

Smart Video Evaluation Toolkit – Sample Application User Guide for Windows

May 2021



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: <http://www.intel.com/design/literature.htm>

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at <http://www.intel.com/> or from the OEM or retailer.

Intel MediaSDK, OpenVINO and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2021, Intel Corporation. All rights reserved.



Contents

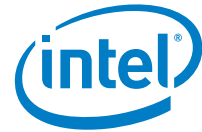
1.0	Installation Guide.....	5
1.1	System installation.....	5
1.2	Install OpenVINO 2021.3 and Media SDK.....	5
1.3	Build SVET sample application and dependent libraries	7
1.4	Prepare inference model	13
1.5	Prepare the video clips for testing.....	14
2.0	Run SVET sample application	15
2.1	Check environment variables.....	15
2.2	Modify the video path in parameter file	15
2.3	Run video_e2e_sample application	16
2.3.1	16-channel video decoding, face detection, composition, encode and display	17
2.3.2	4-channel video decoding, human pose estimation, composition, and display	17
2.3.3	4-channel video decoding, vehicle and vehicle attributes detection, composition, encode and display	18
2.3.4	16-channel RTSP video decoding, face detection, composition, encode and display	19
2.3.5	16-channel RTSP video decoding, RTSP stream storing, face detection, composition, encode, and display	19
2.3.6	HDDL inference, face detection and GPU composition, encode, display	20
2.4	Usage of media codec, inference and display parameters in par file	20
2.4.1	New parameters in Par file.....	20
2.4.2	Decode, encode and display parameters.....	21
3.0	Frequently asked question.....	23



Revision History

Date	Revision	Description
2020/05/26	0.5	Initial release
2020/09/07	0.6	Fixed some bug
2021/05/10	0.7	Update OpenVINO to 2021.3, and Media SDK to 2020 R1

Note: Releases in the table are listed in reverse order so that the latest/newest is in the top row.



1.0 Installation Guide

1.1 System installation

Install Windows 10 enterprise on any Intel® Core™ or Atom™ processors. (Better to be after the 5th generation Core™ and Baytrail for Atom™ processors)

Set up the network correctly

1.2 Install OpenVINO 2021.3 and Media SDK

The sample application video_e2e_sample depends on OpenVINO libraries, Media SDK library and FFmpeg library. We suggest users to install OpenVINO 2021.3 package from <https://software.intel.com/content/www/us/en/develop/tools/opencvino-toolkit.html>

Select options below to download

Operating System:	Select operating system Windows
Distribution:	Select distribution Web & Local (recommended)
Version Type:	Select version 2021.3 (latest)
Installer Type:	Select installer Local

Offline Installer

- Includes all tools in the toolkit
- Recommended for host machines with poor or no internet connection

Initial download: 257 MB
Maximum download: 521 MB (based on operating system)

Select Your Release Type

Standard Release Recommended for new users and users that are currently prototyping. Offers new features, tools, and support to stay current with deep learning technology advancements.	Long-Term Support (LTS) Release Recommended for experienced users that are ready to take their application into production and who do not require new features and capabilities for their application.
--	--

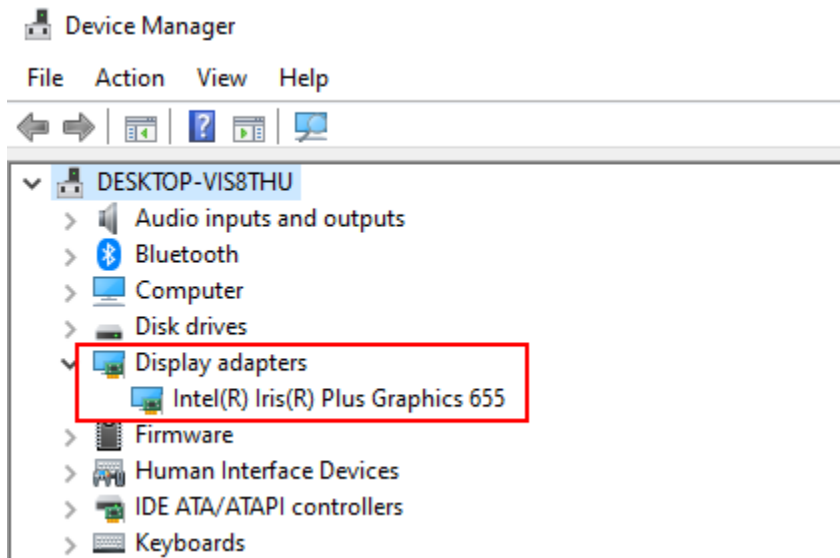
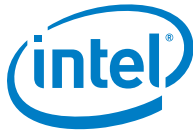
[Download](#)

By downloading, you agree to our [Privacy](#) and [Terms of use](#)

Please Install OpenVINO 2021.3 according to https://docs.openvino toolkit.org/latest/_docs_install_guides_installing_openvino_windows.html

By default, OpenVINO 2021.3 is installed to "C:\Program Files (x86)\IntelSWTools\openvino_2021.3.394".

Make sure the Graphic driver is installed correctly in the Windows device manager, it should look like below status.



If the driver is not installed properly, you can download the Graphic driver by below link:

https://downloadcenter.intel.com/product/80939/Graphics-Drivers?elq_cid=3226637&erpm_id=5670311

Please Install Media SDK 2020 R1 according to:

<https://software.intel.com/en-us/media-sdk/choose-download/client>

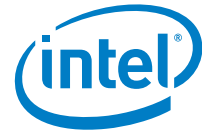
Media SDK installation guide as below:

- 1, Install the SDK and library files. (Note Installation requires full administrative rights.)
- 2, Set up the Microsoft Visual Studio* environment with the SDK and library directories.
- 3, Run the Intel_Media_SDK_20xx_xx.msi installer. The default installation directory is c:\Program Files (x86)\IntelSWTools\Intel(r)_Media_SDK_XXXX where XXXX is the year and version number.
- 4, After installation, restart your system to initialize INTELMEIASDK_WINSOCK_PATH and install the remaining files.

