

Video Conferencing Sample

[Overview](#)

[Features](#)

[Software Requirements](#)

[Package Contents](#)

[Application Data Flow](#)

[How to Build the Application](#)

[Running the Software](#)

[Known Limitations](#)

[Legal Information](#)

Overview

Video Conferencing 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 features, intended for generic video conferencing tasks.

- How to configure **SDK** Encode for generating low latency bit stream
- How to recover from situation when receiving part couldn't decode some frame due to network packets loss
- How to change target bitrate of **SDK** Encode dynamically
- How to force **SDK** Encode to insert a key frame at particular position out of predefined GOP structure
- How to force **SDK** Encode to create long term reference from particular frame
- How to use per frame mode QP (quantization parameter)
- How to enable temporal scalability
- How to enable Reference Picture Marking Repetition SEI
- How to change encoding resolution dynamically
- How to measure latency of **SDK** Encode
- How to enable **SDK** Encode rolling intra refresh feature

Features

Video Conferencing Sample supports the following video formats:

input (uncompressed)	YUV420
output (compressed)	H.264 (AVC)

Note: For format YUV420, **Video Conferencing Sample** assumes the order Y, U, V in the input file.

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

Video Conferencing Sample package contains the following:

<install-folder>/sample_videoconf/

readme-videoconf.pdf	This file
CMakeLists.txt	CMake* configuration file
include	Header files for the sample
src	Source files for the sample

<install-folder>/sample_videoconf/include/

interface.h	Interface of pipeline
actions.h	Implementation of dynamic encoding events
action_processor.h	Helper class to organize events based on types and activation frame number
brc.h	Sample Implementation of BitRate Control (BRC) using FrameMode CQP
video_conf_pipeline.h	Declaration of video conferencing pipeline class

<install-folder>/sample_videoconf/src/

video_conf_pipeline.cpp	Implementation of video conferencing pipeline class
sample_video_conf.cpp	Input parameters parsing, and pipeline launching

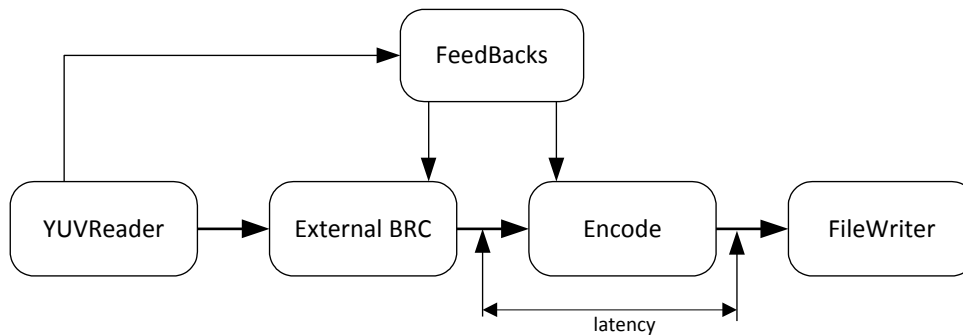
<install-folder>/sample_common/

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

Application data flow

In **Video Conferencing Sample** feedbacks on the quality of encoding are imitated and created during initialization stage based on command line parameters, while in a real conferencing application they are to be created dynamically based on the receiver reports.

The diagram below describes the flow of the data in the application:



1. YUV Frame is read from a file
2. Feedbacks buffer is examined for available feedbacks. If some feedbacks are ready for the current frame, they are applied to change Encode or External BRC settings
3. Latency timer started
4. YUV frame is encoded
5. Latency timer stopped
6. Encoded frame is written to a file

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

The executable file requires the following command-line switches:

-i <InputFile>	Input (uncompressed) video file, name and path
-o <OutputFile>	Output (compressed) video file, name and path
-w <width>	Width of input video frame
-h <height>	Height of input video frame

The following command-line switches are optional:

-b <bitrate>	Bitrate of the encoded stream in Kbits/second
-f <framerate>	Frame rate of the encoded stream (30 by default)
-hw	Use platform-specific implementation of SDK (default)
-sw	Use software implementation of SDK . Should not be set since there is no software library on Linux platforms.
-bs <frame, scale>	Calculate new target bitrate as current target bitrate * scale . If external bitrate control enabled(-brc), it will be reset with new target value prior to encoding of frame , no changes to SDK encode will be done. If external bitrate control disabled, encoder itself is reset with new target bitrate prior to encoding of frame . Note: the behavior of SDK encode with regards to bitrate change is described in Appendix C of mediasdk-man.pdf.
-bf <frame, broken_frame>	Simulates the situation when video conferencing receiver can not decode brokenFrame and this information reached video conferencing sender prior to encoding of frame . Note: the behavior SDK encode with regards to rejected reference list selection is described in Appendix C of mediasdk-man.pdf.
-gkf <frame_num>	Generates IDR frame at specific position.
-ltr <frame_num>	Marks specific frame_num as longterm reference. Note: the behavior of SDK encode with regards to preferred and long-term references lists selection is described in

	Appendix C of mediasdk-man.pdf.
-ts <num_layers>	Will create up to 4 temporal layers. Note: the behavior of SDK encode with regards to temporal layer creation is described in mfxExtAvcTemporalLayers structure description in mediasdk-man.pdf.
-brc	Enables external bitrate control based on perframe QP
-l0 <frame> <l0_len>	Specifies number of reference frames in L0 array for encoding frame
-latency	Calculates per frame latency of SDK Encode and writes to the standard output along with respective frame types (I, P). Note: to estimate latency of generic videoconferencing scenario (encode + decode), per frame latency values for SDK Decode can be obtained using Decoding Sample with input stream encoded by Video Conferencing Sample .
-ir <cycle_size, qp_delta>	Specifies that intra frame will be completely refreshed in cycle_size number of pictures. This value is taken from the interval [3; 29] qp_delta defines the quantization parameter difference for the inserted intra macroblocks. The valid values for this parameter belong to the interval [-51; 51]
-par <parameters_file>	Specifies parameters file that may contain any of supported options.
-?	Prints help.

Below are examples of command-lines that can be used to execute **Video Conferencing Sample**:

```
sample_videoconf.exe -i input.yuv -o output.h264 -w 720 -h 480 -b 10000 -hw -bs 10 2
```

```
sample_videoconf.exe -i input.yuv -o output.h264 -w 720 -h 480 -b 10000 -hw -bf 10 8
```

```
sample_videoconf.exe -i input.yuv -o output.h264 -w 720 -h 480 -b 10000 -hw -ltr 10
```

Known Limitations

- Frames might be not fully restored after encoding with intra refresh (-ir).

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