

Hyperyon Dual
Stream

Windows Installation Manual



e-con Systems

Your Product Development Partner

Version 1.2

e-con Systems

8/11/2020

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

INTRODUCTION TO OPENCV	3
PREREQUISITES	3
DESCRIPTION	3
BUILDING OPENCV	4
STEP 1 – LAUNCHING CMAKE WINDOW	4
STEP 2 – VISUAL STUDIO VERSION SELECTION	4
STEP 3 – CONFIGURE AND GENERATE CMAKE	5
STEP 4 – VIDEOIO FILE REPLACE	6
STEP 5 – BUILDING BASECLASSES	6
STEP 6 – BUILD OPENCV IN VISUAL STUDIO	9
BUILDING SAMPLE CODE	10
TROUBLESHOOTING	13
SUPPORT	14

Introduction to OpenCV

Open Source Computer Vision Library (OpenCV) is an open source computer vision and machine learning software library. OpenCV libraries are used to communicate with Cameras. APIs introduced in the OpenCV can be supported with e-con Systems Hyperyon Dual Stream cameras.

This document helps you to install OpenCV in Windows and build a sample code to access the Hyperyon Dual Stream camera with OpenCV.

Prerequisites

The prerequisites are as follows:

- Click here(<https://cmake.org/download/>) to download CMake.
- Download OpenCV from here(<https://github.com/opencv/opencv>).

Using git

Click on code option and copy the URL

```
$ git clone <OpenCV_URL>
$ cd opencv
$ git checkout <opencv_version(3.3.1 or 3.4.1)>
```

Direct Download

For 3.3.1 (<https://github.com/opencv/opencv/archive/3.3.1.zip>),

For 3.4.1 (<https://github.com/opencv/opencv/archive/3.4.1.zip>).

- Create a source directory in the opencv folder and move all the files to the source folder
- Create a build directory in the opencv/
- Build OpenCV in your PC using Visual Studio 2017.

Description

The following steps have been tested on Windows 10. OpenCV must work on any other relatively modern version of Windows OS.

Building OpenCV

OpenCV is a sample command line application used to demonstrate some of the features of the e-con Systems cameras with OpenCV APIs.

- Step 1. [Launching CMake Window](#)
- Step 2. [Visual Studio Version Selection](#)
- Step 3. [Configure and Generate CMake](#)
- Step 4. [Replace Videoio File](#)
- Step 5. [Building Base Class Libraries](#)
- Step 6. [Build OpenCV](#)

Step 1 – Launching CMake Window

In CMake window, select the OpenCV sources as source folder and OpenCV/build as build folder and click **Configure** button.

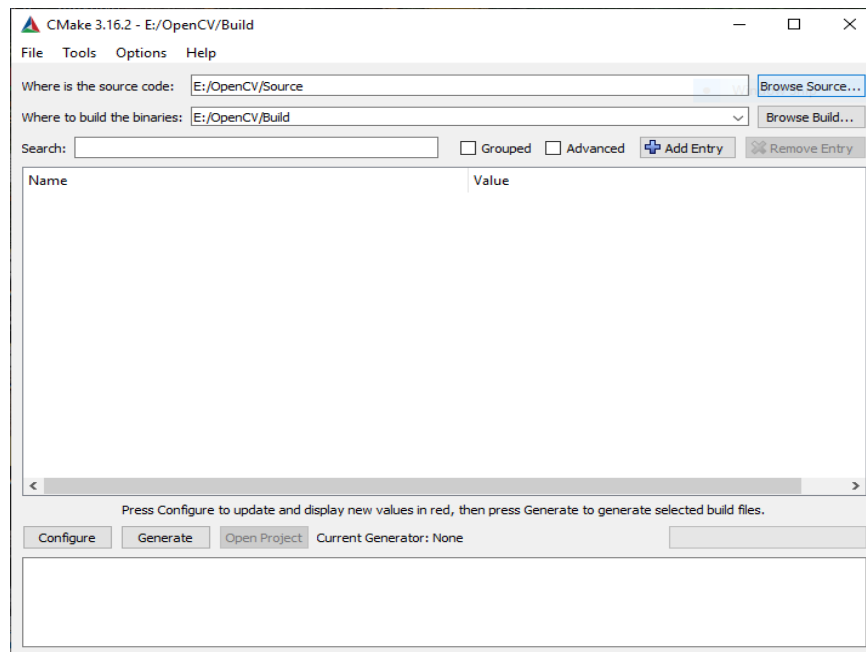


Figure 1: CMake Source and Build Directory Specification Window

Step 2 – Visual Studio Version Selection

A window prompting to select current Visual Studio version (VS2017) in your PC and x32 and x64 version appears. Select the appropriate options as shown below.

