

Intel® Media Software Development Kit 2017 R1 Release Notes (Version 7.0.0.487)

[Overview](#)

[New Features](#)

[System Requirements](#)

[Package Contents](#)

[Installation](#)

[Known Limitations](#)

[Legal Information](#)

Overview

The **Intel® Media Software Development Kit*** (further referred to as the Media SDK) is a software development package that exposes the media acceleration capabilities of Intel® platforms for video and audio decoding, encoding, RAW video and photo processing as well as capabilities for capturing screen content. The API covers a wide range of Intel platforms. The Media SDK targets general application developers who want to integrate accelerated media into their applications.

New Features

The Intel® Media SDK 2017 R1 introduces API version 1.23. This version is backward compatible with the previous API versions.

Following fixes, improvements and features were added for 6th Generation Intel® Core™, 7th Generation Intel® Core™, Intel® Celeron® and Pentium®.

- API updates comparing to 1.19:
 - mfxcamera.h: mfxExtCamTotalColorControl structure added to control YUV->YUV conversion filter.
 - mfxcamera.h: mfxExtCamCscYuvRgb structure added to control conversion from RGB to YUV color format.
 - mfxplugin++.h: QueryPlatform function added to get current platform codename in application via plugin.
 - mfxstructures.h: MFX_PICSTRUCT_FIELD_SINGLE, MFX_PICSTRUCT_FIELD_TOP, MFX_PICSTRUCT_FIELD_BOTTOM, MFX_PICSTRUCT_FIELD_PAIRED_PREV, MFX_PICSTRUCT_FIELD_PAIRED_NEXT added for specify picture type.

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

Page 1 of 20

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

Copyright © 2007-2017, Intel Corporation

- mfxstructures.h: MFX_REFRESH_NO, MFX_REFRESH_VERTICAL, MFX_REFRESH_HORIZONTAL, MFX_REFRESH_SLICE added for intra refresh support.
- mfxstructures.h: mfxExtCodingOption3::RepartitionCheckEnable added to control AVC encoder attempts to predict from small partitions.
- mfxstructures.h: mfxExtCodingOption3::AdaptiveMaxFrameSize added to control max size of P and B frames.
- mfxstructures.h: mfxExtCodingOption3::EnableMBForceIntra and mfxExtMBForceIntra structure added for force specified macroblocks to be encoded as intra.
- mfxstructures.h: mfxExtCodingOption3::LowDelayBRC added to specifies frame size tolerance for rate controls: VBR, QVBR, VCM.
- mfxstructures.h: mfxExtCodingOption3::BRCPanicMode added to control panic mode in MPEG2 encoders.
- mfxstructures.h: mfxExtEncoderROI::ROI Mode, mfxExtEncoderROI::ROI::DeltaQP and MFX_ROI_MODE_PRIORITY, MFX_ROI_MODE_QP_DELTA added to control QP adjustment for ROIs.
- mfxstructures.h: mfxExtDecVideoProcessing structure added to control decoder to resize output frames via fixed function resize engine.

For detailed API description please refer to mediasdk-man.pdf and mediasdkusr-man.pdf for plugin API.

- Common:
 - Added Control Flow Guard
- HW AVC Encode:
 - Improved dynamic intra refresh.
 - Slice based intra refresh support added.
 - Low delay BRC and sliding window BRC support added.
 - Max frame size I/P support added.
 - Explicit weighted P prediction and weighted B prediction support added.
 - Fade detection support added as preview for latest version driver 15.45.
 - Control RGB input color space via mfxExtVideoSignalInfo.

- Forced intra based on ROIs support added for 6th Generation Intel® Core™, Intel® Celeron® and Pentium® with latest version driver 15.45.
 - Scene change quality improvement.
- HW HEVC Encode:
 - Added support max frame size BRC constraints.
- HW RAW Accelerator Plug-in
 - Added support total color conversion.
 - Added support control color space conversion from RGB to YUV.
- Tracer updated to 1.23 API.

All the new features listed above are not supported by software implementation of the Media SDK Library.

In a particular platform specific hardware implementation of the Media SDK Library some of the features may also be unsupported. Make sure to call `Query` functions to check the actual support.

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

Please see the RAW Media Accelerator plug-in Reference Manual for details "`<install-folder>\doc\mediasdk-raw-accelerator-man.pdf`"

Please see the Screen Capture plug-in Reference Manual for details "`<install-folder>\doc\mediasdkscreenap-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`"

For information on JPEG*/Motion JPEG Video Coding support please see "`<install-folder>\doc\mediasdkjpeg-man.pdf`"

System Requirements

Hardware

- IA-32 or Intel® 64 architecture processors with support for Intel® Streaming SIMD Extensions 2 instructions – for running software implementation of the Media SDK Library.