Other Media SDK Installation document, you can reference to:

<https://software.intel.com/en-us/media-sdk/documentation/featured-documentation>

when you build video_e2e_sample, it will depend on Media SDK sample code (sample_common and sample_plugins), which you can find in below folder, after Media SDK installed:



C:\Users\iotg\Documents\Intel(R) Media SDK 2020 R1 - Media Samples 8.4.32.0

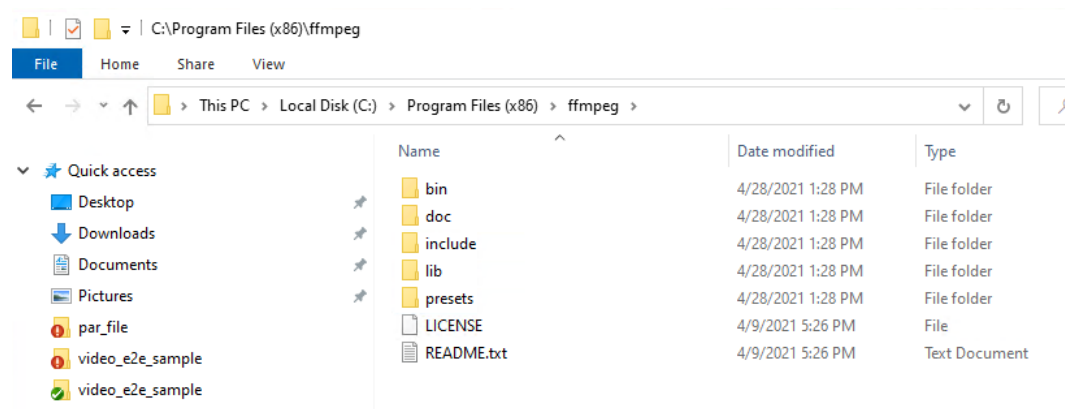
And for the RTSP function, this video_e2e_sample building also depends on FFMPEG library, it can be downloaded from:

<https://www.gyan.dev/ffmpeg/builds/ffmpeg-release-full-shared.7z>

in dev and shared folder, download ffmpeg-release-full-shared.7z and extracted to

C:\Program Files (x86)\ffmpeg

The folder structure will look like below:



which will be used to compile the SVET in visual studio.

1.3 Build SVET sample application and dependent libraries

Download svet_e2e_sample_w.zip sample package, or you can git clone the svet_e2e_sample_w code from github <https://github.com/intel-iot-devkit/concurrent-video-analytic-pipeline-optimization-sample-w>

and extract it to a folder, such as

C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w

Copy the Media SDK dependent sample code to svet_e2e_sample_w,

```
cp C:\Users\iotg\Documents\Intel(R) Media SDK 2020 R1 - Media Samples
8.4.32.0\sample_common

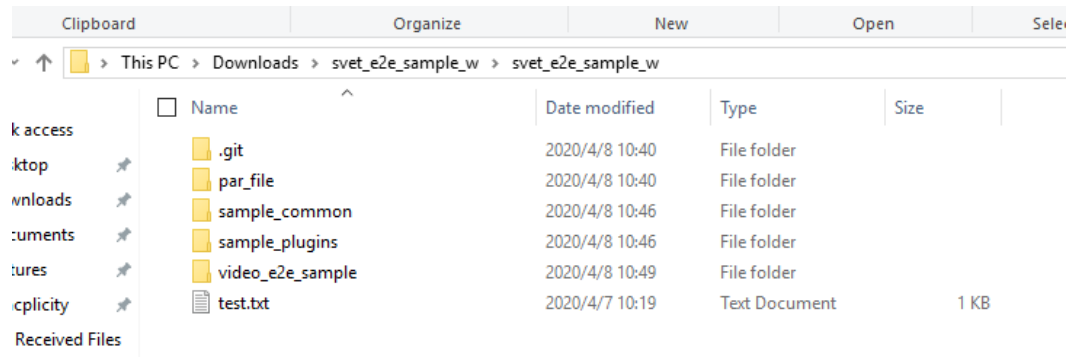
C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w

cp C:\Users\iotg\Documents\Intel(R) Media SDK 2020 R1 - Media Samples
8.4.32.0\sample_plugins

C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w
```



Then the folder will look like below.



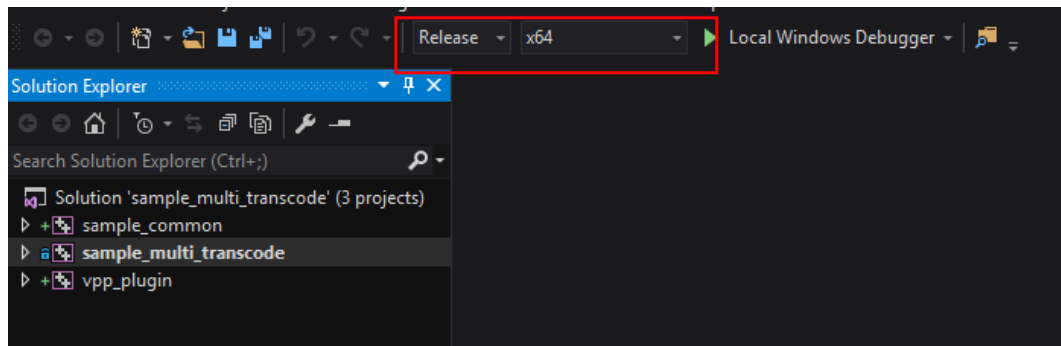
Open Visual studio (suggest to use Visual studio 2017 and Windows SDK 10), Windows SDK download link:

<https://go.microsoft.com/fwlink/p/?LinkId=838916>

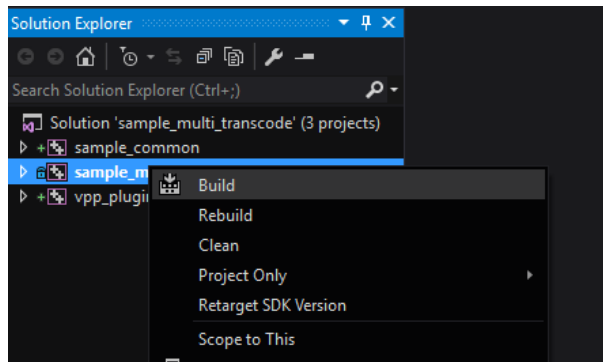
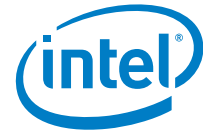
Open File->Open Project, choose:

C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\video_e2e_sample\sample_multi_transcode.vcxproj

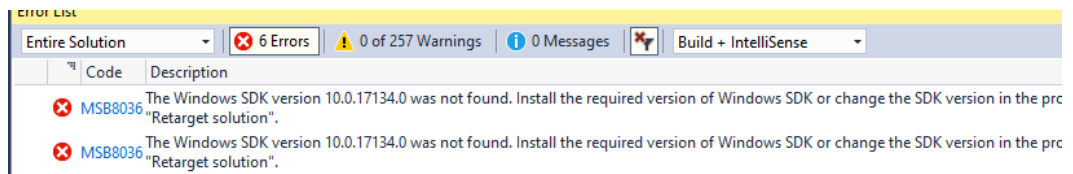
If the project load successes, it will look like below.



