Intel® Media Software Development Kit 2013 Release Notes

(Version 4.0.0000554.52230)

Overview

The **Intel® Media Software Development Kit** (Intel® Media SDK) is a software development library that exposes the media acceleration capabilities of Intel® platforms for decoding, encoding and video preprocessing. The API library covers a wide range of Intel platforms. The Intel Media SDK targets general application developers who want to integrate encoding and decoding into their applications.

Please see the "`<install-folder>\doc\MediaSDK Sample Guide.pdf`" for information on sample source code provided with the Intel Media SDK.

New Features

Intel® Media SDK 2013 introduces API version 1.6. This version is backwards compatible with API version 1.4. API version 1.6 introduces extensions to support JPEG*/Motion JPEG Encoding (for details please see "`<install-folder>\doc\mediasdkjpeg-man.pdf`").

Intel® Media SDK 2013 also introduces the source code of the static dispatcher library and a GUI utility Intel Media SDK Samples Browser which allows developers to navigate across samples and learn about their features.

This release also includes the following changes to API version 1.4 released in Intel® Media SDK 2012 R2:

- `mfxExtJPEGQuantTables` and `mfxExtJPEGHuffmanTables` structures were added for JPEG encoder and decoder initialization.
- Enumerator for `mfxFrameData::Corrupted` was extended with `MFX_CORRUPTION_ABSENT_TOP_FIELD` and `MFX_CORRUPTION_ABSENT_BOTTOM_FIELD` flags to signal field absence.

- `mfxInfoMFX` was extended with the `SliceGroupsPresent` field to indicate that slice groups are present in the bitstream.
- `mfxExtCodingOption2` with the `IntRefTypefield, IntRefType, IntRefCycleSize, IntRefQPDelta, MaxFrameSize, BitrateLimit` fields was added. It can be used as extended buffer to configure rolling intra refresh feature of H.264 encode and advanced bitrate control settings.
- `MFX_BITSTREAM_EOS` value was added to `BitstreamDataFlag` enumerator to indicate that input bitstream buffer contains the end of the stream.
- `mfxExtVPPImageStab` structure was added to configure VPP image stabilization filter algorithm (used as extended buffer). `MFX_IMAGESTAB_MODE_UPSCALE` and `MFX_IMAGESTAB_MODE_BOXING` enumerators specify stabilization mode `mfxExtVPPImageStab::Mode`.
- Support of `MFX_LEVEL_AVC_52` was added.
- Extended bitrate control support `mfxExtCodingOption2::MBBRC` and `mfxExtCodingOption2::ExtBRC` was added.
- `mfxBitstream::DecodeTimeStamp` was added.
- `mfxExtVppAuxData::PicStruct` picture structure detection for H.264 PAFF support was added.
- IECP declaration was removed.

Please see the Intel Media SDK Reference Manual for details "`<install-folder>\doc\mediasdk-man.pdf`"

For information on the USER class please see "`<install-folder>\doc\mediasdkusr-man.pdf`"

For information on Multi-view Video Coding support please see "`<install-folder>\doc\mediasdkmvc-man.pdf`"

Additionally, the Intel Media SDK package now contains a standalone library which exposes an Application Programming Interface (API) for Stereoscopic 3D (S3D) rendering with Intel® HD Graphics 3000/2000 and later. Please see Package Contents section for locations of header `igfx_s3dcontrol.h` and library `igfx_s3dcontrol.lib` files.

This library can be used either independently or along with any version of the Intel Media SDK library to utilize, for instance, decoding functionality.

System Requirements

Hardware

- IA-32 or Intel® 64 architecture processors with support for Intel® Streaming SIMD Extensions 2 instructions.
- 200 MB free hard disk space.

