DepthVista

DepthVista CMD Build Manual

Version 1.1

e-con Systems

6/5/2023

e-con Systems

Think Camera. Think e-con.

Disclaimer

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

Contents

Introduction to DepthVista

DepthVista is a 3D camera based on Time of Flight (TOF) technology, USB Video Class (UVC) compliant, USB 3.2 Gen 1 SuperSpeed USB camera from e-con Systems, a leading Embedded Product Design Services Company which specializes in advanced camera solutions.

DepthVista is an RGB-D camera containing both RGB and TOF depth cameras. RGB camera has 1/2.6" AR0234CS CMOS digital image sensor with global shutter from onsemi™. It has dedicated high performance color image signal processor. TOF depth camera has 1/4" CCD sensor and dedicated depth processor. DepthVista is a two-board solution containing camera board with the USB 3.2 Gen 1 interface and laser board along with enclosure.

This document describes how to build the DepthVista console application on the host PC (Windows).

Prerequisites

The prerequisites are as follows:

- DepthVista console application source code.
- Visual Studio (VS 2017).
- DepthVistaSDK.

Description

DepthVista has USB interface controller with USB Type-C connector to interface with the host PC. It is a ready-to-manufacture camera board with all the necessary firmware built-in and is compatible with the UVC version 1.0 standard. You can integrate this camera into the products, and this helps to cut short the time-to-market.

DepthVista is a UVC compatible and will work with the standard drivers available with Windows and Linux OS. There is no need for any additional driver installation. So, video streaming through UVC is possible without any special drivers on OSes that have built-in support for UVC standards.

Table 1: DepthVista supported Format, Resolutions, and Frame Rates

S.No	Format	Camera Mode	Resolution	Frame Rate (fps) USB 3.2 Gen 1
1	UYVY	RGB Mode	2.3MP (1920 x 1200)	30
			FHD (1920 x 1080)	30
			HD (1280 x 720)	60

			VGA (640 x 480)	60
2	Y16 (RAW 12-bit)	TOF Mode	Depth (640 x 480)	30
			IR (640 x 480)	30
			Depth + IR (640 x 960)	30
3		RGB-D Mode	1280 x 600 (RGB-D)	30
			1443 X 960 (RGB-D)	30

The TOF camera in DepthVista can be used in two depth modes as follows:

- **Far Mode**: Effective depth range is between 1000 mm to 6000 mm.
- **Near Mode**: Effective depth range is between 200 mm to 1200 mm.

The TOF camera controls of DepthVista are as follows:

- TOF Data Mode
- TOF Depth Range
- TOF Mask
- TOF Gain

The RGB camera controls of DepthVista are as follows:

- Brightness
- Contrast
- Saturation
- Gamma
- Gain
- Sharpness
- White Balance
- Exposure
- Power line frequency

Installing OpenCV 4.2.0

This section will describe the procedures for installing the OpenCV libraries for building in windows.

The procedures for installing the OpenCV libraries for building in windows are as follows:

1. Download the installer for OpenCV 4.2.0 (64 bit) from the following link.

   https://sourceforge.net/projects/opencvlibrary/files/4.2.0/opencv-4.2.0-vc14_vc15.exe/download.

   Note: For 32-bit, user must build OpenCV from the source.

2. Double click on the **opencv-4.2.0-vc14_vc15.exe file**. The installer launch screen will appear as shown below.

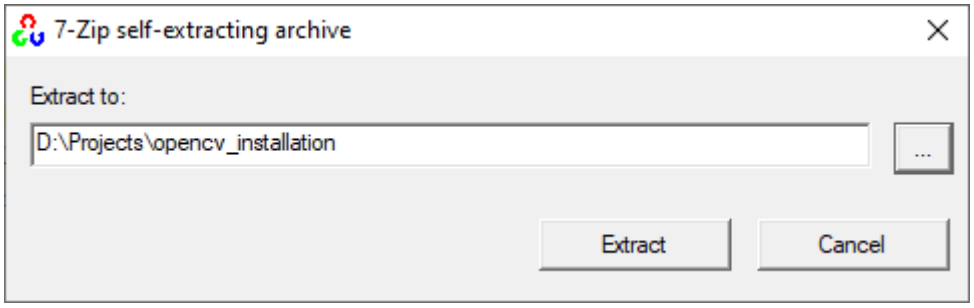

Figure 1: Extraction Location Screen.

