OpenCV

Windows Installation Manual

e-con Systems

Your Product Development Partner

Version 1.0

e-con Systems

4/6/2018

Disclaimer

Contents

INTRODUCTION TO OPENCV 3

PREREQUISITES 3
DESCRIPTION 3

BUILDING OPENCV 4

STEP 1 - LAUNCHING CMAKE WINDOW 4
STEP 2 - SELECTING VISUAL STUDIO VERSION 4
STEP 3 - CONFIGURING AND GENERATING CMAKE 5
STEP 4 - REPLACING VIDEOIO FILE 7
STEP 5 - BUILDING OPENCV IN VISUAL STUDIO 7

BUILDING SAMPLE CODE 8

TROUBLESHOOTING 11

SUPPORT 12

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 all e-con Systems cameras.

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

Prerequisites

The prerequisites are as follows:

- Click here to download CMake.
- Click here to download Sources for OpenCV_v3.3.1.
- Create a Build directory in the OpenCV_v3.3.1.
- Build OpenCV in your PC using Visual Studio

Description

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

Building OpenCV

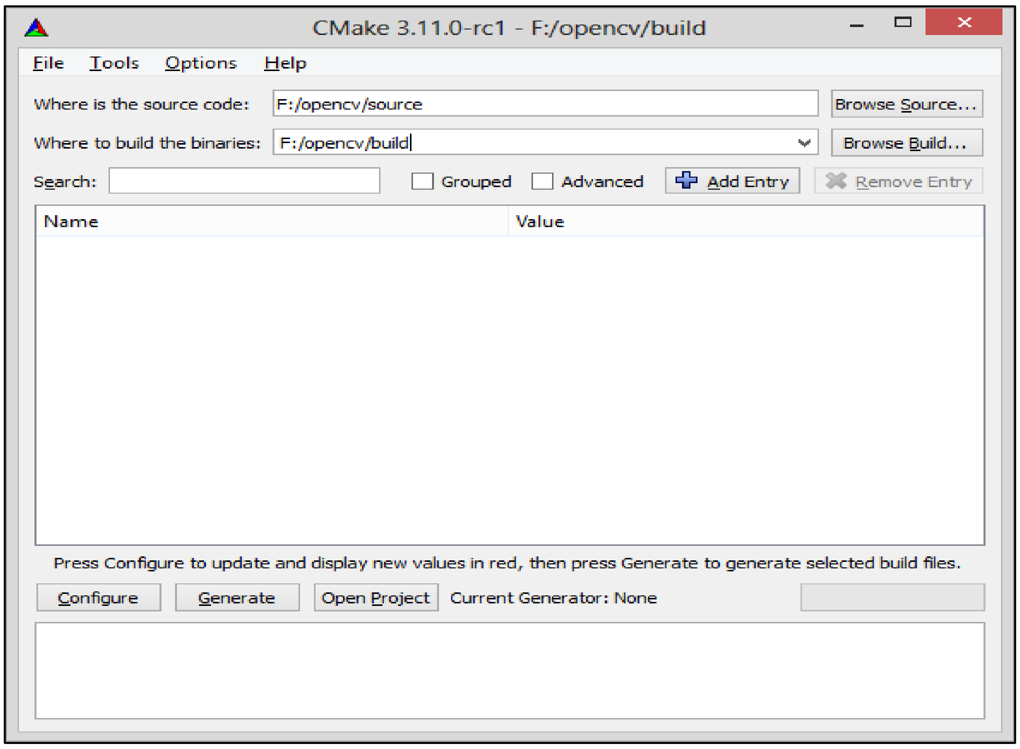
OpenCV is a sample command line application used to demonstrate some of the features of the e-con Systems cameras with OpenCV APIs.

The steps to build OpenCV are as follows:

Step 1. Launching CMake Window
Step 2. Selecting Visual Studio Version
Step 3. Configuring and Generating CMake
Step 4. Replacing Videoio File
Step 5. Building OpenCV in Visual Studio

Step 1 - Launching CMake Window

In CMake window, select the OpenCV sources as source folder and OpenCV_v3.3.1/build as build folder and click **Configure** button.