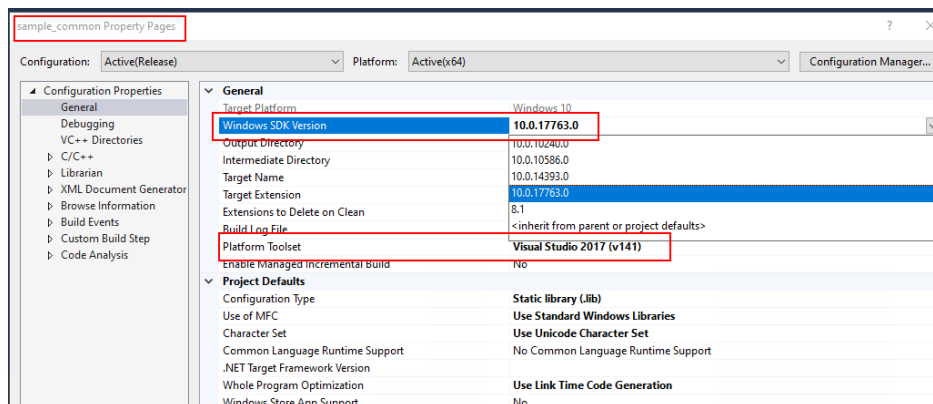
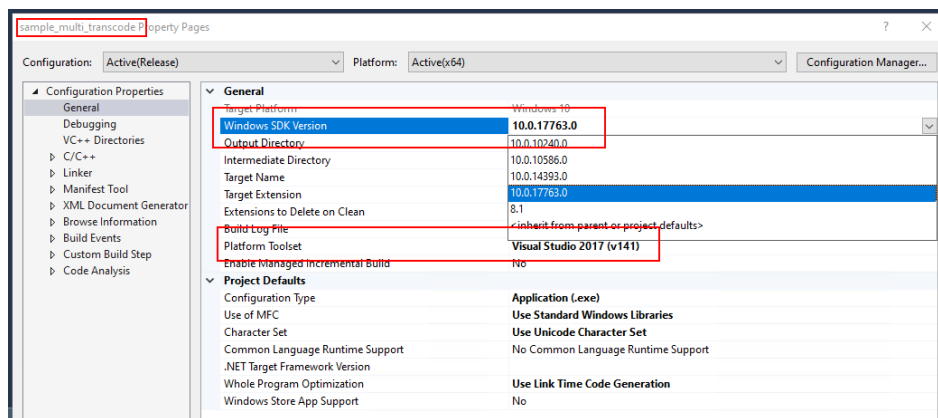
Choose "Release" and "X64", and Right Click "Sample_multi_transcode", choose Build, it will start building.

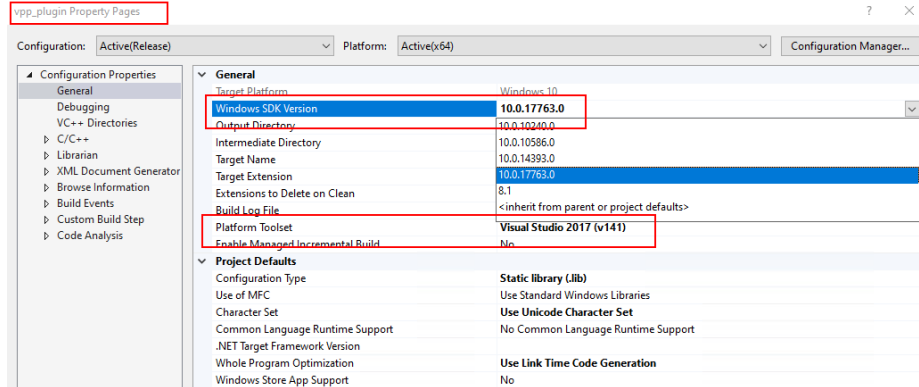
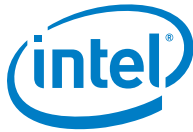


The first compile issue you may meet, "Windows SDK version 10.0.17.134.0 was not found":

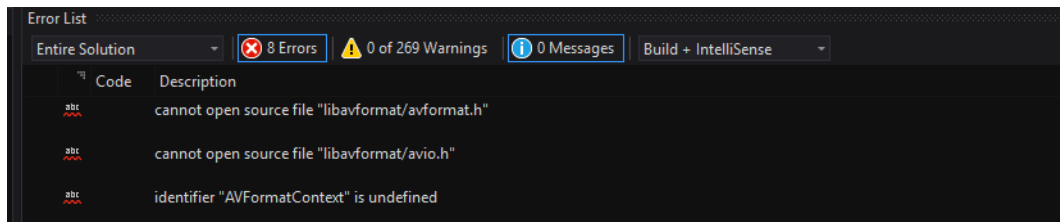


Fix method, change the three project's property config like below, the SDK and platform toolset could be the version which your computer already installed:

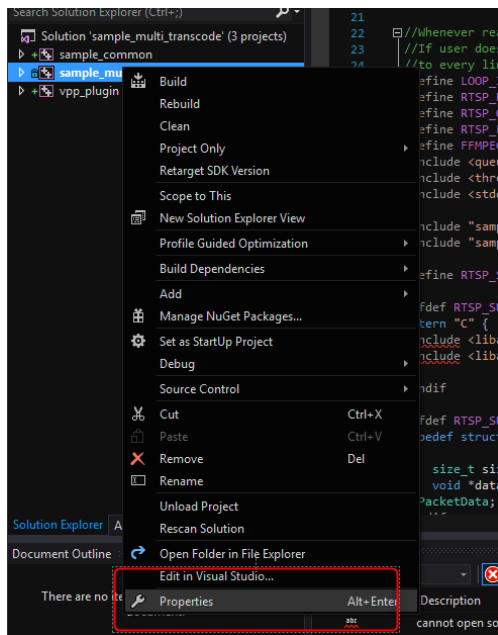




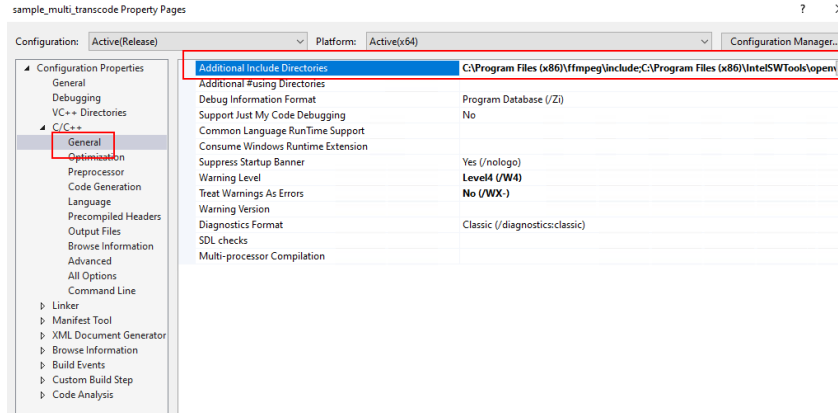
The second compile issue is like below, “cannot find source file “libavformat/avformat.h”