3. Select the directory in which the binaries are to be extracted. Click **Extract.**

   Once you click **Extract**, binaries will be extracted, and the window will look as shown below.

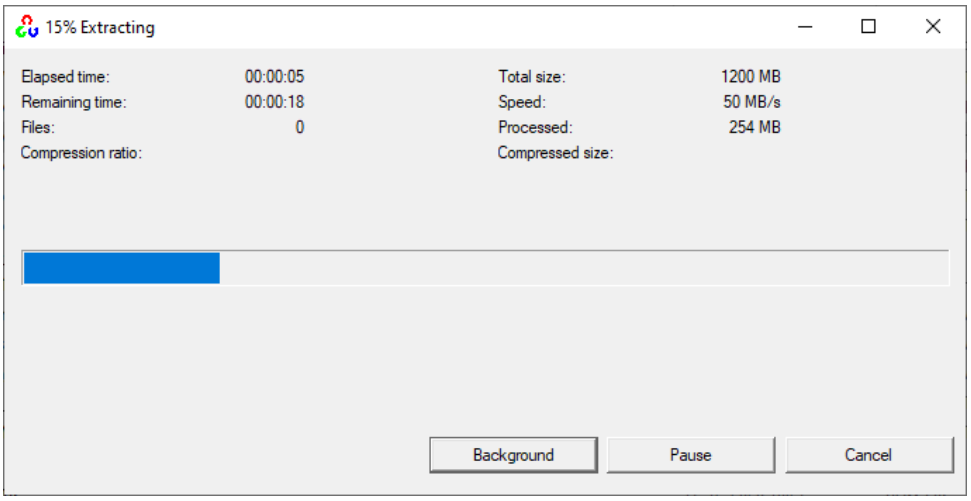

Figure 2: Extraction Process Screen

Building DepthVista Application

This section will describe the procedures for building the DepthVista Application.

The steps for building the DepthVista Application are as follows:

1. Extract the **SDK Package**
   <Extracted Directory>/Windows/Source/CPP will contain the DepthVistaCmd.sln file.
   <Extracted Directory>/Windows/Bin will contain the SDK which includes the headers, .lib files and .dll files.
2. Open DepthVistaCmd Project in Visual Studio.

   The steps to open DepthVistaCmd Project are as follows:

   1. Open the new instance of visual studio.
   2. Click **File**->**Open**->**Project/Solution** as shown below.

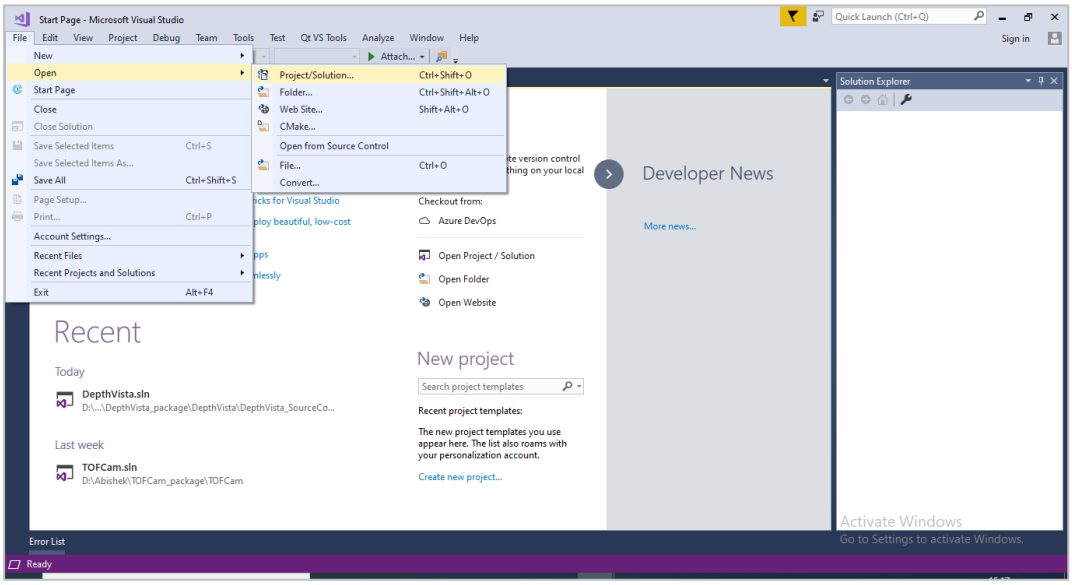

Figure 3: Opening Solution File In Visual Studio

3. Browse the **DepthVista** project and select **DepthVistaCmd.sln**.
4. Select **Solution configuration** (Debug / Release) and **Solution Platform** Win32 or x64) (based on your requirement.