Where is the source code: F:/opencv/source

Where to build the binaries: F:/opencv/build

Search:

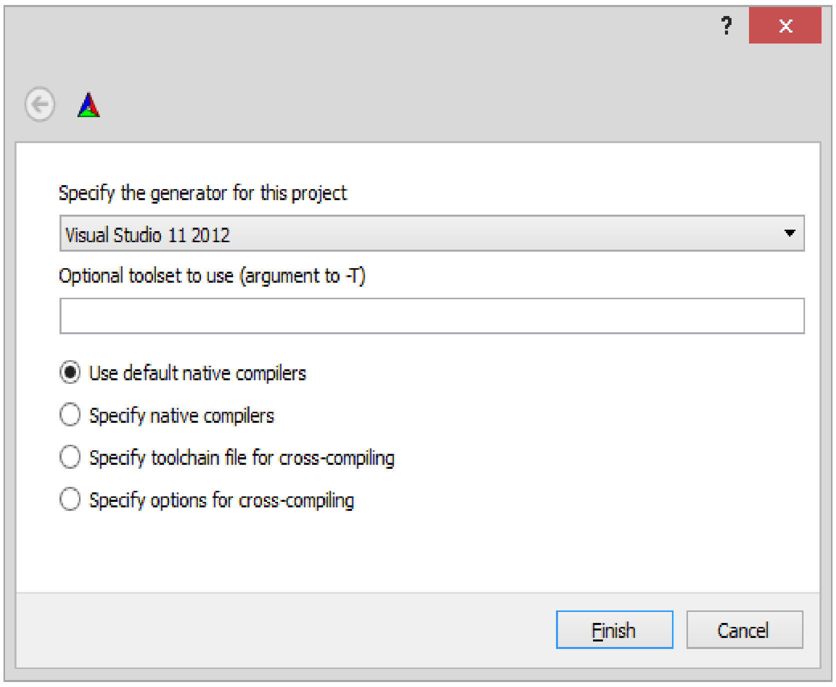
Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

Figure 1: CMake Source and Build Directory Specification Window

Step 2 - Selecting Visual Studio Version

A window prompting to select Visual Studio version along with x32 and x64 version appears. Select the appropriate options as shown below.


Specify the generator for this project

Visual Studio 11 2012

Optional toolset to use (argument to -T)

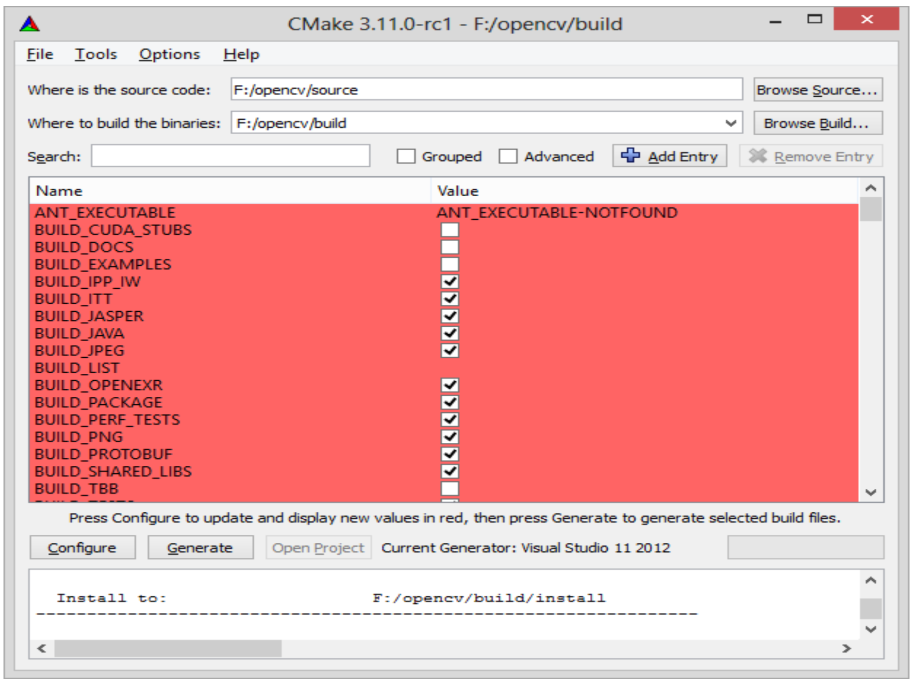
[✓ Use default native compilers]
Specify native compilers
Specify toolchain file for cross-compiling
Specify options for cross-compiling