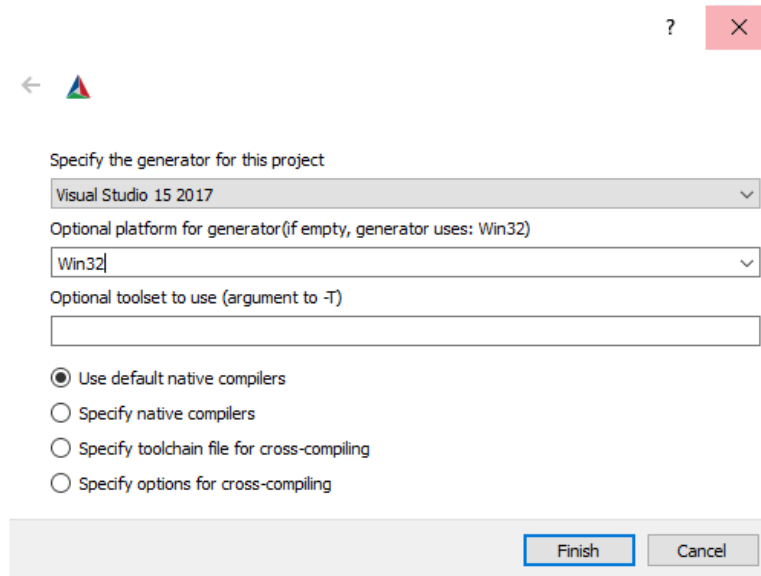
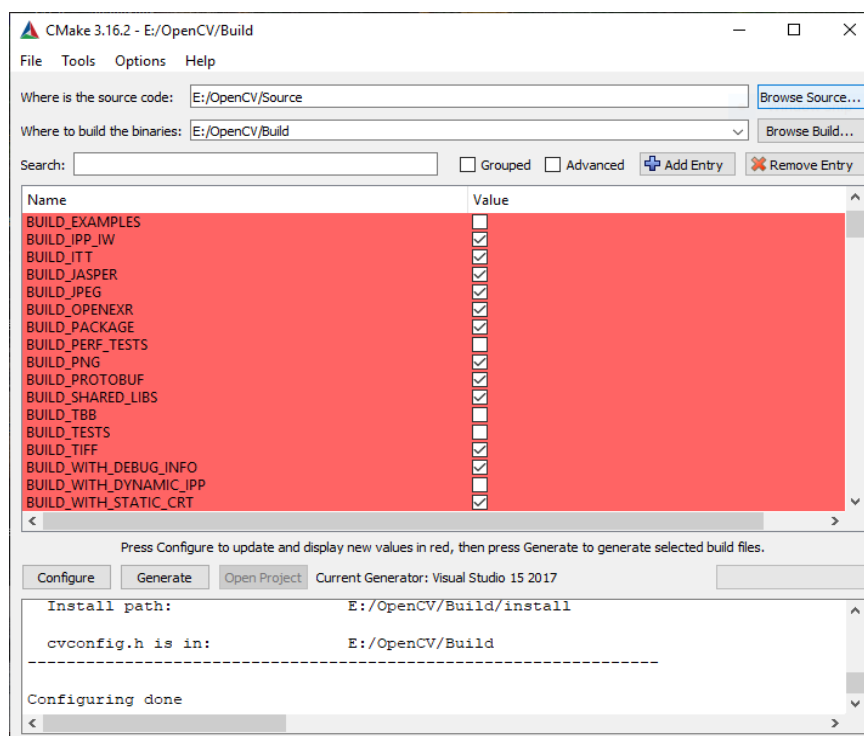