   Note: Win32 and x86 are the same.

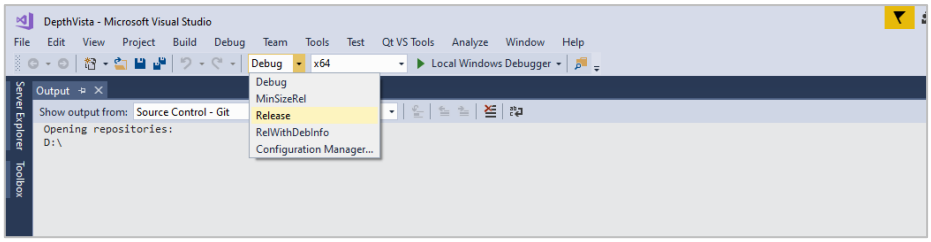

Figure 4: Choosing Solution Configuration

DepthVistaCmd.sln will contain two projects as shown below:

- DepthVistaConsoleApp
- DepthVistaIMU

Note: Steps for Configuring both the projects are same. So, in this document we have demonstrated the steps for configuring DepthVistaConsoleApp project. Follow the same steps for configuring DepthVistaCmd).

Configuring Project

The steps to configure the project are as follows:

1. Right click on the project **DepthVistaConsoleApp** in the **Solution Explorer** and select **Properties** as shown below.

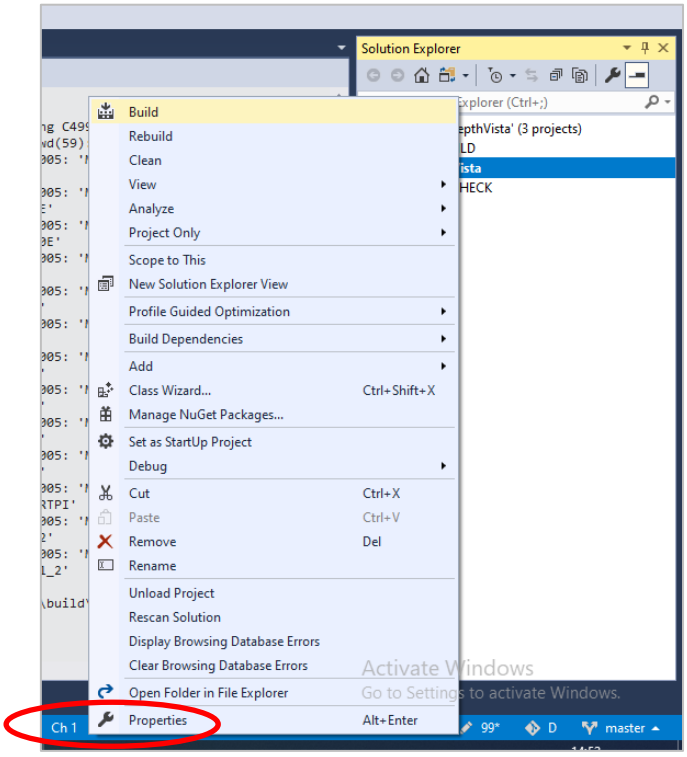

Figure 5: Opening Properties Page

Then the properties window will open as shown below. Make sure that the Configuration and Platform in the property page and in Visual Studio UI are set as same.

