

# Media Samples Guide

---

## Overview

---

Samples work with **Intel® Media SDK 2016** and **Intel® Media Server Studio 2016 for Windows\* Server**.

They demonstrate how to incorporate the **Intel® Media SDK 2016 for Windows** and **Intel® Media Server Studio – SDK** (hereinafter referred to as "**SDK**") API into various applications.

Some samples can work with **Intel® Media Server Studio – HEVC Decoder & Encoder** (hereinafter referred to as "**HEVC Encoder**", "**HEVC Decoder**", "**HEVC**").

Not all of the samples listed below might be applicable and supported for a particular product. Make sure to check the respective release notes document for potential limitations.

## What's New

---

- Video Transcoding Sample (sample\_multi\_transcode) is extended with set of VPP filters:
  - Composition
  - Denoise
  - Detail (edge detection)
  - Frame rate control (FRC)
  - Deinterlace
  - Color space conversion
- Rendering capability is added to Video Transcoding Sample.
- Looping/breaking capability (-timeout option) is added to Video Transcoding Sample, both for single or multiple session modes.
- Latency measurement is added to Video Transcoding Sample
- Intel® Media SDK RAW Media Accelerator Sample is extended with 3D LUT gamma correction and Chroma aberration correction options
- sample\_decvpp and sample\_decode are now merged into one single Video Decoding Sample (sample\_decode). All the functionality of sample\_decvpp is available in sample\_decode now.
- Decoding sample is extended with deinterlace and color space conversion filters

## Package contents

---

**Full list of available samples:**

- **Video Decoding Sample**

Console application which performs decoding of elementary compressed video stream to raw frames. Includes the following features:

- stereoscopic 3D (S3D) rendering of elementary MVC (Multi-View Video Coding) streams
- decoding of HEVC (High Efficiency Video Coding) video via **HEVC Decoder**
- decoding with video post processing (color conversion) of raw video sequences

---

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

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

Copyright © 2016, Intel Corporation

- screen capturing via screen capture plugin
- **Video Encoding Sample**  
Console application which performs encoding of raw video frames into elementary compressed stream. Includes the following features:
  - video resizing
  - video rotation via User Plug-in Sample
  - video rotation via User Plug-in Sample using OpenCL™
  - encoding HEVC video via **HEVC Encoder**
- **Video Processing Sample**  
Console application which performs various video processing algorithms on raw frames.
- **Video Transcoding Sample**  
Console application which performs transcoding of elementary video stream from one compressed format to another. Includes the following features:
  - multiple video streams transcoding
  - video resizing, de-interlacing
  - video rotation via User Plug-in Sample
  - video rotation via User Plug-in Sample using OpenCL
  - video processing using VPP algorithms
- **OpenCL Video Motion Estimation Sample**  
Console application which provides step-by-step guidelines on the using Intel's motion estimation extension for OpenCL standard. The motion estimation extension includes a set of host-callable functions for frame-based Video Motion Estimation.
- **OpenCL Interoperability Sample**  
GUI application which demonstrates how to use **SDK** and Intel® OpenCL SDK together for efficient video decoding and fast post-processing.
- **Intel® Media SDK RAW Media Accelerator Sample (Camera Sample)**  
Console application that demonstrates how to use **SDK** and Intel® Media SDK RAW Media Accelerator together for efficient RAW camera data capturing and processing.
- **HEVC GPU Assist APIs Sample**  
The sample provides examples of the typical data and control flow to use the **HEVC GPU Assist APIs** effectively. Could work in two modes – as a standalone application that demonstrates patterns of new API and in a tandem with full H265 encoder (this mode could be useful for encoder debugging and testing).

Each sample includes:

- a readme file for each sub-sample
- source and header files for each sub-sample

**Samples** package has one installer for all sub-samples.

## Software & Hardware Requirements

---

### Hardware:

- Hardware requirements are the same as described in Intel® Media Server Studio Release Notes or Intel® Media Software Development Kit Release Notes (whichever samples are used with)
- (Optional) HDMI\* 1.4, eDP\* 1.1 or similar based monitor/TV as primary display
- (Optional) Active shutter glasses