Figure 2: Selecting Visual Studio Version in CMake

Step 3 - Configuring and Generating CMake

The steps to configure and generate CMake are as follows:

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



Where is the source code: F:/opencv/source

Where to build the binaries: F:/opencv/build

Search:

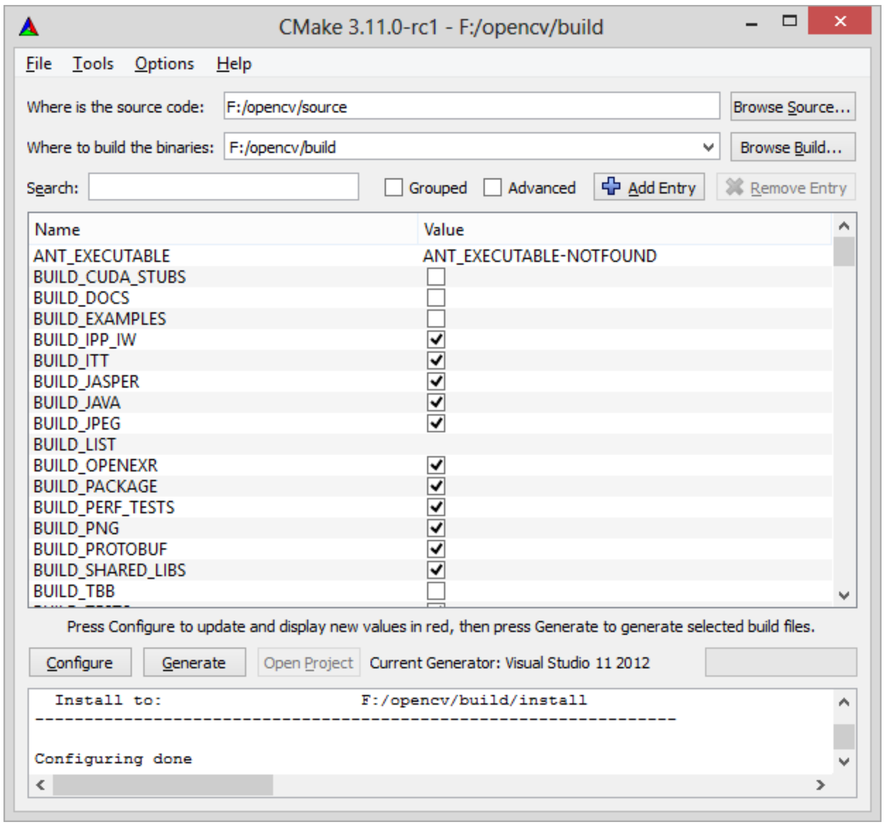
Name	Value
ANT_EXECUTABLE	ANT_EXECUTABLE-NOTFOUND
BUILD_CUDA_STUBS	☐
BUILD_DOCS	☐
BUILD_EXAMPLES	☐
BUILD_IPP_IW	✓
BUILD_ITT	✓
BUILD_JASPER	✓
BUILD_JAVA	✓
BUILD_JPEG	✓
BUILD_LIST	
BUILD_OPENEXR	✓
BUILD_PACKAGE	✓
BUILD_PERF_TESTS	✓
BUILD_PNG	✓
BUILD_PROTOBUF	✓
BUILD_SHARED_LIBS	✓
BUILD_TBB	☐