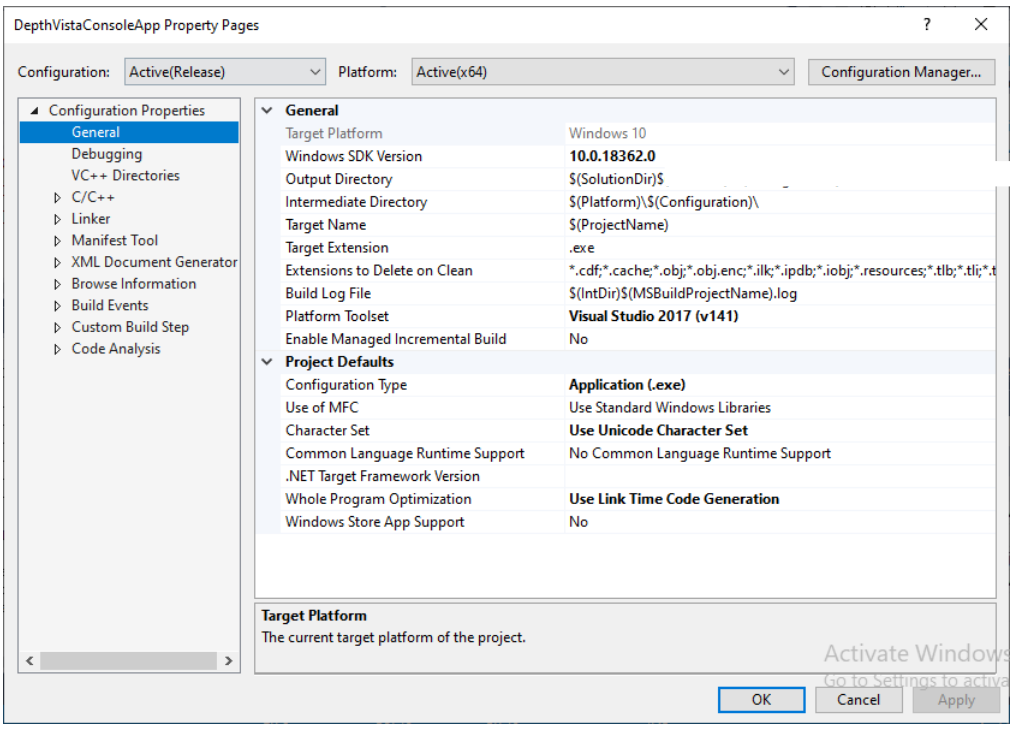

Figure 6: Visual Studio Property Page

2. Navigate to **C++->General->Additional Include Directories** and select **<Edit…>** as shown below.

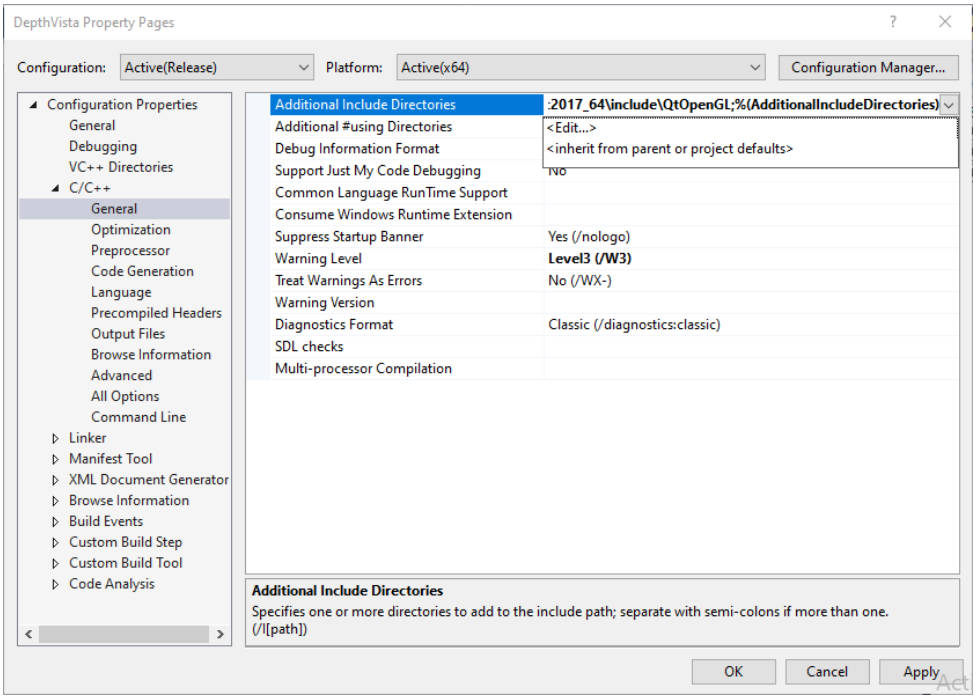

Figure 7: Adding Additional Include Directories.