Figure 2: Visual Studio Version selection in CMake

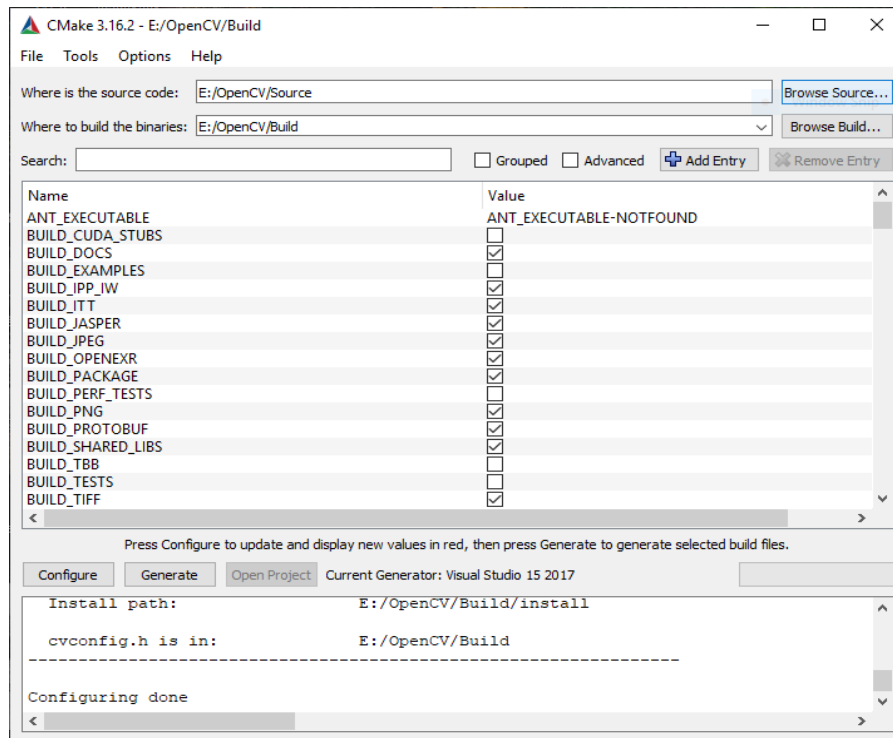
Step 3 – Configure and Generate CMake

The steps to configure and generate CMake are as follows:

1. Click **Configure** after selecting the Visual Studio.



2. Uncheck **BUILD_PERF_TESTS** and **BUILD_TESTS**.
3. Check the **BUILD_opencv_world** to build single binary, including all the modules instead of a collection of separate binaries.
4. Click **Configure** till all red flag goes off.



5. Click **Generate** to create Visual Studio solution file in the OpenCV build directory.

Step 4 – Videoio File Replace

Replace the **videoio** folder with the folder downloaded from the e-con's github (<https://github.com/econsystems/HyperyonDualStream/tree/master/source>) with **OpenCV/Sources/modules/** location.

Step 5 – Building BaseClasses

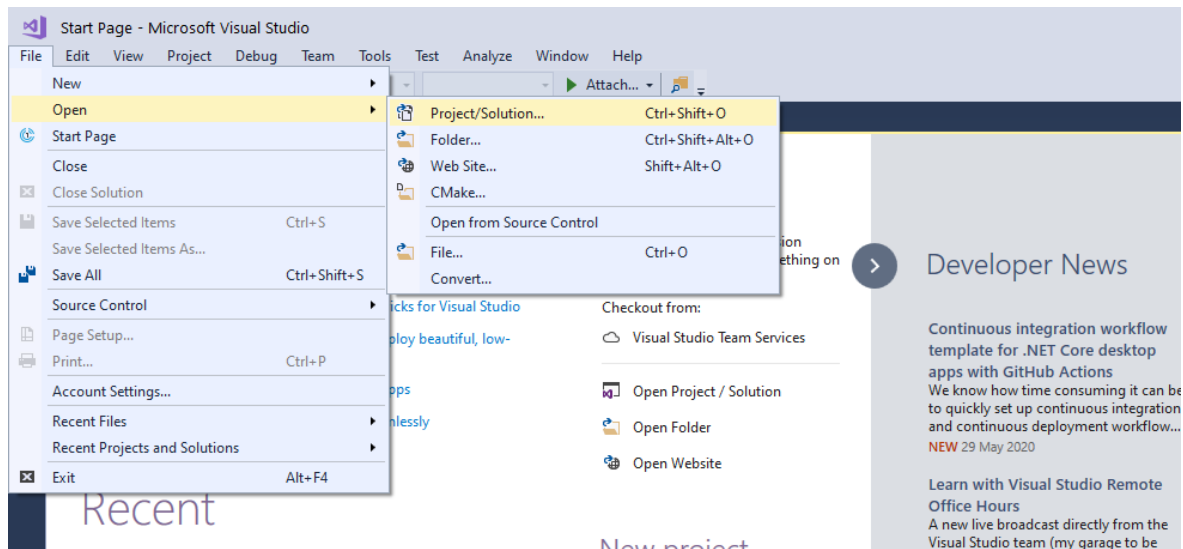
Note: Provided the BaseClasses library files in the package you can make use of it for building OpenCV.

