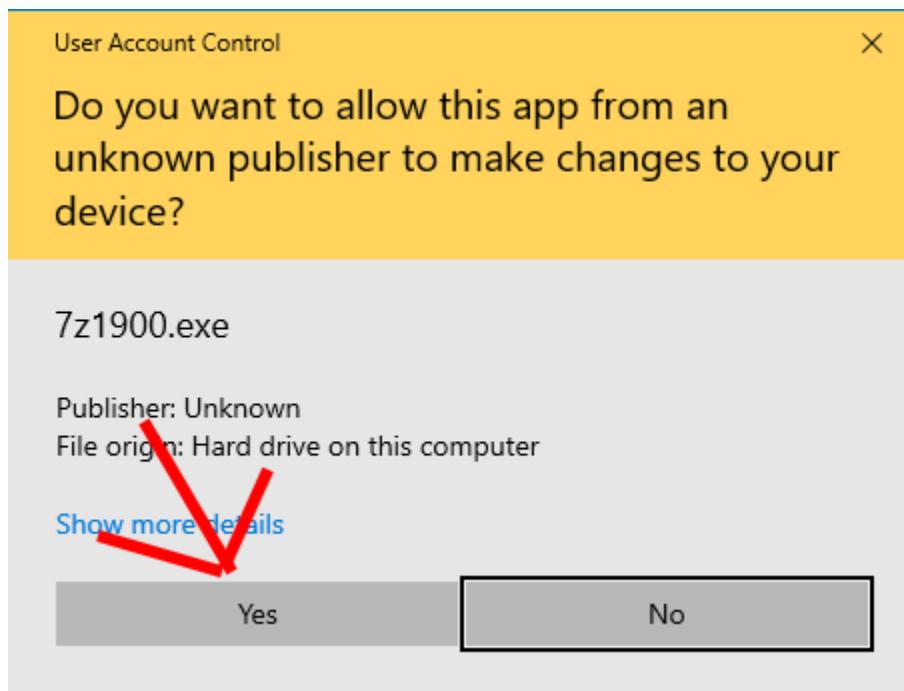
C:\Users\ComputerUser\Desktop>cd mogasens_csv
```

Navigate into the directory using the following command: cd mogasens_csv

```
Command Prompt
remote: Compressing objects: 100% (27/27), done.
remote: Total 5563 (delta 8), reused 13 (delta 2), pack-reused 5530
Receiving objects: 100% (5563/5563), 1.81 MiB | 651.00 KiB/s, done.
Resolving deltas: 100% (3650/3650), done.
Cloning into 'C:/Users/ComputerUser/Desktop/mogasens_csv/external/scnlib/test/doctest'...
remote: Enumerating objects: 199, done.
remote: Counting objects: 100% (199/199), done.
remote: Compressing objects: 100% (129/129), done.
remote: Total 10293 (delta 81), reused 151 (delta 66), pack-reused 10094
Receiving objects: 100% (10293/10293), 5.83 MiB | 4.61 MiB/s, done.
Resolving deltas: 100% (6623/6623), done.
Submodule path 'external/scnlib/benchmark/google-benchmark': checked out '7ee72863fdb1ccb2af5a011250b56af3f49b7511'
Submodule path 'external/scnlib/test/doctest': checked out 'f13a00cc27ed3c1ec4f755572ab7556c4cb01716'
Submodule path 'external/tl': checked out '3331072339e72f36351edab8d53e824b31ada429'