3. Enter the directory in which OpenCV include files are present and then Click **OK**.

Including Additional Directories

Include files will be present inside <OpencvExtractionDirectory>\opencv\build\include.

When OpenCV is built from source, include the following directories:

- <OpenCV Build Directory>\install\include

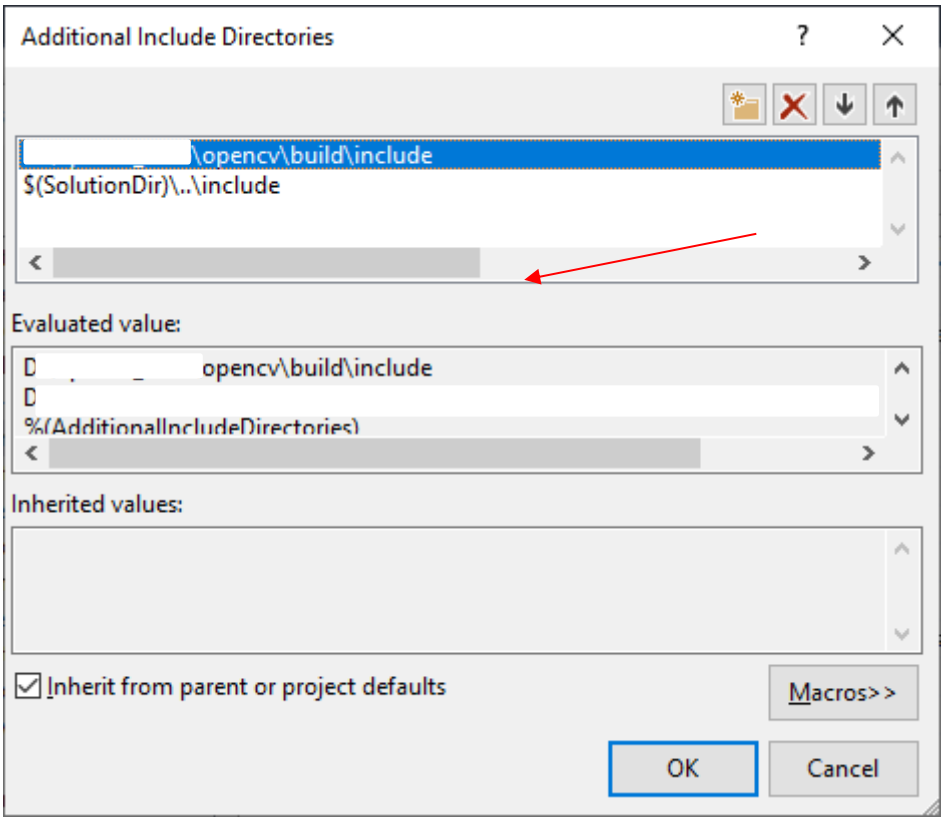

Figure 8: Adding Additional Include Directories.

Navigate to **Linker->General->Additional Library Directories** and select **edit** as shown below.

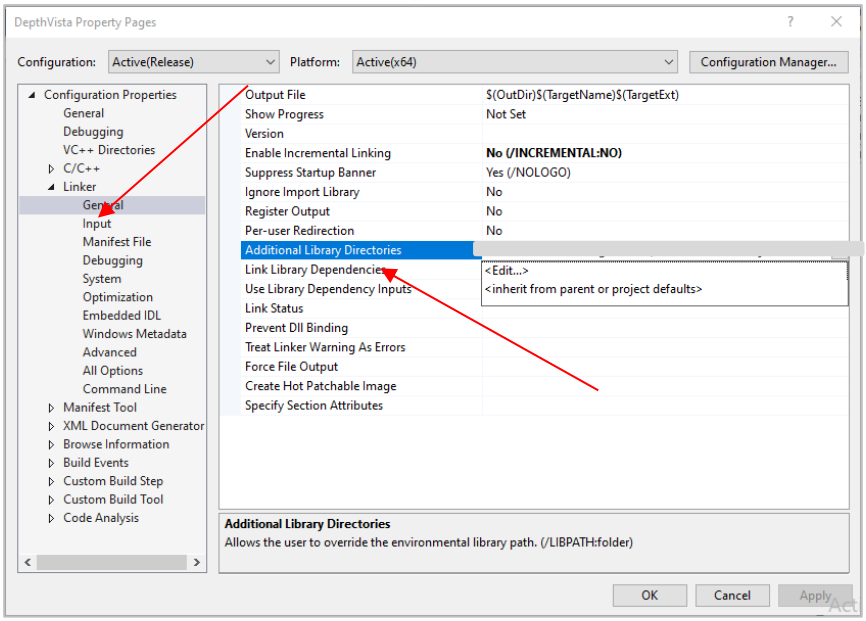

Figure 9: Adding Additional Library Directories.