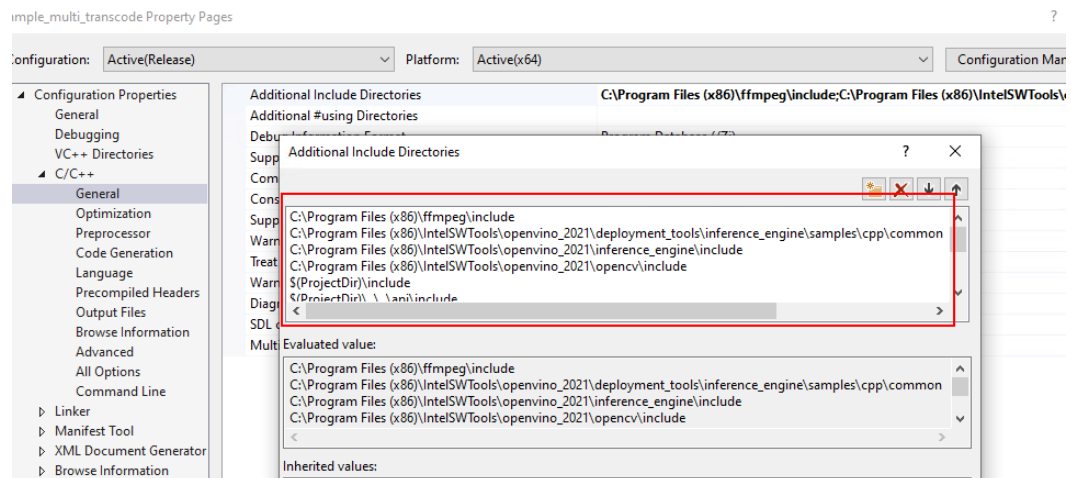
You should check the Additional Include Directories, by right clicking. “sample_multi_transcode” and choose “properties”, select C/C++->General->Additional Include Directories.



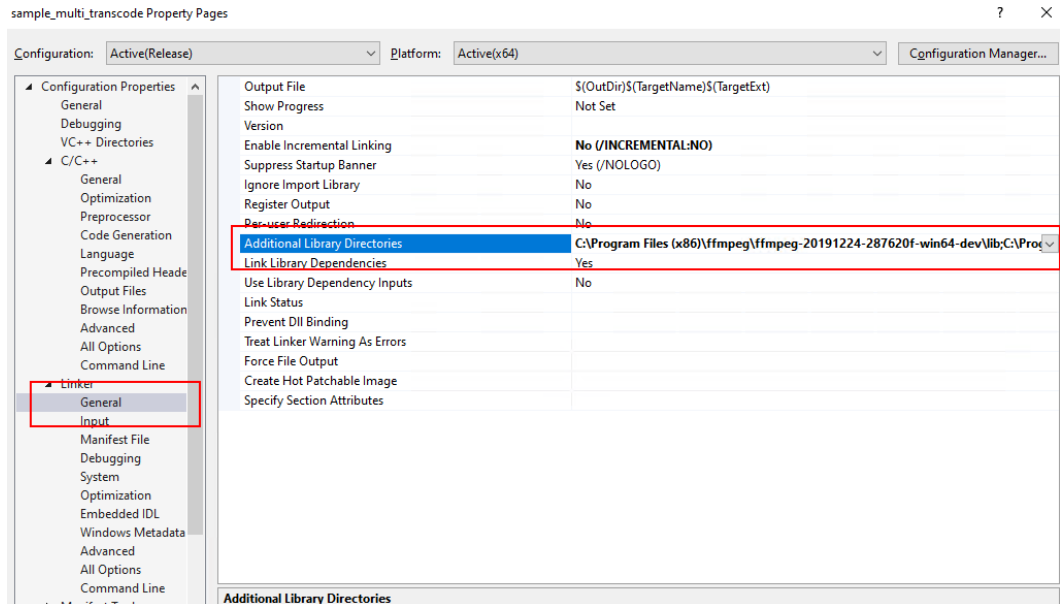
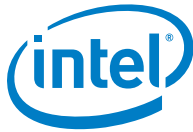
Additional Include Directories looks like below:



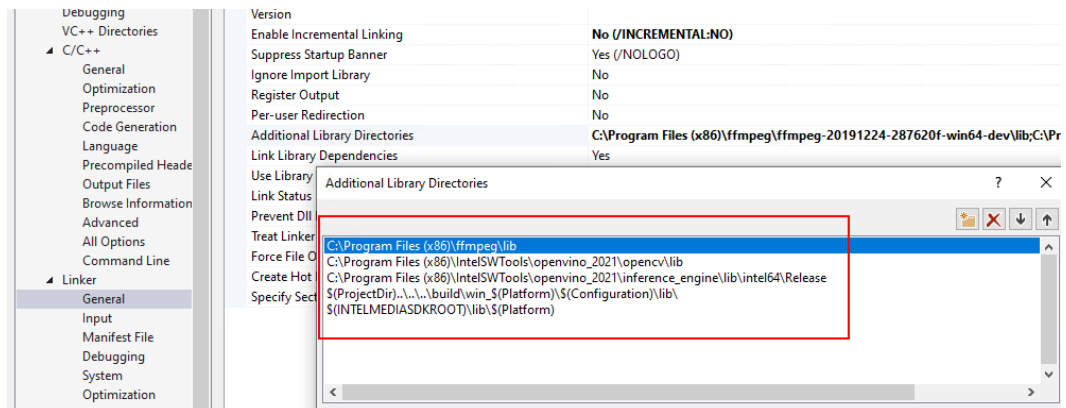
Click the down arrow at the end and Click the <Edit...> button, you will see below dialog, please make sure, below Include folder can be found on your PC.



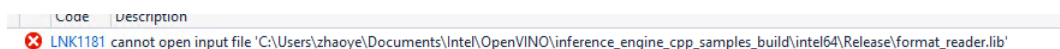
And the same thing for the “Additional Library Directories” and “Additional Dependencies” in Linker->General->Additional Library Directories and Linker->Input->Additional Dependencies.



Additional Dependencies, click the down arrow and <Edit...>, then a dialog jump out, it looks like below:



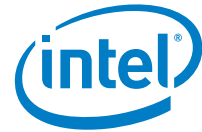
Then, the third compile issue may be is "format_reader.lib" is not found:



Fix method is running below bat script to build this library.