C:\Users\ComputerUser\Desktop>cd mogasens_csv
C:\Users\ComputerUser\Desktop\mogasens_csv>build.bat
```

Start the build process using the following command: build.bat

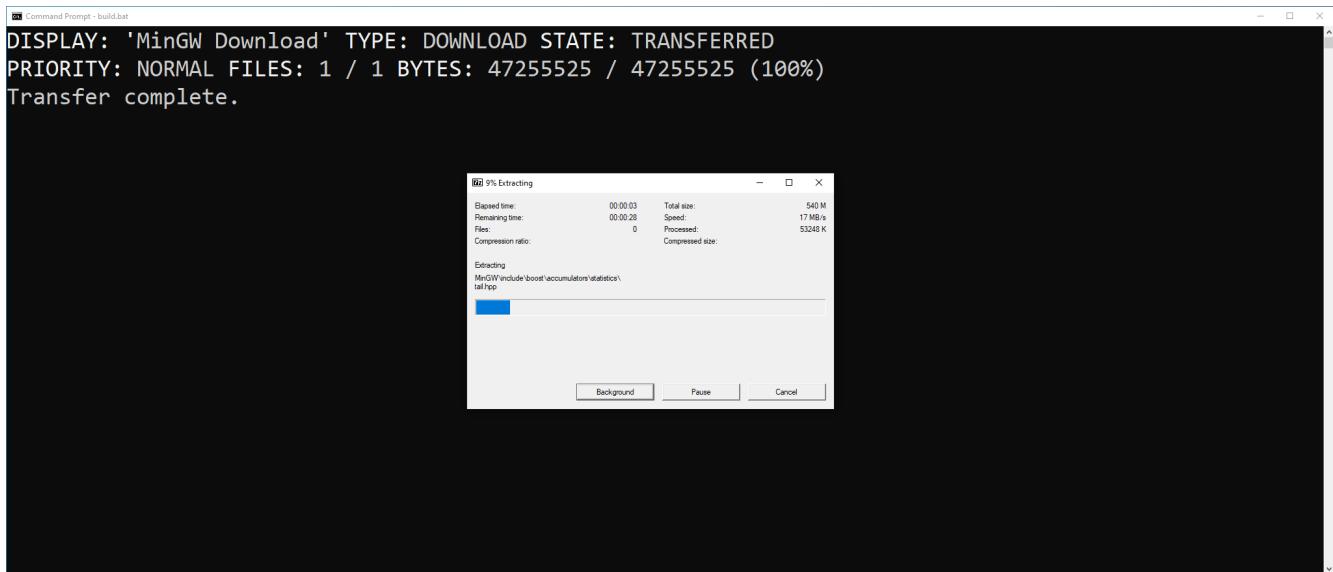


Microsoft Windows will show a UAC prompt for 7z1900.exe. This is the 7z installer of the goto-ruby Git submodule used to extract the portable Ruby distribution that is included. Click on “Yes”.

```
Command Prompt - build.bat
DISPLAY: 'MinGW Download' TYPE: DOWNLOAD STATE: TRANSFERRING
PRIORITY: NORMAL FILES: 0 / 1 BYTES: 2974790 / 47255525 (6%)
TRANSFER RATE: 837.07 KB/S TIME REMAINING: 52 Seconds
```

Next you can see Microsoft BITSADMIN downloading Stephan T. Lavavej's MinGW distribution from nuwen.net

This MinGW distribution includes a port of the GNU C++ Compiler as well as GNU Make so that the application can be built with the CMake generated Makefiles.



The MinGW distribution downloaded is a 7zip self extracting archive. Don't click any of the buttons, just wait for it to finish.

```

[1] Command Prompt - build.bat
DISPLAY: 'MinGW Download' TYPE: DOWNLOAD STATE: TRANSFERRED
PRIORITY: NORMAL FILES: 1 / 1 BYTES: 47255525 / 47255525 (100%)
Transfer complete.
-- The C compiler identification is GNU 9.2.0
-- The CXX compiler identification is GNU 9.2.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Users/ComputerUser/Desktop/mogasens_csv/mingw_dist/MinGW/bin/gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: C:/Users/ComputerUser/Desktop/mogasens_csv/mingw_dist/MinGW/bin/g++.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Success
-- Found Threads: TRUE
-- Version: 7.0.0
-- Build type:
-- CXX_STANDARD: 17
-- Performing Test has_std_17_flag
-- Performing Test has_std_17_flag - Success
-- Performing Test has_std_1z_flag
-- Performing Test has_std_1z_flag - Success
-- Performing Test SUPPORTS_USER_DEFINED_LITERALS

```

Next we can see the build process starting. CMake will generate MinGW Makefiles which will then be used to build the project.

```

[1] Command Prompt
[ 60%] Linking CXX executable feature_extraction_app.exe
[ 60%] Built target gtest_main
Scanning dependencies of target fmc_test
Scanning dependencies of target feature_extraction_test
[ 62%] Building CXX object fix_csv/test/CMakeFiles/fmc_test.dir/main.cpp.obj
[ 65%] Building CXX object fix_csv/test/CMakeFiles/fmc_test.dir/remove_zeros_from_field_test.cpp.obj
[ 67%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/channel_test.cpp.obj
[ 69%] Linking CXX static library ..\..\lib\libgmock.a
[ 72%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/column_test.cpp.obj
[ 72%] Built target fix_mogasens_csv_app
[ 74%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/data_point_test.cpp.obj
[ 76%] Linking CXX executable fmc_test.exe
[ 79%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/error_test.cpp.obj
[ 79%] Built target gmock
[ 81%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/exception_test.cpp.obj
[ 81%] Built target feature_extraction_app
[ 83%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/main.cpp.obj
Scanning dependencies of target gmock_main
[ 86%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/sensor_test.cpp.obj
[ 88%] Building CXX object external/googletest/googlemock/CMakeFiles/gmock_main.dir/src/gmock_main.cc.obj
[ 90%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/to_string_test.cpp.obj
[ 93%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/read_csv_file_test.cpp.obj
[ 95%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/data_set_test.cpp.obj
[ 95%] Built target fmc_test
[ 97%] Linking CXX static library ..\..\lib\libgmock_main.a
[ 97%] Built target gmock_main
[100%] Linking CXX executable feature_extraction_test.exe
[100%] Built target feature_extraction_test

C:\Users\ComputerUser\Desktop\mogasens_csv>

```

After a short while compilation has finished.

