

Multi-Transcoding Sample

Overview

Multi-Transcoding Sample works with **Intel® Media Server Studio 2015 for Linux Server**.

It demonstrates how to use **Media Server Studio – SDK** (hereinafter referred to as "**SDK**") API to create a console application that performs the transcoding (decoding and encoding) of a video stream from one compressed video format to another, with optional video processing (resizing) of uncompressed video prior to encoding. The application supports multiple input and output streams meaning it can execute multiple transcoding sessions concurrently.

The main goal of this sample is to demonstrate CPU/GPU balancing in order to get maximum throughput on Intel hardware-accelerated platforms (with encoding support). This is achieved by running several transcoding pipelines in parallel and fully loading both CPU and GPU.

This sample also demonstrates integration of user-defined functions for video processing (picture rotation plug-in) into **SDK** transcoding pipeline.

This version of sample also demonstrates surface type neutral transcoding (opaque memory usage).

The sample can work together with **Intel® Media Server Studio – HEVC Decoder & Encoder** (hereinafter referred to as "**HEVC**").

Features

Multi-Transcoding Sample supports the following video formats:

input (compressed)	H.264 (AVC), MPEG-2 video, VC-1, HEVC (High Efficiency Video Coding)
output (compressed)	H.264 (AVC), MPEG-2 video, HEVC (High Efficiency Video Coding)

Software and Hardware Requirements

- For general **SDK** requirements, please, see <msdk_install-folder>/media_server_studio_sdk_release_notes.pdf
- For the samples specific requirements, please, see <install-folder>/Media Samples Guide.pdf

Package Contents

Multi-Transcoding Sample package contains the following:

<install-folder>/sample_multi_transcode	
readme-multi-transcode.pdf	This file

CMakeLists.txt	CMake* configuration file
include	Header files for the sample
src	Source files for the sample

<install-folder>/sample_multi_transcode/include	
sample_multi_transcode.h	Header file for the transcoding application
pipeline_transcode.h	Header file for the transcoding pipeline class
transcode_utils.h	Header file for the sample utilities

<install-folder>/sample_multi_transcode/src	
sample_multi_transcode.cpp	Source file for the transcoding application
pipeline_transcode.cpp	Source file for the transcoding pipeline class
transcode_utils.cpp	Source file for the sample utilities

<install-folder>/sample_common	
CMakeLists.txt	CMake configuration file
include	Header files for the common sample resources
src	Source files for the common sample resources

How to Build the Application

Use the *build.pl* script located at <install-folder>. For the details on how to build samples see <install-folder>/Media Samples Guide.pdf. Shortly, you may invoke the following commands to build the sample:

```
$ export MFX_HOME=/mediasdk/installation/folder
$ cd <install-folder>
$ ./build.pl --cmake=intel64,make,release --clean
$ cd <install-folder>/__cmake/intel64.make.release && make
```

Output will be placed in the following folder: <install-folder>/__cmake/intel64.make.release/__bin/release

Running the Software

Sample is buildable in a few variants depending on LibVA backends availability and support:

- sample_multi_transcode_drm – sample variant to run on the system without Graphic Server installed (for example, X)
- sample_multi_transcode_x11 – sample variant to run under X

The executable file `sample_multi_transcode_**` (** - one of the supported backends) requires the following command-line switches to function properly:

<code>-par <par_file></code>	A parameter file is a configuration file of specific structure. It contains several command lines, each line corresponding to a single transcoding, decoding or encoding session.
------------------------------------	---

The following command-line switches are optional:

<code>-p <perf_file></code>	Instructs sample to write performance statistics to a specified file.
<code>--vmem <limit></code>	Virtual memory limit for the process in bytes.

Below is an example of command-line that can be used to execute the **Multi-Transcoding Sample**:

```
$ sample_multi_transcode_drm -par 1.par -p 1.perf
```

For each line in the parameter file, the following command-line arguments are mandatory:

<code>-i::h264 h265 mpeg2 vc1 <Input File></code> OR <code>-i::source</code>	Input video type and (compressed) video file name and path. The session receives the output of the session using the <code>-o::sink</code> option at input (only one session using <code>-o::sink</code> is allowed in a parameter file.).
<code>-o::h264 h265 mpeg2 <Output File></code> OR <code>-o::sink</code>	Output video type and (compressed) video file, name and path. Output of this session serves as input for all sessions using the <code>-i::source</code> (a parameter file can contain several sessions using <code>-i::source</code>).

Note 1: You need to have **HEVC** installed to run with `h265` codec.

The following command-line switches are optional:

<code>-b <bitrate></code>	Bitrate of the encoded stream in Kbits/second
<code>-f <framerate></code>	Frame rate of the encoded stream (30 by default)
<code>-u <quality, speed, balanced></code>	Target usage (balanced by default). This parameter specifies a trade-off between quality and speed.