(<https://github.com/econsystems/HyperyonDualStream/tree/master/Binary/Lib>)

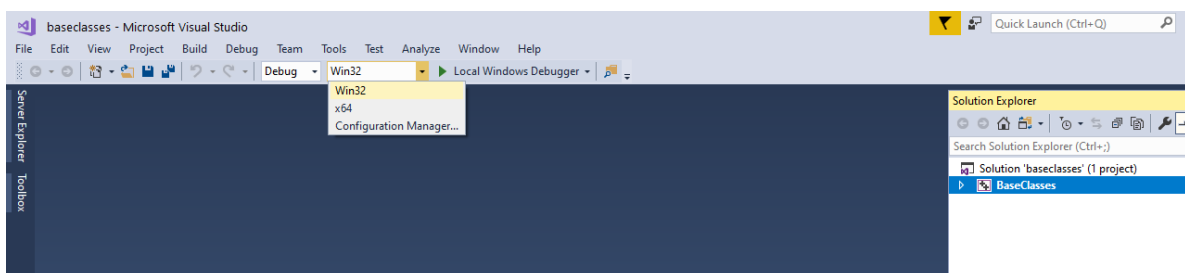
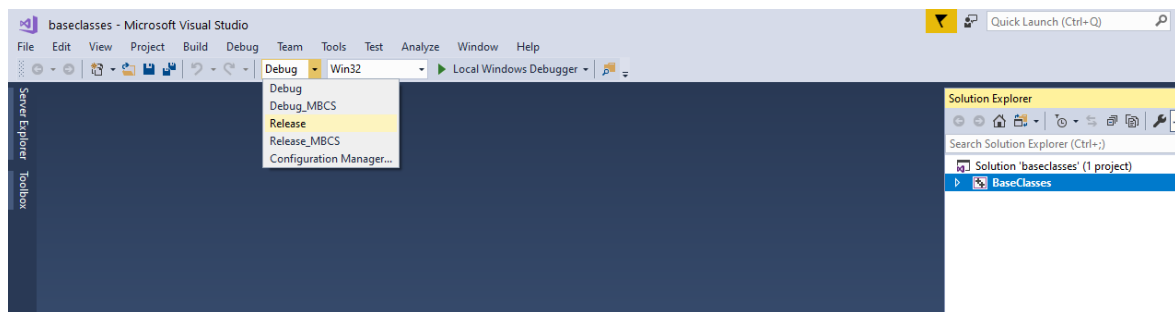
Download BaseClasses project e-con's github

(<https://github.com/econsystems/HyperyonDualStream/tree/master/source>)

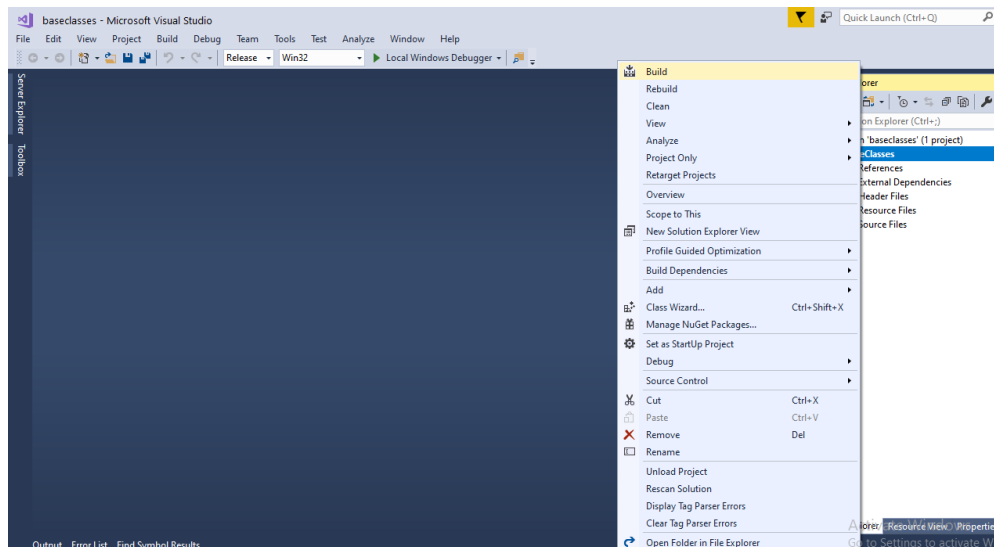
1. Open and Configuring the **baseclasses** project:
 - a) Open the new instance of visual studio (2017).
 - b) Click **File-> Open -> Project/Solution**



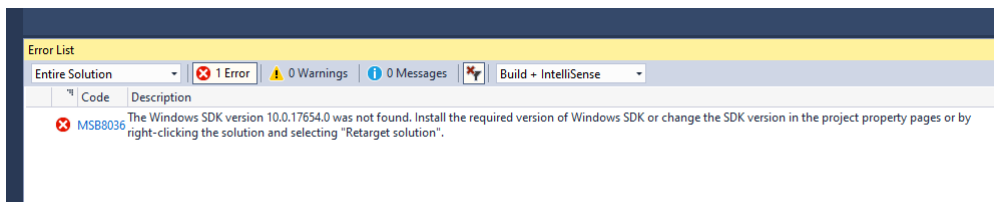
- c) Browse the baseclasses project and select **baseclasses.sln**.
- d) Choose Solution configuration (Debug / Release) and Solution Platform (Win32 or x64) (based on your requirement).