- Hardware acceleration is available through hardware implementations of the Media SDK Library coming with Intel® Iris™ and HD Graphics Driver for Microsoft* Windows* 7/8.1/10 on platforms with:
 - 3rd Generation Intel® Core™ processors,
 - 4th Generation Intel® Core™ processors,
 - 5th Generation Intel® Core™ processors,
 - 6th Generation Intel® Core™ processors,
 - 7th Generation Intel® Core™ processors,
 - Intel® Core™ M processors,
 - Selected SKUs of Intel® Celeron™ and Intel® Pentium™ processors with Intel HD Graphics which support Intel® Quick Sync Video.
 - Selected SKUs of Intel® Atom™ processors with Intel® HD Graphics which support Intel Quick Sync Video.
- 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, Microsoft Windows 8.1, Microsoft Windows 10 and updates.
- For Microsoft DirectX* 11 functionality - Microsoft Windows 8.1, Microsoft Windows 10 and updates
- Microsoft Visual C++* 2005 with Service Pack 1, or later version of Microsoft Visual C++.

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.

<code><install-folder></code>	The Media SDK Release Notes (this file), End User License Agreement (EULA) <code>"Media_SDK_EULA.rtf"</code> , a license file <code>license.txt</code>
<code><install-folder>\bin\<arch></code>	The Media SDK Dynamic Library, software implementation: <code>libmfxsw32.dll</code> for IA-32 architecture <code>libmfxsw64.dll</code> for Intel® 64

	<p>architecture</p> <p>version 7.0.0.487</p> <p>The Media SDK Dynamic Library for audio processing, software implementation:</p> <p>libmfxaudiosw32.dll for IA-32 architecture</p> <p>libmfxaudiosw64.dll for Intel® 64 architecture</p> <p>version 6.0.0.1</p>
<p><install-folder>\ bin\<arch>\22d62c07e672408fbb4cc20ed7a053e4</p>	<p>Intel® Media SDK for Windows – Screen Capture plug-in:</p> <ul style="list-style-type: none"> • Screen Capture plug-in <p>mfxplugin64_screen_capture.dll for Intel® 64 architecture</p> <p>mfxplugin32_screen_capture.dll for IA-32 architecture</p> • Configuration file <p>plugin.cfg</p> <p>Plugin product version: 1.23.3.164</p>
<p><install-folder>\ bin\<arch>\54542616243341e693ae899942ce7355</p>	<p>Intel® Media SDK for Windows – RAW Accelerator plug-in:</p> <ul style="list-style-type: none"> • Camera plug-in <p>mfxplugin_camera_pipe64.dll for Intel® 64 architecture</p> <p>mfxplugin_camera_pipe32.dll for IA-32 architecture</p> • Configuration file <p>plugin.cfg</p> <p>Plugin product version: 1.23.7.164</p>
<p><install-folder>\doc</p>	<p>Media SDK documentation:</p> <ul style="list-style-type: none"> • Media SDK Reference Manual <p>mediasdk-man.pdf</p> • Media SDK Extensions for User-Defined Functions <p>mediasdkusr-man.pdf</p>

	<ul style="list-style-type: none"> • Media SDK Extensions for Multi-view Video Coding mediasdkmvc-man.pdf • Media SDK Extensions for JPEG*/Motion JPEG mediasdkjpeg-man.pdf • Media SDK Library Distribution and Dispatching Process Description mediasdk-distrib.pdf • Screen Capture Decoder Reference Manual mediasdkscreencap-man.pdf • RAW Accelerator Reference Manual media-raw-accelerator-man.pdf • Reference Manual for Audio Processing mediasdkaudio-man.pdf
<pre><install-folder>\include</pre>	<p>External Media SDK headers:</p> <ul style="list-style-type: none"> • Type definitions in <code>mfxdefs.h</code> • Structure definitions in <code>mfxstructures.h</code>, <code>mfxstructures.h</code>, <code>mfxstructures.h</code>, <code>mfxvstructures.h</code> and <code>mfxcommon.h</code> • SDK session related definitions in <code>mfxsession.h</code> • Video function definitions in C in <code>mfxvideo.h</code> • C++ wrapper of the SDK video functions in <code>mfxvideo++.h</code> • Audio function definitions in C in <code>mfxaudio.h</code> • C++ wrapper of the SDK audio functions in <code>mfxaudio++.h</code> • Extensions for Multi-view Video Coding options <code>mfxmvc.h</code> • Extensions for User-Defined Functions <code>mfxplugin.h</code>