### Software:

- Microsoft Windows\* 7, Microsoft Windows\* 8, Microsoft Windows\* 8.1 or Microsoft Windows\* 10
- For Microsoft DirectX\* 11 functionality - Microsoft Windows 8 or Microsoft Windows 8.1.
- Microsoft Visual C++\* 2010 or later version of Microsoft Visual C++ (if exact version is not specified in particular sample readme).
- For samples - Microsoft Windows SDK for Windows 7 or Microsoft Windows SDK for Windows 8.
- For Microsoft DirectX 11 enabled samples - Microsoft Windows SDK for Windows 8.
- Intel® Media SDK 2016 or Intel® Media Server Studio 2016
- For **OpenCL User Plug-in, OpenCL Video Motion Estimation Sample and OpenCL Interoperability Sample - CodeBuilder** (part of **Intel® Media SDK 2016 or Intel® Media Server Studio 2016**) for compilation, OpenCL driver – to run the samples.
- For PTIR feature in **Video Processing Sample - Intel® Media Server Studio 2016 Professional** edition is needed.

## Installation

---

Run the `IntelMediaSamples.msi` installer from the package to install all the samples.

## Build Instructions

---

1. `INTELMEDIASDK_WINSDK_PATH` environmental variable:
  - **Samples** depend on Microsoft\* Windows\* SDK include and library files.
  - **Samples** installer will try to set `INTELMEDIASDK_WINSDK_PATH` environment variable used in sample project files to locate those include and library files.
  - You may want to set (possibly to “”) `INTELMEDIASDK_WINSDK_PATH` variable manually (possibly with the help of `<install-folder>\samples\set_INTELMEDIASDK_WINSDK_PATH.bat`) in the following cases:
    - The variable was not set during installation due to no Microsoft Windows SDK installed or installed to a non-default location.
    - Your Microsoft Visual Studio\* environment is already set up with Microsoft Windows SDK include and library directories (e.g. via registration script for Microsoft Visual C++\* 2005). Set the variable to “” or delete it in this case.
    - You wish to use a different Microsoft Windows SDK version than was auto-detected and set at installation.
2. `INTELMEDIASDKROOT` environmental variable:
  - Samples depend on **SDK** external headers and **SDK** dispatcher library which are searched in folders `INTELMEDIASDKROOT\include` and `INTELMEDIASDKROOT \lib\<arch>` respectively.
  - `INTELMEDIASDKROOT` is set by **SDK** installer and points to the **SDK** installation folder.
3. OpenCL headers and libraries for **OpenCL User Plug-in, OpenCL Video Motion Estimation Sample and OpenCL Interoperability Sample**:
  - These samples require OpenCL headers and libraries to be available. All needed files are located in **Code Builder** (part of **Intel® Media SDK 2016 or Intel® Media Server Studio 2016**). Please install Code Builder from the package and set up `INTELOCLSDKROOT` environment variable to `<code-builder-install-folder>\include\cl\` folder.
4. Building with Microsoft Visual C++\*:
  - Use provided with each sample solution file `.sln` with Microsoft Visual C++ version 2005 or later to build the respective sample. Locate the resulting executable file in the folder `<install-folder>\_build\<PlatformName>\<ConfigurationName>`.
5. Choosing Microsoft Direct3D\* version to build with:

- If version of the installed Microsoft Windows\* SDK is 8.0 or above, then Microsoft Direct3D\* 11.1 surfaces support will be enabled in sample by default. You can enable or disable it manually using `MF_X_D3D11_SUPPORT` macros defined in `<install-folder>\sample_common\sample_defs.h`
6. Building using Debug configuration.
- `Debug_WithDebugAPI` configurations of samples are dependent on debug versions of dispatcher library (`libmfx_d.lib`). This library is an open source project, so you may build it from source code. Note, that `libmfx_d.lib` is not provided in binary form, so it has to be built manually. Source code of dispatcher library is available at **Intel® Media SDK 2016** and **Intel® Media Server Studio 2016 for Windows Server**

## Running the Software

---

Running **Samples** with Intel® OpenCL dependency:

- Make sure proper OpenCL driver is installed on the machine. Please refer to <https://software.intel.com/en-us/articles/opencl-drivers> for details.
- Add a path to `opencl.dll` from the driver installation above to `PATH` variable

## Legal Information

---

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting [Intel's Web Site](#).

MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation.

Intel, the Intel logo, Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

### Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSE3 instruction sets and

other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel.

Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

---

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

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

Copyright © 2016, Intel Corporation