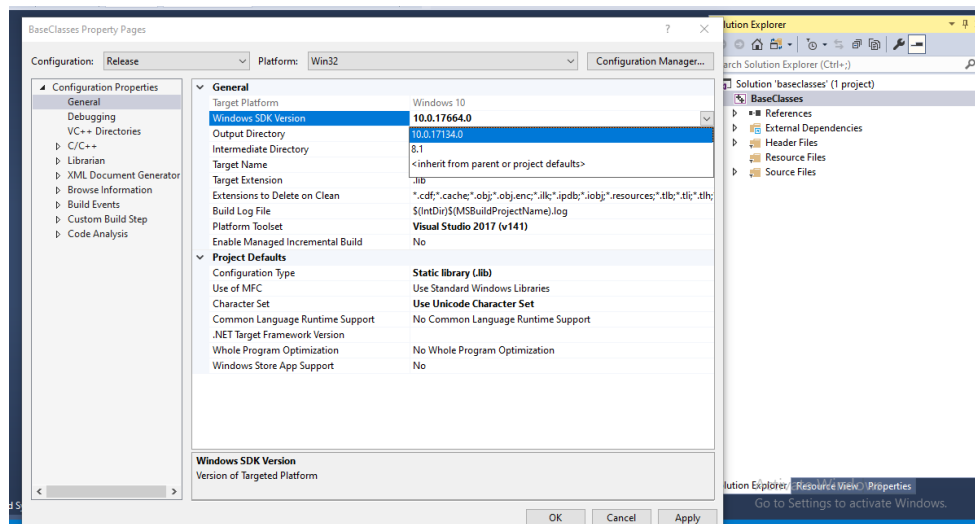
2. Build the baseclasses project:
 - a) Give right click on baseclasses solution and select Build Solution (or Rebuild Solution).



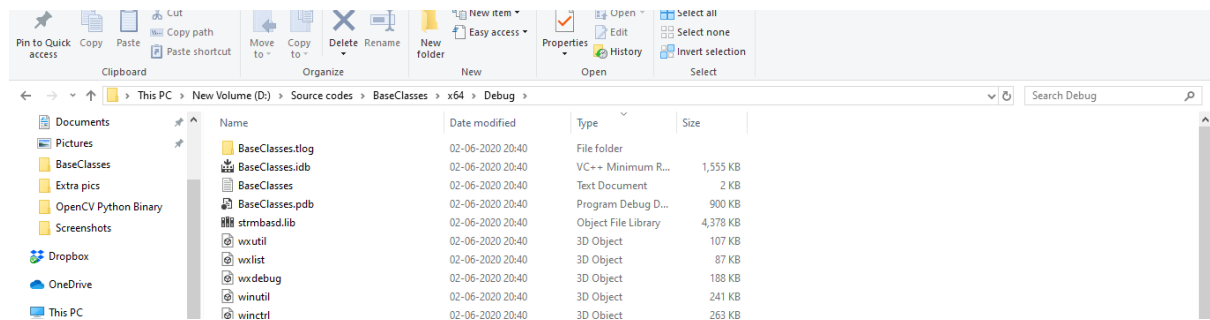
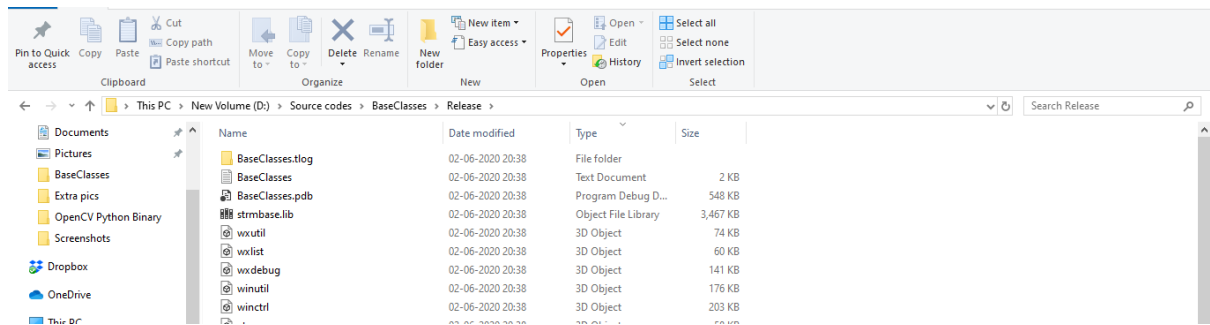
- b) While building the solution, if u get the following error you change the Configuration Properties.