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 11 2012

```
  Install to:                    F:/opencv/build/install
-----------------------------------------------------------------
```

2. Click **Configure** till all red flag goes off.



Where is the source code: F:/opencv/source

Where to build the binaries: F:/opencv/build

Search:

Name	Value
ANT_EXECUTABLE	ANT_EXECUTABLE-NOTFOUND
BUILD_CUDA_STUBS	☐
BUILD_DOCS	☐
BUILD_EXAMPLES	☐
BUILD_IPP_IW	✓
BUILD_ITT	✓
BUILD_JASPER	✓
BUILD_JAVA	✓
BUILD_JPEG	✓
BUILD_LIST	
BUILD_OPENEXR	✓
BUILD_PACKAGE	✓
BUILD_PERF_TESTS	✓
BUILD_PNG	✓
BUILD_PROTOBUF	✓
BUILD_SHARED_LIBS	✓
BUILD_TBB	☐

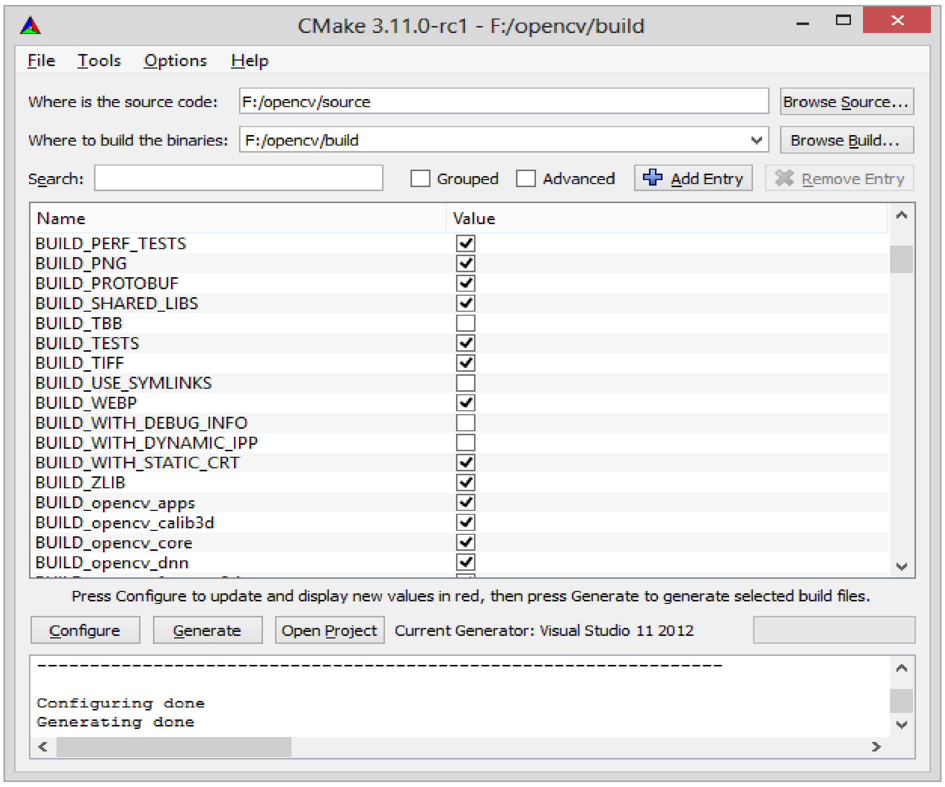
Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 11 2012

```
  Install to:                    F:/opencv/build/install
-----------------------------------------------------------------

Configuring done
```