<code>-w <width></code>	Width of destination video frame. If specified and the value is different from the source frame width, video preprocessing (VPP) for scaling (resizing) is invoked.
<code>-h <height></code>	Height of destination video frame. If specified and the value is different from the source frame height, video preprocessing (VPP) for scaling (resizing) is invoked.
<code>-hw -sw</code>	Implementation of SDK : platform-specific or software. If not specified, hardware is used. Software implementation should not be set since there is no software library on Linux platforms.
<code>-join</code>	Join the session to another session. If a parfile contains several lines with a <code>-join</code> switch, the first session in the list acts as parent and all other sessions are joined to the parent. Option is effective only with SDK API 1.1 and higher.
<code>-priority <0,1,2></code>	Set priority for the session (available only if session is joined): <ul style="list-style-type: none"> 0 = Low 1 = Normal (default) 2 = High
<code>-n <num></code>	The number of frames to process
<code>-async <depth></code>	Depth of asynchronous pipeline. 1 by default.
<code>-angle 180</code>	Invokes sample plug-in for 180 degrees picture rotation, CPU implementation by default or Intel® OpenCL™ implementation if <code>-opencl</code> option is specified(see below). For CPU implementation file <code>sample_rotate_plugin.so</code> must be available.
<code>-opencl</code>	Invokes Intel® OpenCL™ implementation of 180 degrees picture rotation. Option <code>-angle 180</code> must be specified. File <code>libsampl_plugin_openc1.so</code> must be available. File <code>ocl_rotate.cl</code> must exist in the local folder when running the application with this option.
<code>-deinterlace</code>	Enables deinterlacing for an interlaced input stream.
<code>-l <num_slice></code>	Number of slices in each video frame. 0 by default. If <code>num_slice</code> equals zero, the encoder may choose any slice partitioning allowed by the codec standard.
<code>-mss <slice_size></code>	Maximum slice size in bytes. Supported only with hardware library (<code>-hw</code>) and H.264 encoder. This option is not compatible with <code>-l</code> .
<code>-la</code>	Use the look ahead bitrate control algorithm (LA BRC) for H.264 encoder.
<code>-lad <depth></code>	Depth parameter for the LA BRC, the number of frames to be analyzed before encoding. In range [10,100].

set

This option could be applied if you want use same user codec plugin in multiple sessions. This plugin will be used for all sessions defined below. You can specify plugin as 32-character hexadecimal guid string or path to .so.

The command-line interface allows 2 usage models (which can be mixed within one parameter file):

1. Multiple intra-session transcoding: several transcoding sessions, any number of sessions can be joined. Each session includes decoding, preprocessing (optional), and encoding.
2. Multiple inter-session transcoding: output of a single decoding session serves as input for several encoding sessions. Either all or none of the sessions are joined. Any of the encoding sessions can optionally include preprocessing (resizing).

Below are several examples of parameter file contents.

Single intra-session transcoding:

```
-sw -i::mpeg2 input.mpeg2 -async 10 -o::h264 output.h264 -n 100 -w 320 -h 240 -f 30 -b 2000 -u speed
```

Multiple intra-session transcoding, several sessions joined:

```
-sw -i::mpeg2 input1.mpeg2 -async 10 -o::mpeg2 output1.mpeg2
-hw -i::h264 input2.h264 -o::mpeg2 output2.mpeg2 -join
-hw -i::h264 input3.h264 -o::mpeg2 output3.mpeg2 -join
```

Multiple inter-session transcoding, all sessions joined:

```
-hw -i::h264 input.h264 -o::sink -join
-sw -o::mpeg2 output1.mpeg2 -i::source -join -w 640 -h 480
-sw -o::mpeg2 output2.mpeg2 -async 2 -u 3 -i::source -join
```

Mixed model:

```
-sw -i::vc1 input.mpeg2 -async 10 -o::h264 output.h264 -n 100 -w 320
-h 240 -f 30 -b 2000 -u speed
-hw -i::h264 input.h264 -o::sink -join
-sw -o::mpeg2 output1.mpeg2 -i::source -join -w 640 -h 480
-sw -o::mpeg2 output2.mpeg2 -async 2 -u 3 -i::source -join
```

Multiple intra-session transcoding, same **HEVC** is used in both cases:

```
set -i::h265 15dd936825ad475ea34e35f3f54217a6
-sw -i::h265 input1.265 -o::h264 output1.264
-sw -i::h265 input2.265 -o::mpeg2 output2.mpeg2
```

Multiple intra-session transcoding, **HEVC** is used:

```
set -i::h265 /path/to/so/decoder_plugin.so
set -o::h265 /path/to/so/encoder_plugin.so
-sw -i::h265 input1.265 -o::h264 output1.264
-sw -i::mpeg2 input2.mpeg2 -o::h265 output2.265
```

Please, also pay attention on “Running the Software” section of <install-folder>/Media Samples Guide.pdf document where you will find important notes on backend specific usage (drm and x11).

Tips

1. To achieve maximum throughput use `-async >= 5` and the `-join` option when running several transcoding pipelines.
2. If you need only one transcoding session you can avoid creating a par file and pass the arguments of this session to the application using command line. E.g.:

```
sample_multi_transcode_drm -i::mpeg2 input.mpeg2 -async 10 -  
o::h264 output.h264 -n 100 -w 320 -h 240 -f 30 -b 2000 -u speed -  
p 1.perf -hw
```

Known Limitations

- Configurations <multiple joined inter-session transcoding where one of the encoders is MPEG2> are not supported when sample application uses platform-specific **SDK** implementation on systems with Intel® HD Graphics 3000/2000 and 4000/2500. Application can exit with error or hang. An example of a corresponding par file is given below:

```
-hw -i::h264 input.h264 -o::sink -join  
-hw -o::mpeg2 output1.mpeg2 -i::source -join  
-hw -o::h264 output2.h264 -i::source -join
```

Systems with Intel® Iris™ Pro Graphics, Intel® Iris™ Graphics and Intel® HD Graphics 4200+ Series are free from this limitation.

- **Multi-Transcoding Sample** if run with `-f` option does not perform frame rate conversion. Specified frame rate simply overrides frame rate value of the original stream and is used as decoder and encoder setting.
- Sample architecture is to host a VPP component in each encoding session. For 1:N pipeline if each encoding session needs to do same VPP operation, e.g. de-interlacing, this is not optimal because the same operation will be done N times. If you are building your own application with pipeline similar to this sample you should insert an additional VPP into decoding session to optimize such scenarios.

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 and the Intel logo 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