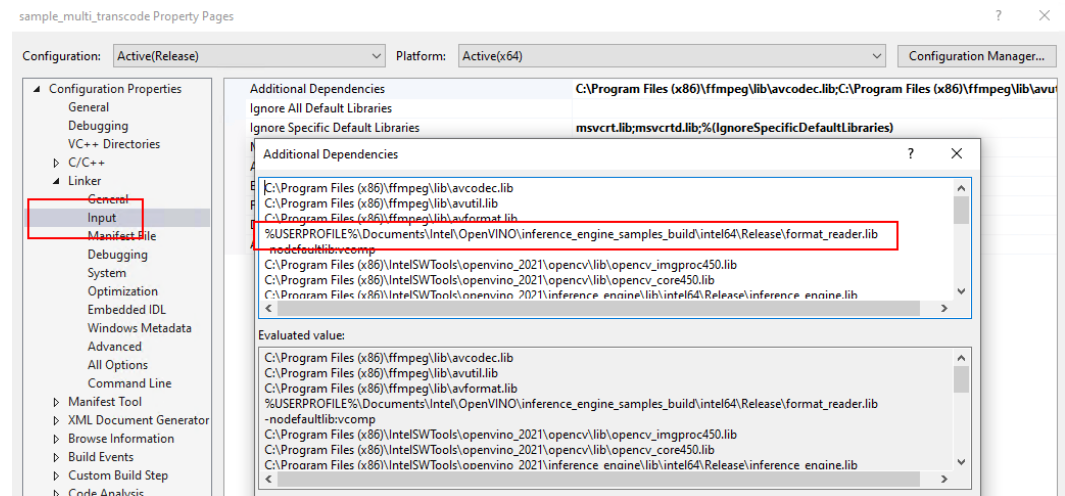
C:\Program Files
(x86)\IntelSWTools\openvino_2021\deployment_tools\inference_engine\samples\cpp
\build_samples_msvc.bat

You will find the format_reader.lib in :



%USERPROFILE%\Documents\Intel\OpenVINO\inference_engine_cpp_samples_build\intel64\Release\format_reader.lib

Then set this folder link to sample_multi_transcode property:



Then you will finish the config and see below succeeded log, and the exe file will be generated in ..\build\win_x64\Release\bin\video_e2e_sample.exe.



1.4 Prepare inference model

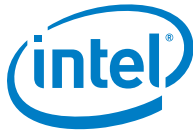
Download inference IR file through script\download_IR_file.py

You can execute below script in windows console (requires full administrative rights):

```
Cd svet_e2e_sample_w\script
python _IR_file.py
```

you will find all the inference IR file in folder
svet_e2e_sample_w\video_e2e_sample\model\:

below IR file is needed for later test.



This PC > Downloads > svet_e2e_sample_w > svet_e2e_sample_w > video_e2e_sample > model				
	Name	Date modified	Type	Size
access	face-detection-retail-0004.bin	2019/11/11 15:09	BIN File	1,149 KB
ktop	face-detection-retail-0004.xml	2019/11/11 15:09	XML Document	47 KB
vnloads	human-pose-estimation-0001.bin	2019/11/11 15:11	BIN File	8,006 KB
uments	human-pose-estimation-0001.xml	2019/11/11 15:11	XML Document	65 KB
ures	vehicle-attributes-recognition-barrier-0039.bin	2019/11/7 11:05	BIN File	1,223 KB
cplicity	vehicle-attributes-recognition-barrier-0039.xml	2019/11/7 11:05	XML Document	18 KB
ple_multi_trans	vehicle-license-plate-detection-barrier-0106.bin	2019/11/7 11:05	BIN File	1,257 KB
T	vehicle-license-plate-detection-barrier-0106.xml	2019/11/7 11:05	XML Document	98 KB

1.5 Prepare the video clips for testing

If you don't have video clip for testing, you can download sample videos from <https://github.com/elainewangprc/testvideo/> link:

Wang Elaine Update video clip ...		32a505f 2 days ago	🕒 3 commits
📄 1080p.h264	Upload two test video clips	2 days ago	
📄 README.md	Upload two test video clips	2 days ago	
📄 car_1080p.h264	Update video clip	2 days ago	

Git clone this git project, and copy these two video clips to

svet_e2e_sample_w\video_e2e_sample\content

folder.

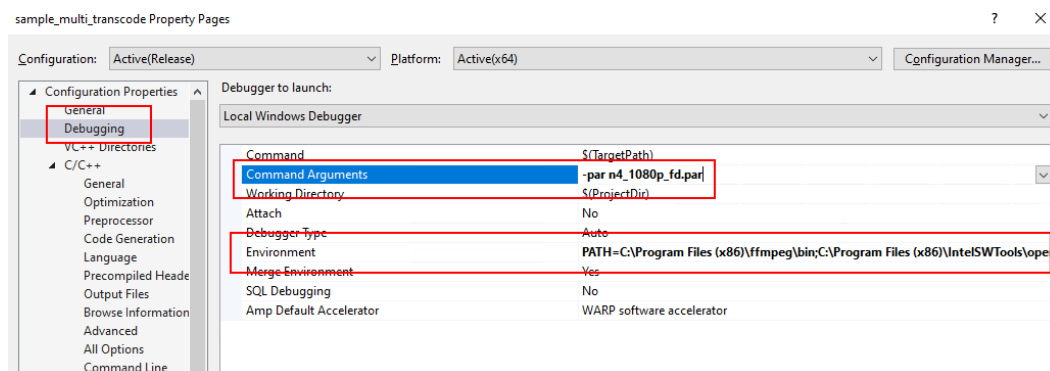


2.0 Run SVET sample application

2.1 Check environment variables

Setting below debugging environment path to Debugging->Environment.

```
PATH=C:\Program Files (x86)\ffmpeg\bin;C:\Program Files  
(x86)\IntelSWTools\openvino_2021\deployment_tools\inference_engine\external\hddl\bin;C:\Program Files (x86)\IntelSWTools\openvino_2021\opencv\bin;C:\Program Files  
(x86)\IntelSWTools\openvino_2021\deployment_tools\inference_engine\external\tbb\bin;C:\Program Files  
(x86)\IntelSWTools\openvino_2021\deployment_tools\nggraph\lib;C:\Program Files  
(x86)\IntelSWTools\openvino_2021\deployment_tools\inference_engine\bin\intel64\Release;%PATH%
```



If you are running the application in console, you should set the environment path by:

```
set PATH=C:\Program Files (x86)\ffmpeg\bin;C:\Program Files  
(x86)\IntelSWTools\openvino_2021\deployment_tools\inference_engine\external\hddl\bin;C:  
\Program Files (x86)\IntelSWTools\openvino_2021\opencv\bin;C:\Program Files  
(x86)\IntelSWTools\openvino_2021\deployment_tools\inference_engine\external\tbb\bin;C:  
Program Files (x86)\IntelSWTools\openvino_2021\deployment_tools\nggraph\lib;C:\Program  
Files  
(x86)\IntelSWTools\openvino_2021\deployment_tools\inference_engine\bin\intel64\Release;  
%PATH%
```