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



Where is the source code: F:/opencv/source

Where to build the binaries: F:/opencv/build

Name	Value
BUILD_PERF_TESTS	✓
BUILD_PNG	✓
BUILD_PROTOBUF	✓
BUILD_SHARED_LIBS	✓
BUILD_TBB	☐
BUILD_TESTS	✓
BUILD_TIFF	✓
BUILD_USE_SYMLINKS	☐
BUILD_WEBP	✓
BUILD_WITH_DEBUG_INFO	☐
BUILD_WITH_DYNAMIC_IPP	☐
BUILD_WITH_STATIC_CRT	✓
BUILD_ZLIB	✓
BUILD_opencv_apps	✓
BUILD_opencv_calib3d	✓
BUILD_opencv_core	✓
BUILD_opencv_dnn	✓

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 11 2012

```
----------------------------------------------------------------

Configuring done
Generating done
```

Step 4 - Replacing Videoio File

Replace the **videoio** folder with the folder downloaded from the e-con blog () in **OpenCV_v3.3.1/Sources/modules/** location.

Step 5 - Building 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_videoio properties tab under Linker > Input> Additional Dependencies.
3. Build **CMakeTargets/All Build** and **CMakeTargets/Install** separately in both the Debug/Release Configuration of the Visual Studio.

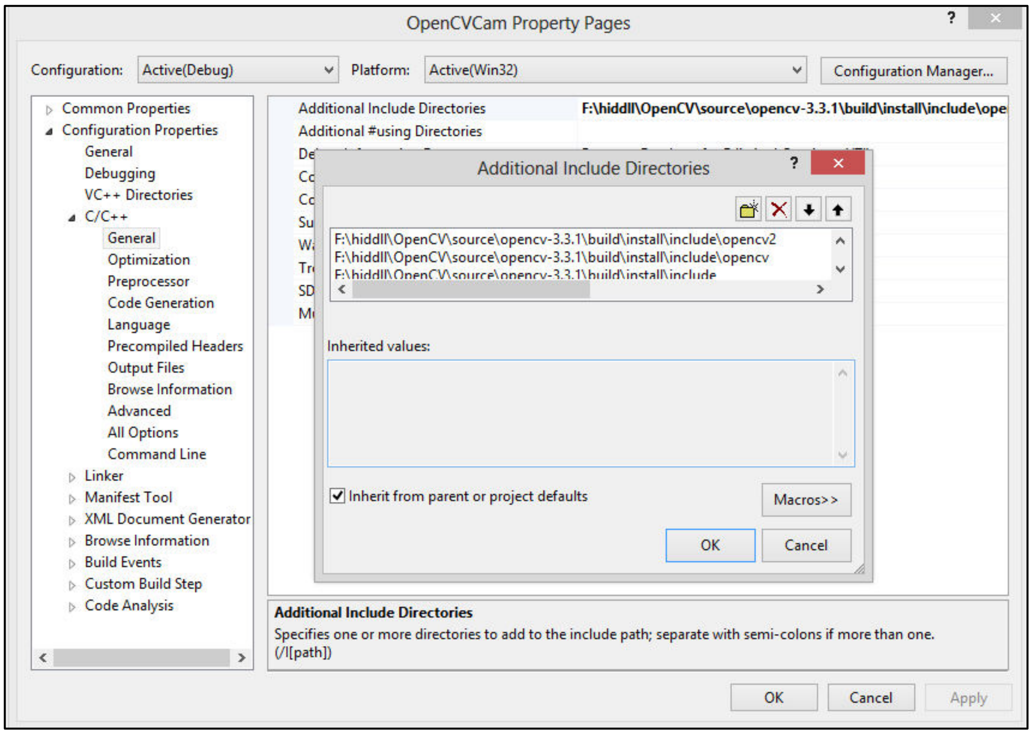
Building Sample Code

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 ().
2. Link the OpenCV header under **C/C++ > General > Additional Include Directories files** from the following:

   - OpenCV_v3.3.1/build/install/include
   - OpenCV_v3.3.1/build/install/include/opencv
   - OpenCV_v3.3.1/build/install/include/opencv2


OpenCVCam Property Pages

Configuration: Active(Debug) Platform: Active(Win32) Configuration Manager...