- c) **BaseClasses -> Properties -> Configuration Properties -> General -> Windows SDK Version** choose available SDK version.



- d) Build the solution.
e) After Building you can find the strmbase.lib (for Release), strmbasd.lib (for Debug).



f) The **strmbase** / **strmbasd** lib has to be linked in the OpenCV Project.

Step 6 – Build OpenCV in Visual Studio

The steps to build OpenCV in Visual Studio are as follows:

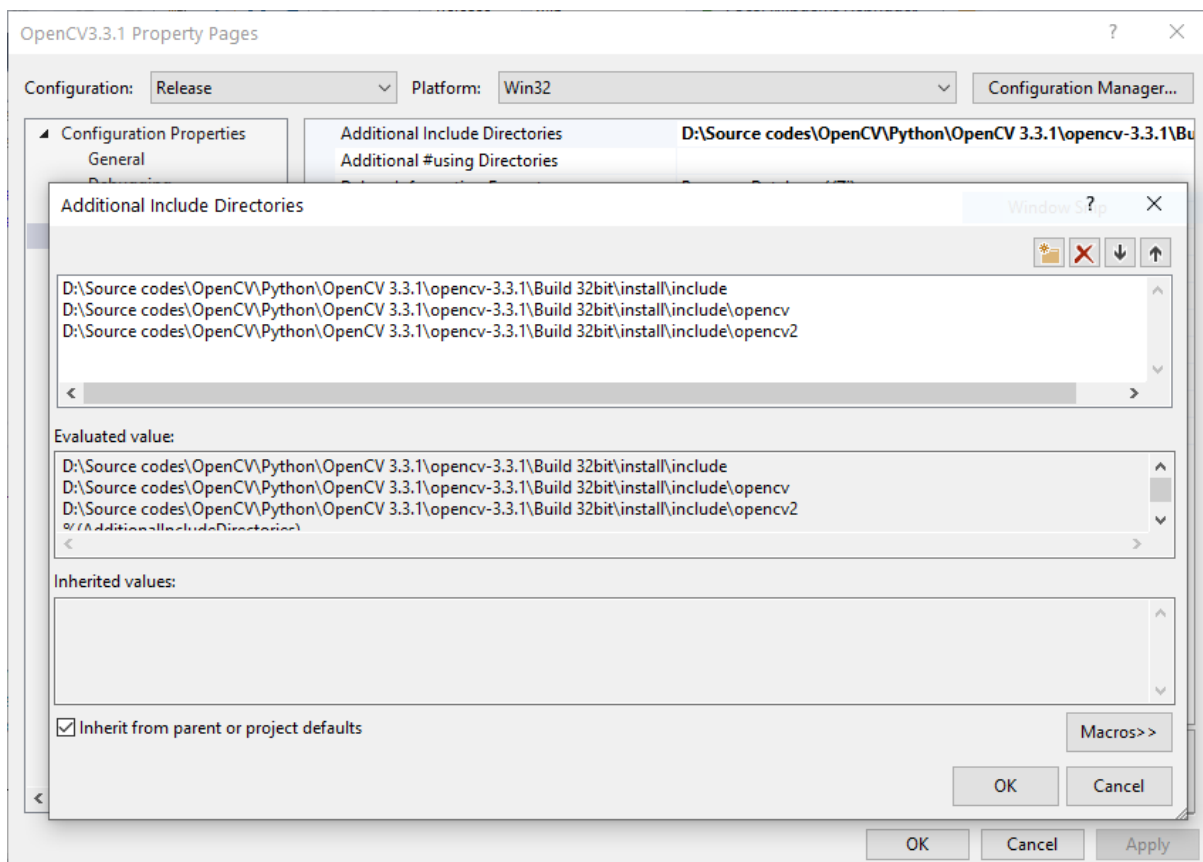
1. Run the **OpenCV.sln** found in the build directory of OpenCV using Visual Studio.
2. Add **setupapi.lib** in modules/opencv_world properties tab under Linker > Input> Additional Dependencies.
3. Add **strmbase.lib** for **release** mode and **strmbasd.lib** for **debug** mode from the **BaseClasses** solution directory, in modules/opencv_world properties tab under **Linker->General->Additional Library directories** and mention the lib name in **Linker->Input->Additional dependencies**.
4. Build **CMakeTargets/All Build** and **CMakeTargets/Install** separately in both the Debug/Release Configuration of the Visual Studio.