2.2 Modify the video path in parameter file

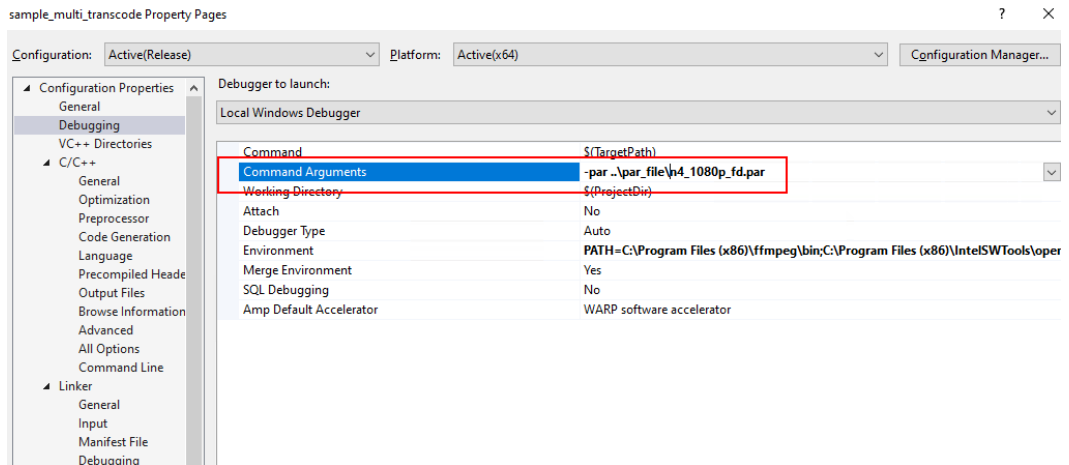
Modify the video path (following "-i::h264") of **every line** in example par file under C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\par_file\n4_1080p_hp.par.



```
-i::h264 content\2.h264 -dc::rgb4 -join -hw_d3d11 -threads 3 -async 10 -timeout 3600 -o::sink -vpp_comp_dst_x 0 -v  
-i::h264 content\2.h264 -dc::rgb4 -join -hw_d3d11 -threads 3 -async 10 -timeout 3600 -o::sink -vpp_comp_dst_x 960  
-i::h264 content\2.h264 -dc::rgb4 -join -hw_d3d11 -threads 3 -async 10 -timeout 3600 -o::sink -vpp_comp_dst_x 0 -v  
-i::h264 content\2.h264 -dc::rgb4 -join -hw_d3d11 -threads 3 -async 10 -timeout 3600 -o::sink -vpp_comp_dst_x 960  
  
-vpp_comp_only 4 -w 1920 -h 1080 -async 10 -join -hw_d3d11 -i::source -ext_allocator
```

Set the par file as Command Arguments:

`-par ..\par_file\n16_1080p_fd.par`

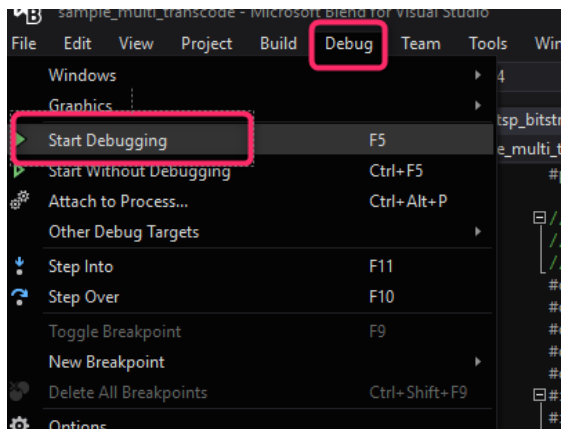


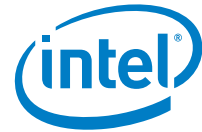
If run the application in console, you should run below command:

```
cd C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\video_e2e_sample  
..\_build\x64\Release\video_e2e_sample.exe -par ..\par_file\n4_1080p_fd.par
```

2.3 Run video_e2e_sample application

Choose the Debug->Start Debugging, Then the application can run.





When running 16-channel inference, loading of inference models can take long time. due to the run time building of OpenCL kernel for each model. It's recommended to enable OpenCL kernel cache by make a "cl_cache" folder in C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\video_e2e_sample\ Then only one time OpenCL kernel building is required, after that, the built kernel can be loaded from cl_cache directory.

2.3.1 16-channel video decoding, face detection, composition, encode and display

Command:

```
cd C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\video_e2e_sample  
..\_build\x64\Release\video_e2e_sample.exe -par ..\par_file\n4_1080p_fd.par
```

The face detection inference is specified by "-infer::fd ./model" in the par file. " ./model " is the directory that stores human pose detection model IR files.



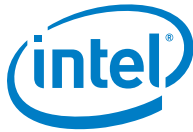
If you want to stop the application, close the console.

If you want to play 200 frames in each decoding session, you can append "-n 200" to parameters lines starting with "-i" in par files.

2.3.2 4-channel video decoding, human pose estimation, composition, and display

Command:

```
cd C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\video_e2e_sample
```

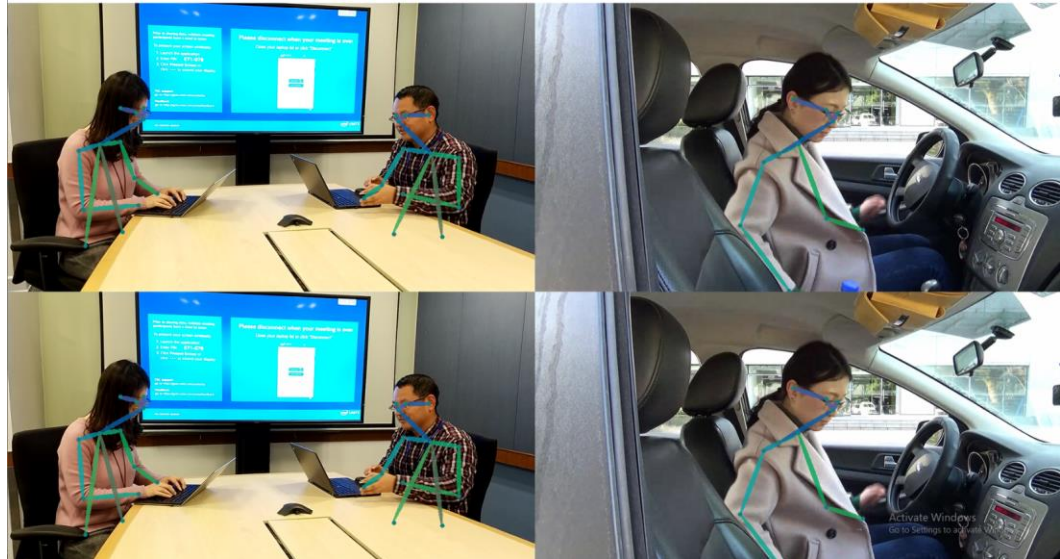


```
..\_build\x64\Release\video_e2e_sample.exe -par ..\par_file\n4_1080p_hp.par
```

Replace the “-par ..\par_file\n16_1080p_fd.par” to “-par ..\par_file\n4_1080p_hp.par”

The human pose detection inference is specified by “-infer::hp ./model” in the par file. “./model ” is the directory that stores human pose estimation model IR files.