Common Properties
Configuration Properties
General
Debugging
VC++ Directories
C/C++
General
Optimization
Preprocessor
Code Generation
Language
Precompiled Headers
Output Files
Browse Information
Advanced
All Options
Command Line
Linker
Manifest Tool
XML Document Generator
Browse Information
Build Events
Custom Build Step
Code Analysis

Additional Include Directories F:\hiddll\OpenCV\source\opencv-3.3.1\build\install\include\ope
Additional #using Directories

Additional Include Directories

F:\hiddll\OpenCV\source\opencv-3.3.1\build\install\include\opencv2
F:\hiddll\OpenCV\source\opencv-3.3.1\build\install\include\opencv
F:\hiddll\OpenCV\source\opencv-3.3.1\build\install\include

Inherited values:

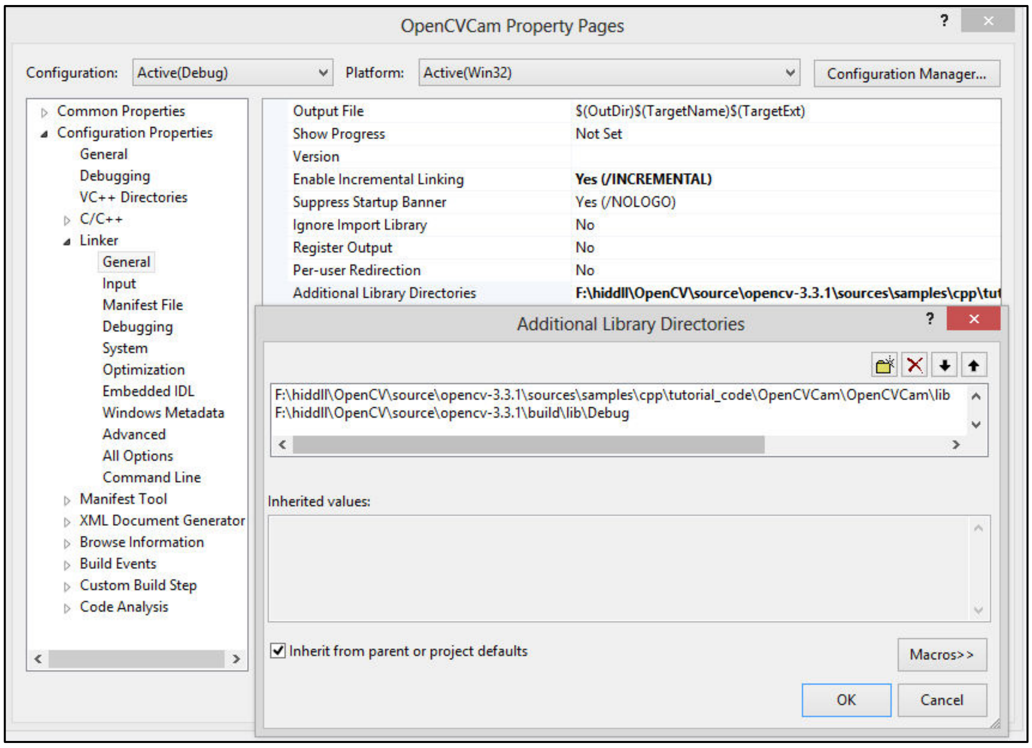
[✓] Inherit from parent or project defaults Macros>>

OK Cancel

Additional Include Directories
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