```

[1] Command Prompt
[ 60%] Linking CXX executable feature_extraction_app.exe
[ 60%] Built target gtest_main
Scanning dependencies of target fmc_test
Scanning dependencies of target feature_extraction_test
[ 62%] Building CXX object fix_csv/test/CMakeFiles/fmc_test.dir/main.cpp.obj
[ 65%] Building CXX object fix_csv/test/CMakeFiles/fmc_test.dir/remove_zeros_from_field_test.cpp.obj
[ 67%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/channel_test.cpp.obj
[ 69%] Linking CXX static library ..\..\lib\libgmock.a
[ 72%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/column_test.cpp.obj
[ 72%] Built target fix_mogasens_csv_app
[ 74%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/data_point_test.cpp.obj
[ 76%] Linking CXX executable fmc_test.exe
[ 79%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/error_test.cpp.obj
[ 79%] Built target gmock
[ 81%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/exception_test.cpp.obj
[ 81%] Built target feature_extraction_app
[ 83%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/main.cpp.obj
Scanning dependencies of target gmock_main
[ 86%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/sensor_test.cpp.obj
[ 88%] Building CXX object external/googletest/googlemock/CMakeFiles/gmock_main.dir/src/gmock_main.cc.obj
[ 90%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/to_string_test.cpp.obj
[ 93%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/read_csv_file_test.cpp.obj
[ 95%] Building CXX object feature_extraction/test/CMakeFiles/feature_extraction_test.dir/data_set_test.cpp.obj
[ 95%] Built target fmc_test
[ 97%] Linking CXX static library ..\..\lib\libgmock_main.a
[ 97%] Built target gmock_main
[100%] Linking CXX executable feature_extraction_test.exe
[100%] Built target feature_extraction_test

C:\Users\ComputerUser\Desktop\mogasens_csv>fix_csvs.bat

```

We can now generate the adjusted .csv files using the following command: fix_csvs.bat

```
C:\> Command Prompt  
Processing resources/Felix/2020-07-02_13.00.09.csv.  
Successfully wrote data to "resources/Felix/2020-07-02_13.00.09_out.csv".  
  
Processing resources/Marcelle/2020-07-02_14.59.59.csv.  
Successfully wrote data to "resources/Marcelle/2020-07-02_14.59.59_out.csv".  
  
Processing resources/Marcelle/2020-07-02_15.13.22.csv.  
Successfully wrote data to "resources/Marcelle/2020-07-02_15.13.22_out.csv".  
  
Processing resources/Marcelle/2020-07-02_15.31.36.csv.  
Successfully wrote data to "resources/Marcelle/2020-07-02_15.31.36_out.csv".  
  
Processing resources/Marcelle/2020-07-02_15.39.22.csv.  
Successfully wrote data to "resources/Marcelle/2020-07-02_15.39.22_out.csv".  
  
Processing resources/Mike/2020-07-02_14.07.33.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.07.33_out.csv".  
  
Processing resources/Mike/2020-07-02_14.14.32.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.14.32_out.csv".  
  
Processing resources/Mike/2020-07-02_14.20.28.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.20.28_out.csv".  
  
Processing resources/Mike/2020-07-02_14.38.40.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.38.40_out.csv".  
  
C:\Users\ComputerUser\Desktop\mogasens_csv>
```

This is what it looks like after it's done.

As long as we don't delete the fix_csv application in the build directory we can invoke fix_csvs.bat again and again without rebuilding.

```
C:\> Command Prompt  
Processing resources/Felix/2020-07-02_13.00.09.csv.  
Successfully wrote data to "resources/Felix/2020-07-02_13.00.09_out.csv".  
  
Processing resources/Marcelle/2020-07-02_14.59.59.csv.  
Successfully wrote data to "resources/Marcelle/2020-07-02_14.59.59_out.csv".  
  
Processing resources/Marcelle/2020-07-02_15.13.22.csv.  
Successfully wrote data to "resources/Marcelle/2020-07-02_15.13.22_out.csv".  
  
Processing resources/Marcelle/2020-07-02_15.31.36.csv.  
Successfully wrote data to "resources/Marcelle/2020-07-02_15.31.36_out.csv".  
  