	<ul style="list-style-type: none"> • C++ wrapper for User-Defined Functions <code>mfxplugin++.h</code> • Extensions for JPEG*/Motion JPEG Video coding options <code>mfxjpeg.h</code> • Structure definitions for RAW Media Accelerator <code>mfxcamera.h</code> • Structure definitions for Screen Capture plug-in <code>mfxsc.h</code> • Definitions for vp8 and vp9 codecs <code>mfxvp8.h</code>, <code>mfxvp9.h</code> • Definitions for enc, pak functions and additional features <code>mfxenc.h</code>, <code>mfxpak.h</code>, <code>mfxla.h</code>
<pre><install-folder>\ lib\ <arch></pre>	Static Dispatcher Library <code>libmfx.lib</code> and <code>libmfx_vs2015.lib</code> for Microsoft* Visual Studio 2015
<pre><install-folder>\ igfx_s3dcontrol\ include</pre>	S3D API definitions <code>igfx_s3dcontrol.h</code>
<pre><install-folder>\ igfx_s3dcontrol\ lib\<arch></pre>	Static S3D Control Library <code>igfx_s3dcontrol.lib</code>
<pre><install-folder>\ igfx_s3dcontrol\</pre>	Displaying S3D with Intel® HD Graphics Developers Guide <code>Displaying S3D with Intel HD Graphics.pdf</code>
<pre><install-folder>\ opensource\</pre>	Source code of the Media SDK dispatcher
<pre><install-folder>\ tools\</pre>	Contains the following tools in binary form: <ul style="list-style-type: none"> • Media SDK Tracer in folder <code>mediasdk_tracer</code>. This utility performs runtime recording of the Media SDK API calls and

	<p>parameters to a log file.</p> <ul style="list-style-type: none"> Media SDK System Analyzer in folder <code>mediasdk_sys_analyzer</code>. This utility analyzes the system and reports back the Media SDK related capabilities, graphics driver and components status.
--	---

Installation

- Installation requires full administrative rights.
- Run the installer `Intel_Media_SDK_2017_R1.msi`.

Known Limitations

Media SDK libraries and driver have the following known limitations. Unless explicitly specified each limitation is relevant for both software and hardware implementations of SDK dynamic library.

API

- The following APIs are not supported by the software implementation of the SDK library:
 - `mfxExtEncoderCapability`, `mfxExtEncoderResetOption`, `mfxExtAVCEncodedFrameInfo`;
 - `MFX_RATECONTROL_LA`, `MFX_RATECONTROL_ICQ`, `MFX_RATECONTROL_LA_ICQ`, `MFX_RATECONTROL_VCM`, `MFX_RATECONTROL_QVBR`, `MFX_RATECONTROL_LA_HRD` and any options related to these BRC modes;
 - `mfxExtVPPComposite`, `mfxExtVPPDeinterlacing`;
 - `mfxExtAVCRefListCtrl::ApplyLongTermIdx`, `LongTermIdx`;
 - `mfxExtEncoderROI`;
 - `mfxExtCodingOption2::Trellis`, `MBBRC`, `ExtBRC`, `RepeatPPS`, `BRefType`, `AdaptiveI`, `AdaptiveB`, `LookAheadDS`;
 - `mfxInfoMFX::ICQQuality`;
 - `mfxEncodeCtrl::SkipFrame`;
 - `MFX_PLUGINID_HEVCE_HW`, `MFX_EXTBUFF_HEVC_PARAM`, `mfxExtHEVCParam`;
 - `CreateAccelerationDevice`;
 - `MFXInitEx`, `MFXDoWork`, `mfxInitParam`, `mfxExtThreadsParam`;
 - `mfxInfoMFX::LowPower`;