OK Cancel Apply

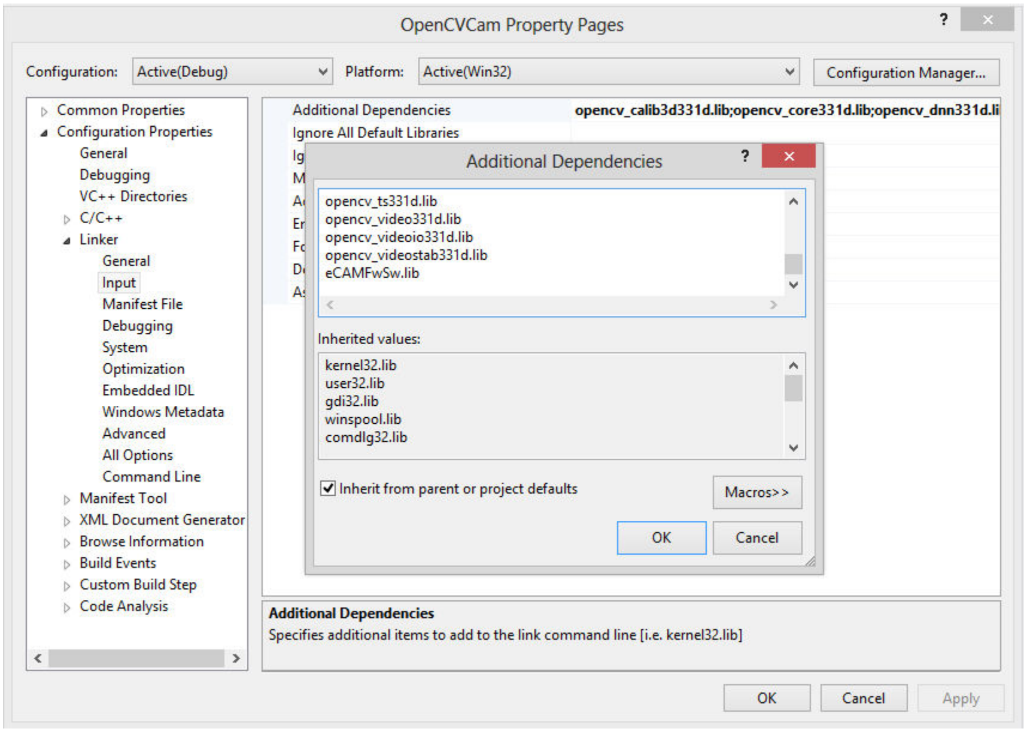
3. Link the library files of OpenCV from the **OpenCV_v3.3.1/build/lib/Release** for configuration type release under **Linker > General > Additional Library Directories**.



Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	**Yes (/INCREMENTAL)**
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**F:\hiddll\OpenCV\source\opencv-3.3.1\sources\samples\cpp\tut**

```
F:\hiddll\OpenCV\source\opencv-3.3.1\sources\samples\cpp\tutorial_code\OpenCVCam\OpenCVCam\lib
F:\hiddll\OpenCV\source\opencv-3.3.1\build\lib\Debug
```

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



Additional Dependencies	**opencv_calib3d331d.lib;opencv_core331d.lib;opencv_dnn331d.li**
Ignore All Default Libraries	

```
opencv_ts331d.lib
opencv_video331d.lib
opencv_videoio331d.lib
opencv_videostab331d.lib
eCAMFwSw.lib
```

Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
comdlg32.lib
```

Additional Dependencies
Specifies additional items to add to the link command line [i.e. kernel32.lib]

5. Add runtime libraries in the sample application **root** folder from the **OpenCV_v3.3.1/build/bin/Debug** or **OpenCV_v3.3.1/build/bin/Release** based on the configuration of the sample application. The runtime libraries for **release version** are:

- opencv_core331.dll
- opencv_highgui331.dll
- opencv_imgcodecs331.dll
- opencv_imgproc331.dll
- opencv_videoio331.dll
- eCAMFwSw.dll

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

- opencv_core331d.dll
- opencv_highgui331d.dll
- opencv_imgcodecs331d.dll
- opencv_imgproc331d.dll
- opencv_videoio331d.dll
- eCAMFwSw.dll

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_videoio 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.

Support

Contact Us

If you need any support on OpenCV sample application, 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	10-April-2018	Initial Draft	Chandra Sekar. V