Processing resources/Marcelle/2020-07-02_15.39.22.csv.  
Successfully wrote data to "resources/Marcelle/2020-07-02_15.39.22_out.csv".  
  
Processing resources/Mike/2020-07-02_14.07.33.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.07.33_out.csv".  
  
Processing resources/Mike/2020-07-02_14.14.32.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.14.32_out.csv".  
  
Processing resources/Mike/2020-07-02_14.20.28.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.20.28_out.csv".  
  
Processing resources/Mike/2020-07-02_14.38.40.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.38.40_out.csv".  
  
C:\Users\ComputerUser\Desktop\mogasens_csv>generate_images.bat
```

Now we can generate the images from the _out.csv files generated using the following command:
generate_images.bat

```
C:\> Command Prompt - generate_images.bat  
Successfully wrote data to "resources/Mike/2020-07-02_14.14.32_out.csv".  
  
Processing resources/Mike/2020-07-02_14.20.28.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.20.28_out.csv".  
  
Processing resources/Mike/2020-07-02_14.38.40.csv.  
Successfully wrote data to "resources/Mike/2020-07-02_14.38.40_out.csv".  
  
C:\Users\ComputerUser\Desktop\mogasens_csv>generate_images.bat  
Generating images with 6 threads.  
file 1/88  
file 2/88  
file 3/88  
file 4/88  
file 5/88  
file 6/88  
file 7/88  
file 8/88  
file 9/88  
file 10/88  
file 11/88  
file 12/88  
file 13/88  
file 14/88  
file 15/88  
file 16/88  
file 17/88  
file 18/88
```

The images will be generated, this may take a moment.

As long as the _out.csv files are still around we can invoke generate_images.bat again and again without repeating any previous steps

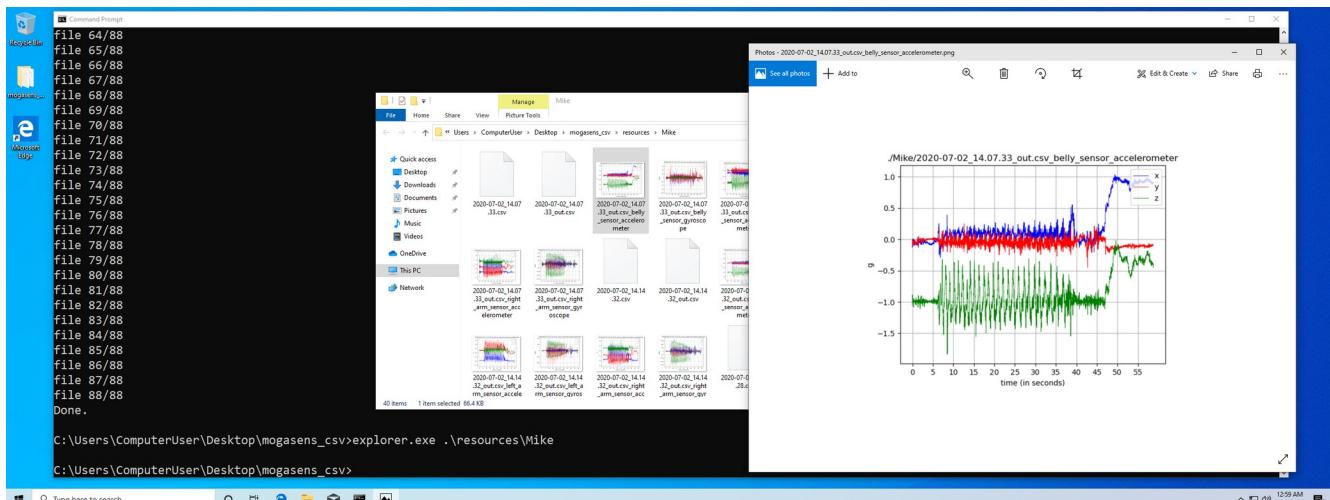
The output has since been changed and will now show percentage values instead.



```
Command Prompt
file 62/88
file 63/88
file 64/88
file 65/88
file 66/88
file 67/88
file 68/88
file 69/88
file 70/88
file 71/88
file 72/88
file 73/88
file 74/88
file 75/88
file 76/88
file 77/88
file 78/88
file 79/88
file 80/88
file 81/88
file 82/88
file 83/88
file 84/88
file 85/88
file 86/88
file 87/88
file 88/88
Done.

C:\Users\ComputerUser\Desktop\mogasens_csv>explorer.exe .\resources\Mike
```

After the images are generated we can view them. Open one of the directories using Windows Explorer, for instance using: explorer.exe .\resources\Mike



Viewing the images

The images will now look differently from the screenshots, because they're wider and generated for every increment of 10 seconds.