- For S3D display functionality using `igfx_s3dcontrol` library:
    1. 2nd Generation Intel® Core™ Processors with Intel® HD Graphics 3000/2000 or later
    2. HDMI* 1.4, eDP* 1.1 or similar based monitor/TV as primary display
    3. Active shutter glasses

Software

- Microsoft* Windows* 7 or Microsoft Windows 8.
- For Microsoft DirectX* 11.1 functionality - Microsoft Windows 8.
- Microsoft Visual C++* 2005 with Service Pack 1, or later version of Microsoft Visual C++.
- For samples - Microsoft Windows SDK for Windows 7 or later.
- For Microsoft DirectX 11.1 enabled samples - Microsoft Windows SDK for Windows 8.
- For Intel® OpenCL™ User Plug-in sample - Intel® SDK for OpenCL™ Applications 2012.

Package Contents

Note: The suffix `<arch>` indicates 32- or 64-bit Microsoft* Windows* (either "`win32`" or "`x64`"). Both are always installed to allow for cross-platform development.

`<install-folder>`	Intel® Media SDK Release Notes (this file), Pre-release End User License Agreement (EULA) "`Intel Media SDK Pre-release EULA.rtf`"
`<install-folder>\bin\<arch>`	Intel® Media SDK Dynamic Library, software implementation: `libmfxsw32.dll` for IA-32 architecture `libmfxsw64.dll` for Intel® 64 architecture
`<install-folder>\doc`	Intel® Media SDK documentation: • Intel® Media SDK Reference Manual `mediasdk-man.pdf` • Intel® Media SDK Extensions for User-Defined Functions `mediasdkusr-man.pdf` • Intel® Media SDK Extensions for Multi-view Video Coding `mediasdkmvc-man.pdf` • Intel® Media SDK Extensions for JPEG*/Motion JPEG

	mediasdkjpeg-man.pdf • Microsoft DirectShow* Filter Interfaces and Specifications MediaSDK Filters Specifications.pdf • Samples Overview MediaSDK Sample Guide.pdf • Intel® Media Developer's Guide Intel_Media_Developers_Guide.pdf • Intel® Media SDK Library Distribution and Dispatching Process mediasdk-distrib.pdf
<install-folder>\ include	External Intel® Media SDK headers: • Type definitions in mfxdefs.h • Structure definitions in mfxstructures.h • Function definitions in C in mfxvideo.h • C++ wrapper of the SDK functions in mfxvideo++.h • Extensions for Multi-view Video Coding options mfxmvc.h • Extensions for User-Defined Functions mfxplugin.h • C++ wrapper for User-Defined Functions mfxplugin++.h • Extensions for JPEG*/Motion JPEG Video coding options mfxjpeg.h
<install-folder>\ lib\ <arch>	• Static Dispatcher Library libmfx.lib • Static Dispatcher Library libmfxmd.lib build with /MD compiler option to compatibility with applications using the multithread- and DLL-specific version of the run-time library
<install-folder>\ igfx_s3dcontrol\ include	• S3D API definitions igfx_s3dcontrol.h
<install-folder>\ igfx_s3dcontrol\ lib\<arch>	• Static S3D Control Library igfx_s3dcontrol.lib
<install-folder>\ igfx_s3dcontrol\	• Displaying S3D with Intel® HD Graphics Developers Guide Displaying S3D with Intel HD Graphics.pdf