Below picture is the screenshot of this demo.



2.3.3 4-channel video decoding, vehicle and vehicle attributes detection, composition, encode and display

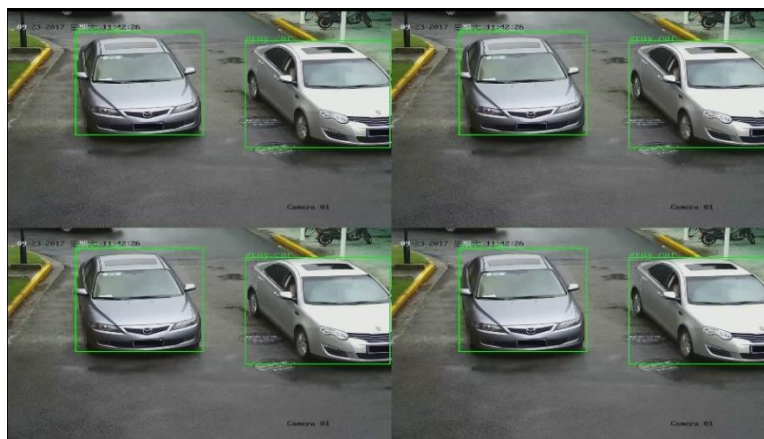
Command:

```
cd C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\video_e2e_sample  
..\_build\x64\Release\video_e2e_sample.exe -par ..\par_file\n4_1080p_vd.par
```

Replace the “-par ..\par_file\n16_1080p_fd.par” to “-par ..\par_file\n4_1080p_vd.par”

The vehicle and vehicle attributes detection inference is specified by “-infer::vd ./model” in the par file. “./model ” is the directory that stores vehicle and vehicle attributes detection model IR files.

Below picture is the screenshot of this demo.



2.3.4 16-channel RTSP video decoding, face detection, composition, encode and display

Command:

```
cd C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\video_e2e_sample  
..\_build\x64\Release\video_e2e_sample.exe -par ..\par_file\n4_1080p_rtsp.par
```

Replace the “-par ..\par_file\n16_1080p_fd.par” to “-par ..\par_file\n4_1080p_rtsp.par”

To use RTSP video stream instead of local video file, you can modify the par file and use RTSP URL to replace local video file path.

```
-i:h264 rtsp://192.168.0.8:1554/simu0000 -join -hw -async 4 -dec_postproc -o::sink -vpp_comp_dst_x  
0 -vpp_comp_dst_y 0 -vpp_comp_dst_w 480 -vpp_comp_dst_h 270 -ext_allocator -infer::fd ./model
```

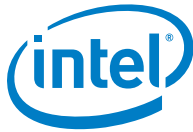
2.3.5 16-channel RTSP video decoding, RTSP stream storing, face detection, composition, encode, and display

Command:

```
cd C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\video_e2e_sample  
..\_build\x64\Release\video_e2e_sample.exe -par ..\par_file\n16_1080p_rtsp.par
```

Replace the “-par ..\par_file\n16_1080p_fd.par” to “-par ..\par_file\n16_1080p_rtsp.par”

The name of RTSP streaming local file is specified by option “-rtsp_save filename” in decoding session in par file. User can choose one or more sessions to invoke the RTSP stream storing



2.3.6 HDDL inference, face detection and GPU composition, encode, display

To perform inference on Intel® Vision Accelerator Design with Intel® Movidius™ VPUs, the following additional installation steps are required:

1. If your Intel® Vision Accelerator Design with Intel® Movidius™ VPUs card requires SMBUS connection to PCIe slot (Raw video data card with HW version Fab-B and before), install the SMBUS driver:
 - a. Go to the <INSTALL_DIR>\deployment_tools\inference-engine\external\hddl\SMBusDriver directory, where <INSTALL_DIR> is the directory in which the Intel Distribution of OpenVINO toolkit is installed.
 - b. Right click on the hddlsmbus.inf file and choose **Install** from the pop up menu.

Command:

```
cd C:\Users\Myname\Downloads\svet_e2e_sample_w\svet_e2e_sample_w\script  
Run_FD_HDDL_test.bat
```

If you want to enable HDDL human pose detection or vehicle detection, you need to change the par_file\n4_1080p_fd_HDDL.par

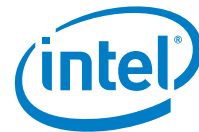
Replace the “-infer::fd” to “-infer::hp” or “-infer::vd”.

2.4 Usage of media codec, inference and display parameters in par file

2.4.1 New parameters in Par file

Comparing to original sample_multi_transcode, we add some new parameters.

Parameter	Usage
-infer::infer_type ir_file_dir	Specify the inference type and directory that stores the IR files. Examples: -infer::fd ./model →face detection -infer::hp ./model →human pose estimation -infer::vd ./model →vehicle and vehicle attributes detection
-infer::Device_type	-infer::GPU →Running inference workload on GPU -infer::HDDL →Running inference workload on HDDL card



-i:h264 rtsp://url	Specify the source H264 file with RTSP URL
-rtsp_save filename.h264	Save RTSP stream to local file. This parameter must be used together with "-i:h264 rtsp://url". If the whole line of session parameters only contains "-i:h264 rtsp://url -rtsp_save filename.h264" and don't have other decoding parameters, we call such sessions as RTSP stream storing session and they must be put into a separated par file.
-infer::interval	Set the inference frame interval, -infer::interval 3, means every 3 frame, do one time inference operation.
-infer::max_detect	Set max detect target number, -infer::max_detect 1, means detect 1 target in a frame.
-o::raw /dev/null	Drop decode result, do not do encode and display.

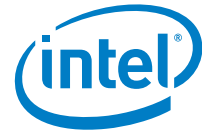
2.4.2 Decode, encode and display parameters

Below table explains the parameters used in example par files. The full parameter list can also be found at https://github.com/Intel-Media-SDK/MediaSDK/blob/master/doc/samples/readme-multi-transcode_linux.md

Parameter	Usage
-i:h264 h264 input_video_filename	Set input file and decoder type
-o:h264 h265 output_video_filename	Set output file and decoder type
-o::sink	The output will be passed to the sink sessions,, e.g. encoding session or composition session
-i::source	The input is coming from source sessions like decoding session
-vpp_comp_dst_x 0 -vpp_comp_dst_y 270 -vpp_comp_dst_w 480 -vpp_comp_dst_h 270	(x, y) position and size of this stream in composed stream
-join	Join session with other session(s). If there are several transcoding sessions, any number of sessions can be joined. Each session includes decoding, preprocessing (optional), and encoding
-hw_d3d11	GPU will be used for HW accelerated video decoding, encoding and post-processing.
-async <async_depth>	Depth of asynchronous pipeline.
-threads <thread_number>	Number of session internal threads to create
-ext_allocator	Force usage of external allocators



-n	Number of frames to transcode (session ends after this number of frames is reached). In decoding sessions (-o::sink) this parameter limits number of frames acquired from decoder. In encoding sessions (-o::source) and transcoding sessions this parameter limits number of frames sent to encoder.
-fps <fps>	Transcoding frame rate limit
-vpp_comp <sourcesNum>	Enables composition from several decoding sessions. Result is written to the file
-vpp_comp_only <sourcesNum>	Enables composition from several decoding sessions. Result is shown on screen.
-ec::nv12 rgb4	Forces encoder input to use provided chroma mode.