For **release** build, enter the directory in which the **opencv_world420.lib** file is present. Then click **OK.**

For **debug** build, enter the directory in which the **opencv_world420d.lib** file is present. Then click **OK.**

Note:

- opencv_world420.lib and opencv_world420d.lib will be present inside <OpenCVExtractionDirectory>\opencv\build\x64\vc15\lib.
- When OpenCV is built from source for x86 platform, **opencv_world420.lib** be present inside <OpenCV Build directory>\install\x86\vc15\lib.
- After making all the mentioned changes click **Apply** button as shown to apply the made changes in the DepthVista project. Then click **OK.**

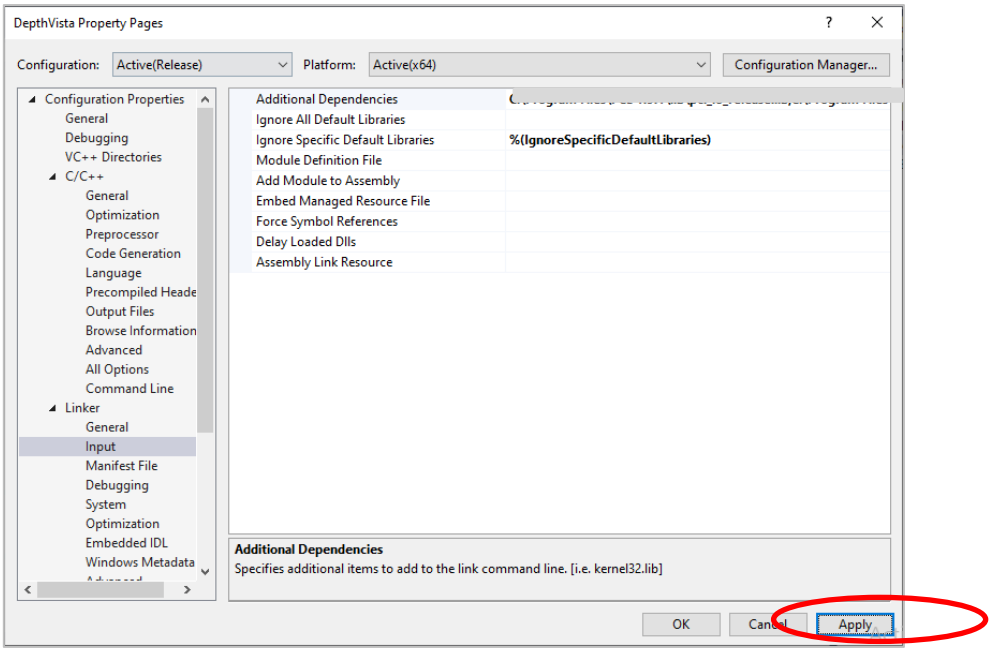

Figure 10: Applying Changes to DepthVista Project

Building DepthVistaConsoleApp Project

The steps to build the DepthVistaConsoleApp project are as follows:

1. Right click on the **DepthVistaConsoleApp** project and select **Build** (or **rebuild**) as shown below.

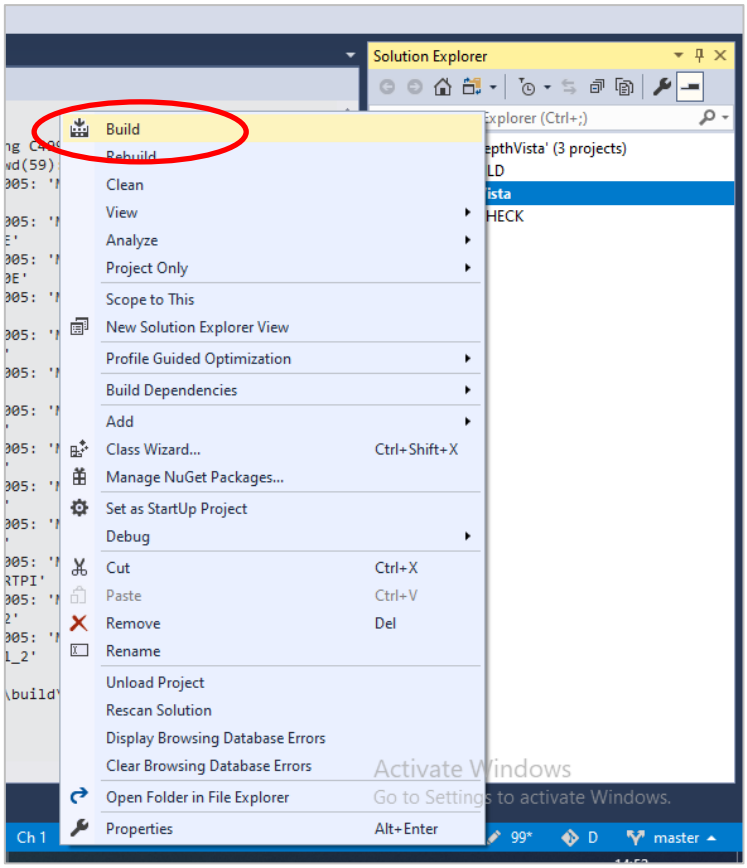

Figure 11: Building DepthVista Project

2. Build the solution.

   Once the build is success, then you can view the message in the output tab of visual studio as shown below. It will also contain the folder in which the **DepthVistaConsoleApp.exe** is placed. It will usually be placed in **Release or Debug** directory from Solution Directory.

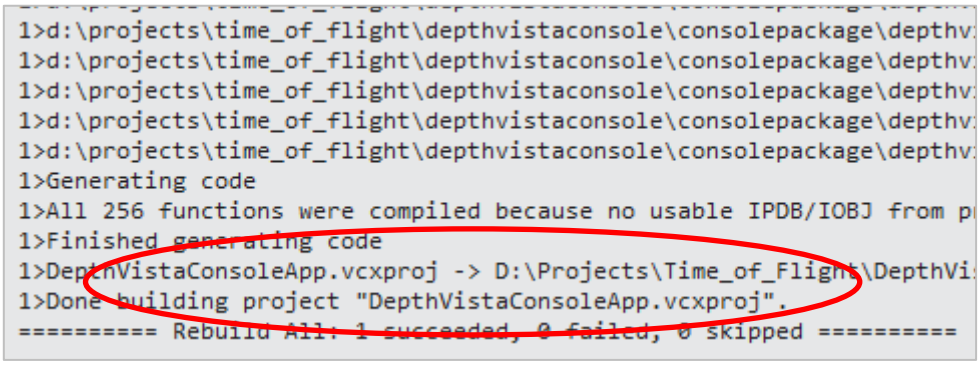

Figure 12: Message on Successful Build of DepthVista Project

After building you can find the **DepthVistaConsoleApp.exe** in the respective Release and Debug folders.

Running DepthVista Application

Add runtime libraries of OpenCV (**opencv_world420.dll** for release build and **opencv_world420d.dll** for debug build) to the directory containing the **DepthVistaConsoleApp.exe** file.

The runtime libraries of OpenCV will be present in the following path:

- <ExtractionDirectory>\opencv\build\x64\vc15\bin, when OpenCV is installed from installer for x64 platform.
- <OpenCV build directory>\install\x86\vc15\bin, when OpenCV is built from source.

The steps to run the DepthVista application are as follows:

1. Run the **DepthVistaConsoleApp.exe** application.
2. Follow the *DepthVista_Console_Application_User_Manual.pdf* provided in the package.

Support

Contact Us

If you need any support on DepthVista 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

1. **Why does Windows systems headers like Windows.h, SDKDDKVer.h, gives error?**

   This error is due to incompatible Windows SDK Version.

   Follow the below step to build the project again:

   1. Open the property page of the project.
   2. Navigate to **Configuration Properties->General->Windows SDK Version.** Select any installed SDK version and build the project again.

Revision History

Rev	Date	Description	Author
1.0	02-November-2022	Initial Draft	Camera Products
1.1	05-June-2023	Document changes	Camera Products