Media Samples Installation Guide

Software & Hardware Requirements

Hardware:

- IA-32 or Intel® 64 architecture processors with support for Intel® Streaming SIMD Extensions 2 instructions.
- For S3D display functionality using igfx_s3dcontrol library (**Video Decoding Sample**, **Transcoding Sample** using Microsoft* DirectShow*):
  - 2nd Generation Intel® Core™ Processors with Intel® HD Graphics 3000/2000 or later
  - HDMI* 1.4, eDP* 1.1 or similar based monitor/TV as primary display
  - Active shutter glasses

Software:

- See `<msdk_install-folder>/media_server_studio_sdk_release_notes.pdf` for **SDK** general requirements. To build **Samples** you additionally need the following components to be installed and properly configured on the system:
- **For CentOS* 7.0:**

```
$ sudo yum install gcc g++ make cmake perl xserver-xorg-dev
```

- **For SLES* 11 SP3:**

```
$ sudo zypper install gcc46 gcc46-c++ make cmake perl xorg-x11-devel
$ ln -s gcc-4.6 gcc
$ ln -s g++-4.6 g++
```

- Samples can be built with GCC/G++ compiler version 4.6 and CMake* version 2.6.2 or higher.
- For **Splitters and Muxers Sample** and **Full Transcoding Sample** you will also need several additional dynamic libraries which are the part of FFmpeg* codec libraries, particularly:

```
libavutil, version 52.38.100
libavcodec, version 55.18.102
libavformat, version 55.12.100
```

  You can install them from the package manager or build from sources. Please, check the official compilation guide at *https://trac.ffmpeg.org/wiki/CompilationGuide* for build instructions.
- For samples with OpenCL (**Video Encoding, Video Transcoding, Video Motion Estimation, Interoperability**) it is required to install **Intel® Media Server Studio – Code Builder** and **Intel® Media Server Studio – Graphics Drivers**.

Build Instructions

To build samples the following environment variable should be setup:

```
$ export MFX_HOME=/mediasdk/installation/folder
```

Go to the samples directory and execute `build.pl` script without arguments to see the help:

```
$ ./build.pl
Copyright (c) 2014 Intel® Corporation. All rights reserved.
This script performs Samples projects creation and build.
Usage: perl build.pl --cmake=ARCH,GENERATOR,CONFIG [--clean] [--build]
```

```
Possible variants:
 ARCH = intel64
 GENERATOR = make
 CONFIG = debug | release
Environment variables:
 MFX_HOME=/path/to/mediasdk/package # required
 MFX_VERSION="0.0.000.0000" # optional
Optional flags:
 --clean - clean build directory before projects generation / build
 --build - try to build projects before generation (requires
cmake>=2.8.0)
Examples:
 perl build.pl --cmake=intel64,make,debug [ only
generate projects ]
 perl build.pl --cmake=intel64,make,debug --build [ generate
and then build ]
 perl build.pl --cmake=intel64,make,debug --build --clean [ generate,
clean and build ]
```

Script invokes specified CMake* projects generator and optionally builds them (option available for cmake>=2.8.0). At the moment only make files generator for UNIX-like systems is supported. Project files will be placed in the folder named by the requested configuration; for example:

```
/__cmake
intel64.make.release
intel64.make.debug
```

To build generated project files use generator-specific approaches. For example, to build samples from make files invoke:

```
$ make -C <install-folder>/__cmake/intel64.make.release
```

With CMake older than 2.8.0 all samples can be built at once with the following command:

```
$ ./build.pl --cmake=intel64,make,release --clean --build
```

Binaries will appear in the following folder:

```
__cmake/intel64.make.release/__bin/release/
```

Samples with Hardware Acceleration support are buildable in a few variants depending on the availability of LibVA backends. For example:

- `sample_full_transcode_drm` – sample variant with HW acceleration support to be run on the system without Graphic Server installation (i.e. LibVA DRM backend is used).
- `sample_full_transcode_x11` – sample variant with HW acceleration support to be run under X Server (i.e. LibVA X11 backend is used).

Running the Software

DRM backend specific notes:

- For application to work thru DRM application should be authorized to access graphics card. VA-API DRM backend supports 2 authentication models:
  - The first model can be applied on the system with no installation of Graphic Server. In this case you need root privileges to run:

    ```
    $ sudo LD_LIBRARY_PATH=$MEDIASDK_INSTALL_FOLDER/bin/x64 \
    ```

```
$ sample_full_transcode_drm h264 -i input.264 -o output.yuv –d3d –hw
```

- The second model assumes that X server is installed and running. In this case DRM authentication will actually go thru LibVA X11 backend and, thus, thru X server which already has access to the graphic card. The only thing user should be sure in is that he is logged on to the X server (or has access) and `DISPLAY` environment variable is set properly. For example:

```
$ export DISPLAY=:0.0
$ sudo LD_LIBRARY_PATH=$MEDIASDK_INSTALL_FOLDER/bin/x64 \
$ sample_full_transcode_drm h264 -i input.264 -o output.yuv –d3d –hw
```

- It can be noted that DRM-itself authentication can still be tried out even with running X server, but you need to remove `DISPLAY` environment variable and use root privileges:

```
$ export -n DISPLAY
$ sudo LD_LIBRARY_PATH=$MEDIASDK_INSTALL_FOLDER/bin/x64 \
$ sample_full_transcode_drm h264 -i input.264 -o output.yuv –d3d –hw
```

X11 backend specific notes:

- To use this backend user should be sure that he is logged into X server or is allowed to make connections to the X server
- If user is allowed to use X and logged into machine remotely (thru SSH) he needs `DISPLAY` environment variable properly set. For example:

```
$ export DISPLAY=:0.0
$ sample_full_transcode_x11 h264 -i input.264 -o output.yuv –d3d –hw
```

Legal Information

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