3.0 *Frequently asked question*

1. The loading time of 16-channel face detection demo is too long

Please enable `cl_cache` by make folder named "cl_cache" in video_e2e_sample, it will accelerate the loading time.

2. How to limit the fps of whole pipeline to 30?

Add "`-fps 30`" to every decoding session.

3. How to limit the frame number of inputs to 1000?

Add "`-n 1000`" to every decoding sessions. However this option won't work if both "`-vpp_comp_only`" and "`-vpp_comp`" are set.

4. Where can I find tutorials for inference engine?

Please refer to

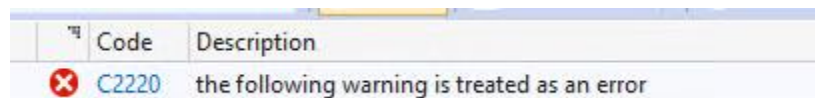
https://docs.openvinotoolkit.org/latest/_docs_IE_DG_Deep_Learning_Inference_Engine_DevGuide.html

5. Where can I find information for the models?

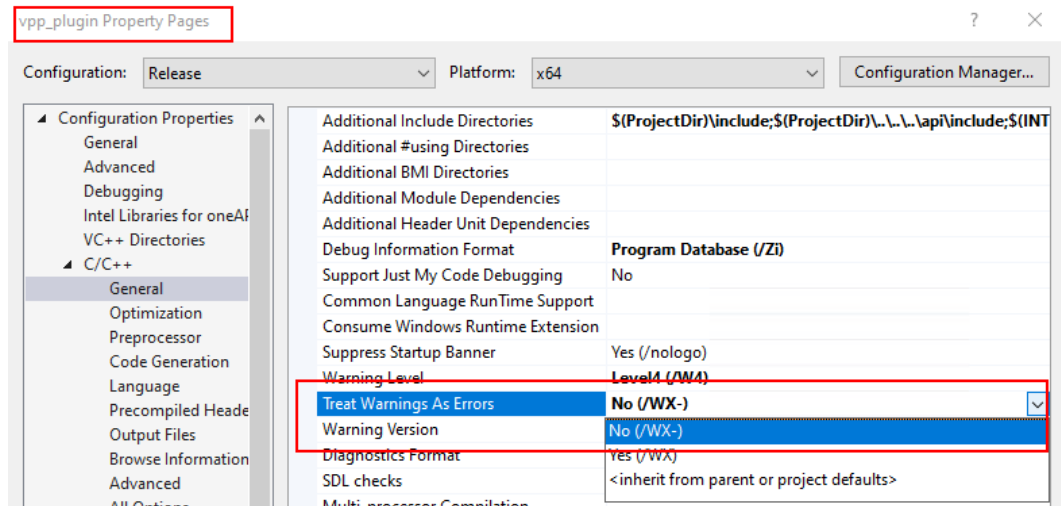
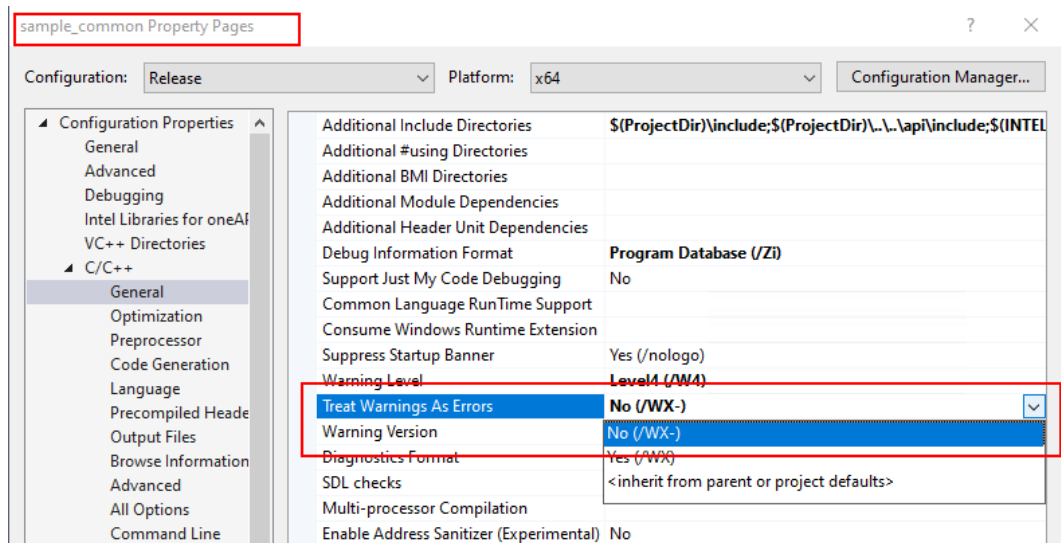
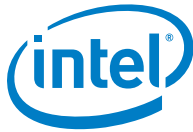
Please refer to https://github.com/opencv/open_model_zoo/tree/master/models/intel. The names of models used in sample application are

face-detection-retail-0004, human-pose-estimation-0001, vehicle-attributes-recognition-barrier-0039, vehicle-license-plate-detection-barrier-0106.

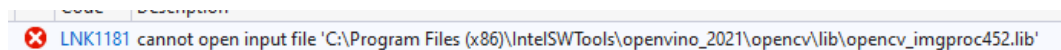
6. If you using Visual Studio 2019, you may found below issue:



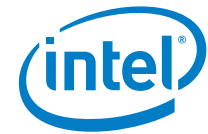
Fix method, set the three project's Treat Warnings As Errors as No:



7, If you find below issue, opencv_imgproc452.lib is not found



You need check the openvino installation folder, it may be installed to "C:\Program Files (x86)\Intel\openvino_2021", and set this link to previews configuration.



↑ > This PC > Local Disk (C:) > Program Files (x86) > Intel > openvino_2021 >			
	Name	Date modified	Type
ccess	bin	4/28/2021 10:39 AM	File folder
p	data_processing	4/28/2021 10:39 AM	File folder
oads	deployment_tools	4/28/2021 10:39 AM	File folder
vents	documentation	4/28/2021 10:39 AM	File folder
is	inference_engine	4/28/2021 10:39 AM	File folder
svm	licensing	4/28/2021 10:39 AM	File folder
Templatel	opencv	4/28/2021 10:39 AM	File folder
.1	python	4/28/2021 10:39 AM	File folder
2			