Building Sample Code

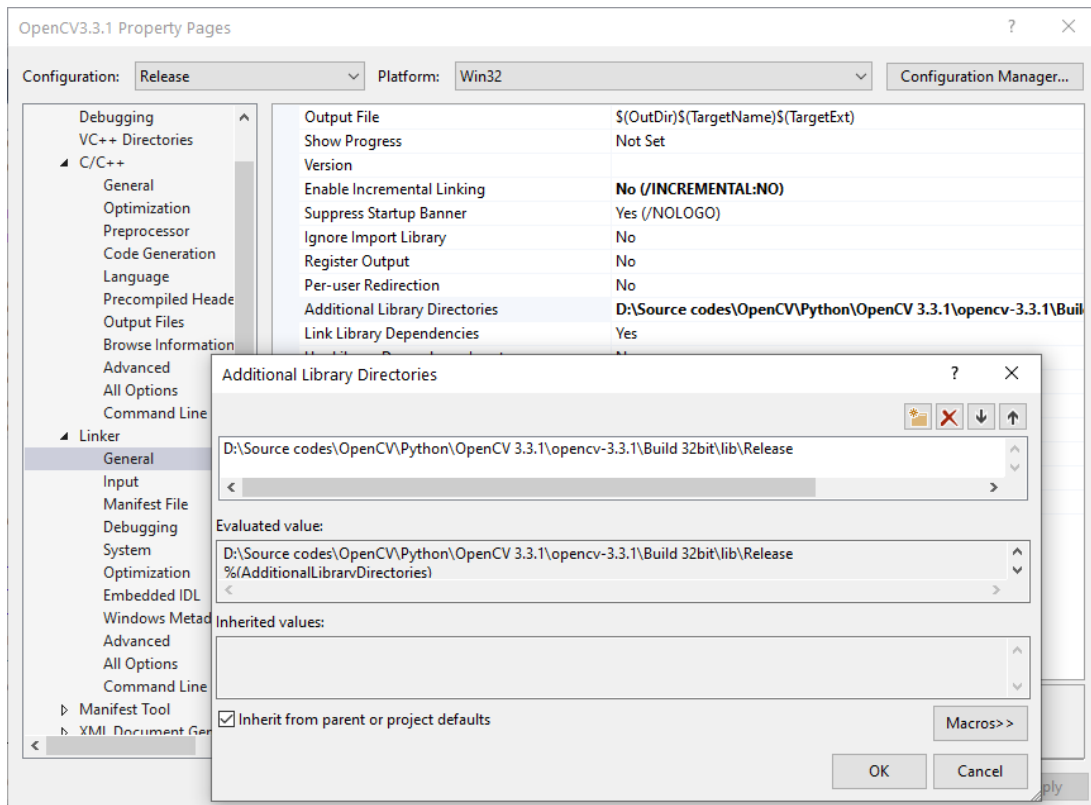
This section describes about how to build the sample code.

The steps to build the sample code are as follows:

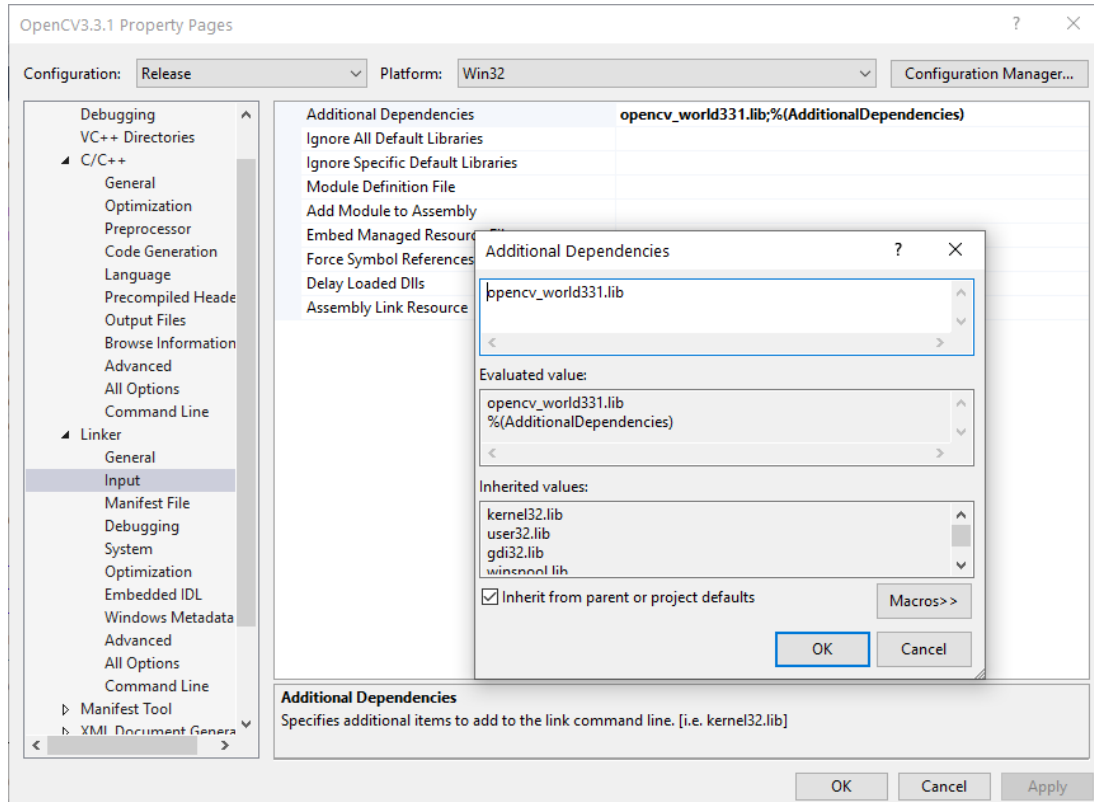
1. Create a new console application in Visual Studio. Add the *.cpp file of the application to be built from the blog
(<https://github.com/econsystems/HypervonDualStream/tree/master/source/HypervonCam>).
2. Change **Application -> Configuration Properties -> General -> project Defaults -> Character Set -> Use Unicode Character Set**
3. Link the OpenCV header under **C/C++ -> General -> Additional Include Directories** files with the following:
 - OpenCV/build/install/include
 - OpenCV/build/install/include/opencv
 - OpenCV/build/install/include/opencv2