<install-folder>\ samples\	Contains the following source code samples: • Intel® Media SDK Encoding Sample in folder sample_encode • Intel® Media SDK Decoding and S3D Rendering Sample in folder sample_decode. A script for building a video wall application using this sample is provided. • Intel® Media SDK Transcoding Sample in folder sample_multi_transcode • Intel® Media SDK Video Processing Sample in folder sample_vpp • Intel® Media SDK Rotation Plug-in Sample in folder sample_user_modules\rotate_cpu • Intel® Media SDK OpenCL™ Plug-in Sample in folder sample_user_modules\rotate_opencl • Intel® Media SDK VPPPlugin Utility Class in folder sample_utilities\vpp_plugin • Intel® Media SDK Application Sample using Microsoft DirectShow in folder sample_dshow_player • Intel® Media SDK Plug-Ins Sample using Microsoft DirectShow in folder sample_dshow_plugins • Intel® Media SDK Application Sample using Microsoft Multimedia Framework Plug-ins in folder sample_studio • Intel® Media SDK Video Conferencing Sample in folder sample_videoconf • Intel® Media SDK Plug-Ins Sample using Microsoft Media Foundation* in folder sample_mfoundation_plugins • Intel® Media SDK Transcoding Sample using Microsoft Windows* 8 User Interface in folder sample_win8ui_transcode
<install-folder>\ samples_bin\<arch>	Pre-built binaries of installed sample applications • If installed, console sample application binaries: sample_encode.exe sample_decode.exe (with sample_video_wall.bat) sample_vpp.exe sample_multi_transcode.exe sample_videoconf.exe • If installed, user plug-in sample binaries: sample_rotate_plugin.dll sample_plugin_opencl.dll (with ocl_rotate.cl) • If installed, Microsoft DirectShow sample binary application

	sample_dshow.exe • If installed, Microsoft DirectShow filters h264_dec_filter.dll h264_enc_filter.dll mpeg2_dec_filter.dll mpeg2_enc_filter.dll vc1_dec_filter.dll mvc_dec_filter.dll jpeg_dec_filter.dll imc_*.dll • If installed, Microsoft Multimedia Framework sample binary application sample_studio.exe sample_studio_builder.exe sample_studio_profiles.txt
<install-folder>\ tools\	Contains the following tools in binary form: • Intel® Media SDK Tracer in folder mediasdk_tracer. This utility performs runtime recording of Intel Media SDK API calls and parameters to a log file. • Intel® Media SDK System Analyzer in folder mediasdk_sys_analyzer. This utility analyzes the system and reports back Intel Media SDK related capabilities, graphics driver and components status. • Intel® Media SDK Samples Browser in folder mediasdk_browser. This is a GUI utility which allows developers to navigate across samples and learn about their features.
<install-folder>\ opensource\	Source code of Intel® Media SDK dispatcher

Installation

1. Installation requires full administrative rights.
2. Extract files from the .ZIP file to the target hard drive.
3. Select the installer that matches the target system architecture:

File Name for Installer	Target System Architecture

`Intel(R)_Media_SDK_win32.msi`	IA-32 architecture running Microsoft* Windows* 7 or Microsoft Windows 8 operating system
`Intel(R)_Media_SDK_x64.msi`	Intel® 64 architecture running Microsoft* Windows* 7 or Microsoft Windows 8 64-bit operating system

Note: If `win32` is installed, the `x64 lib` and `bin` files are also present to enable cross-platform development and vice versa.

Known Limitations

The Intel® Media SDK library has the following known limitations:

- The Intel Media SDK dispatcher `libmfx.lib` is best used with a standard DLL entry point (as recommended by Microsoft*) when used in a DLL application such as a Microsoft DirectShow* filter. The DLL entry point setting can be found under the `Link` > `Advanced` compiler options. Non-standard entry points can be used, but are not recommended.
- Loading of library `libmfxsw32.dll`/`libmfxsw64.dll` not through the dispatcher is unsafe.
- Using the Intel Media SDK in parallel with Intel® Threading Building Blocks could impact performance.
- Frames for different views in single AU in MVC encoder must be provided to encoder in order specified by `mfxMVCViewDependency`.
- `MFX_EXTBUFF_AVC_REFLIST_CTRL` and `MFX_EXTBUFF_CODING_OPTION_SPSPPS` external buffers are not supported by MVC encoder.
- MVC encoder supports `MFX_PROFILE_AVC_STEREO_HIGH` only.
- H.264 encoder doesn't support processing of `mfxExtPictureTimingSEI` template. During initialization 0xFFFF values will be reset to default values. In runtime 0xFFFF values will be put to bitstream as is.
- Known limitations for H.264 Multiple-Segment Encoding:
    1. Hardcoded HRD parameters: `bit_rate_scale = 0, cpb_size_scale = 3`
    2. Encoded `bit_rate_value_minus1, bit_rate_scale` represent BitRate from original SPS within precision of kbps (maximum supported BitRate is `2^16 – 1` kbps).
    3. Encoded `cpb_size_value_minus1, cpb_size_scale` represent CpbSize from original SPS within precision of Kb (maximum supported CpbSize is `2^16 – 1` Kb).
    4. Encoded `time_scale, num_units_in_tick` could be both multiplied by 2 if the time_scale from original SPS is odd.
    5. Conflicts between SPS/PPS and `mfxVideoParam` for parameters that are not covered by SPS/PPS could lead to change of parameters in SPS/PPS.

- `RefPicMarkRepSEI` syntax is not supported by MVC encoder.
- H.264 encoder is known to be slower on `x64` platform than on `win32` platform.
- If the MPEG-2 Video encoder `mfxVideoParam::mfxInfoMFX::CodecProfile` is initialized to 0, then the stream will be encoded as `MFX_PROFILE_MPEG2_MAIN`. Additionally if the MPEG-2 Video encoder `mfxVideoParam::mfxInfoMFX::CodecLevel` is initialized to `0`, then the stream will be encoded as `MFX_LEVEL_MPEG2_MAIN`.
- `MFX_FRCALGM_DISTRIBUTED_TIMESTAMP` is unsupported by InverseTelecine and Deinterlace (60i->60p) VPP filters.
- Target usages 4 and 5 of H.264/MVC encoders are known to be non-monotonic in terms of encoding FPS versus encoding quality.
- H.264 decoder may consume more than 1 frame from the input bitstream and then propagate same timestamp to all of the consumed frames. If accurate time stamp handling is required the application has to make sure that it doesn't store more than one-frame wise data in the input bitstream.
- Target usage 7 of H.264/MVC encoders is known to have a non-monotonic quality vs. bitrate dependency.
- MPEG2 Video, VC-1 and MVC decoders are not optimized for low delay of output frames.
- MVC encoder ignores any user SEI messages for the dependent view.
- `MFX_CORRUPTION_ABSENT_TOP_FIELD`, `MFX_CORRUPTION_ABSENT_BOTTOM_FIELD`, `MFX_BITSTREAM_EOS` are not supported by VC-1, MPEG2 Video and JPEG decoders.
- VPP always uses simple FRC algorithm based on repeat/drop frames and ignores `MFX_FRCALGM_FRAME_INTERPOLATION` flag.
- The feature set of JPEG decoder/encoder is limited to the following:
    1. Baseline mode only
        - DCT based
        - 8-bit samples
        - sequential
        - loadable 2 AC and 2 DC Huffman tables
        - 3 loadable quantization matrixes
        - interleaved and non-interleaved scans
        - single and multiple scans
    2. No extended, lossless and hierarchical modes
        - no 12-bit samples
        - no progressive
        - no arithmetic coding
        - no 4 AC and 4 DC Huffman tables

- H.264 encoder and decode is known to be a little bit slower compared with Intel® Media SDK 2012 R2.
- The output AVC and MVC streams contain SPS and PPS headers before IDR frames only.
- `mfxExtCodingOption2::MBBRC` and `mfxExtCodingOption2::ExtBRC` are not supported by encoders.
- `mfxExtVppAuxData::PicStruct` isn't supported by encoders and VPP.
- VPP scaling for RGB32 interlaced content may produce slightly blurred frames.
- `Pitch` value of `mfxFrameData` structure is limited by 65535; therefore maximum width of RGB32 surface is 16383.
- JPEG decoder does not set `Corrupted` flag of `mfxFrameData` structure, and does not accept `MFX_BITSTREAM_EOS` as `DataFlag` of `mfxBitstream` structure.

Other Limitations

- Limitations related to source code samples are discussed in their corresponding readme files. See "`<install-folder>\doc\MediaSDK Sample Guide.pdf`" for an overview of the samples and additional documentation.

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