4. Link the library files of OpenCV from the **OpenCV/build/lib/Release** for configuration type release under **Linker > General > Additional Library Directories**.



5. List all the library names linked to the project under **Linker > Input > Additional Dependencies** in the Visual Studio Project Property page.



6. Add runtime libraries in the sample application **root** folder from the **OpenCV/build/bin/Debug** or **OpenCV/build/bin/Release** based on the configuration of the sample application.

The runtime libraries for **OpenCV release version 3.3.1** are:

- opencv_world331.dll
- eCAMFwExt.dll
- MFTopologydll.dll

and the runtime libraries for **OpenCV Debug version 3.3.1** are:

- opencv_world331d.dll
- eCAMFwExt.dll
- MFTopologydll.dll

The runtime libraries for **OpenCV release version 3.4.1** are:

- opencv_world341.dll
- eCAMFwExt.dll
- MFTopologydll.dll

and the runtime libraries for **OpenCV Debug version 3.4.1** are:

- opencv_world341d.dll
- eCAMFwExt.dll
- MFTopologydll.dll

- Run the OpenCVCam.exe application using **Administrator** Mode.

Troubleshooting

In this section, you can view the list of commonly occurring issues and their troubleshooting steps.

Linker issues relating to setupdi* while building.

Add **setupapi.lib** in the modules/opencv_world properties tab under **Linker> Input > Additional dependencies**.

There is no install folder present in the opencv<version>/build/

Build the CMakeTargets or **install project** in both Debug and Release configurations.

HID settings are not shown in the command line application.

Change Use Unicode Character set in the Application-> configuration properties -> General -> Project defaults -> character set

In Opencv version 3.4.1, Opencv_test_namespace related errors while building.

Unload the tests accuracy and tests performance projects from the opencv and start the building process again.

IAMVIDEOCONTROL related error while building Opencv.

Copy the **strmbase.lib**, if using release mode or strmbasd.lib, if using debug mode from the BaseClasses project path and paste it in the **build/lib/release (or) debug** directory of the OpenCV. Also based on the x86 or x64 architecture, the libs should be copied and pasted. If the strmbase.lib or strmbasd.lib is not present. Then build the **baseclasses.sln** using visual studio 2017.

If git is not recognized as an internal or external command error

You need to download and install the git exe from internet with the following link.

<https://gitforwindows.org/>

Support

Contact Us

If you need any support on Hyperyon Dual Stream product, please contact us using the Live Chat option available on our website - <https://www.e-consystems.com/>

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - <https://www.e-consystems.com/create-ticket.asp>

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - <https://www.e-consystems.com/RMA-Policy.asp>

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - <https://www.e-consystems.com/warranty.asp>

Revision History

Rev	Date	Description	Author
1.0	02-July-2020	Initial Draft	Murali Mohan. M
1.1	16-July-2020	Changed BaseClasses project solution download link	Murali Mohan. M
1.2	11-August-2020	Add changes	Murali Mohan. M