- MFX_EXTBUFF_DECODED_FRAME, mfxExtDecodedFrameInfo, MFX_EXTBUFF_TIME_CODE, mfxExtTimeCode;
 - mfxExtCodingOption3;
 - mfxExtPredWeightTable;
 - mfxExtDirtyRect;
 - mfxExtMovingRect;
 - mfxInitParam::GPUCopy;
 - mfxInfoMFX::MaxDecFrameBuffering;
 - mfxVideoParam:: AllocId;
 - MFX_MEMTYPE_EXPORT_FRAME;
 - mfxExtCodingOptionVPS;
 - mfxExtVPPRotation;
 - mfxExtVPPSignalInfo;
 - mfxExtVPPMirroring;
 - MFXVideoCORE_QueryPlatform;
 - mfxPayload::CtrlFlags;
 - mfxFrameData::MemType;
 - mfxExtVPPScaling;
 - mfxExtVPPColorFill;
 - mfxExtEncodedSlicesInfo;
 - MFX_MEMTYPE_SHARED_RESOURCE;
 - mfxCoreInterface::QueryPlatform;
 - mfxExtSceneChange;
 - mfxExtVPPColorFill;
 - mfxExtHEVCRegion;
 - mfxExtVPPFieldProcessing;
 - MFX_DEINTERLACING_ADVANCED_SCD;
 - MFX_DEINTERLACING_FIELD_WEAVING;
 - MFX_REFRESH_NO, MFX_REFRESH_VERTICAL, MFX_REFRESH_HORIZONTAL, MFX_REFRESH_SLICE;
 - MFXCoreInterface::QueryPlatform;
 - mfxExtDecVideoProcessing;
 - mfxExtMBForceIntra;
- The below APIs are not supported by the hardware implementation of SDK Library in this release. For other APIs not mentioned in this list make sure to call `Query` functions to check the actual support on a particular platform as it may vary.
 - mfxExtCodingOption2::AdaptiveI, AdaptiveB, UseRawRef;

- mfxExtAVCEncodedFrameInfo::MAD, BRCPanicMode, QP;
- MFX_PLUGINID_VP8D_HW;
- mfxExtChromaLocInfo;
- mfxExtMBQP;
- MFXInitEx, MFXDoWork, mfxInitParam, mfxExtThreadsParam;
- mfxExtDirtyRect;
- mfxExtMoveRect;
- mfxExtCodingOption3::ScenarioInfo, ContentInfo, RepartitionCheckEnable EnableMBQP, DirectBiasAdjustment, GlobalMotionBiasAdjustment;
- mfxExtCodingOptionVPS;
- mfxExtEncodedSlicesInfo;
- mfxExtSceneChange;
- mfxExtHEVCRegion;
- mfxExtVPPFieldProcessing;
- MFX_PICSTRUCT_FIELD_SINGLE, MFX_PICSTRUCT_FIELD_TOP, MFX_PICSTRUCT_FIELD_BOTTOM, MFX_PICSTRUCT_FIELD_PAIRED_PREV, MFX_PICSTRUCT_FIELD_PAIRED_NEXT;
- mfxExtEncoderROI::ROIMode.

Functional

Common/general limitations:

- The 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 SDK dynamic libraries `libmfxsw64.dll` and `libmfxhw64.dll` not through the dispatcher is unsafe.
- Using the software implementation of SDK in parallel with Intel® Threading Building Blocks could impact performance.
- The number of internal tasks in hardware implementation is limited to 1024. This imposes a related limitation on the number of SDK sessions which depends on the number of components in a session and the asynchronous depth of each component: each component (DECODE, ENCODE or VPP) requires one task for synchronous operation and N tasks for asynchronous operation with depth N.
- This release supports only 64-bit Microsoft* Windows* applications.
- Microsoft DirectX* 11.1 is the only supported acceleration infrastructure (due to headless mode requirement).
- On Windows 10 bitstreams produced by any encoder can be not bit exact from time to time, but no visible difference.

- Recommended to use system memory allocated with pointer alignment to 64, otherwise result can contain corruptions and sometimes pipeline can be broken.

JPEG encode, decode:

- The feature set of JPEG decoder/encoder is limited to the following:
 - 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.
 - No extended, lossless and hierarchical modes
 - no 12-bit samples;
 - no progressive;
 - no arithmetic coding;
 - no 4 AC and 4 DC Huffman tables.
- JPEG decoder does not set `Corrupted` flag of `mfxFrameData` structure, and does not accept `MXF_BITSTREAM_EOS` as `DataFlag` of `mfxBitstream` structure.
- Software library implementation can break pipeline in stress multisession encoding and complex threading graphs scenarios when two or more encoders sharing the same `mfxFrameSurface1` object and memory handling implemented through `MemId`.
- Encoder with GPU copy may produce stream witch not bit to bit with stream encoded without GPU copy.

HEVC decode:

- HEVC HW plugin MAIN10 profile limited to Luma and Chroma BitDepth 10, all other BitDepth are unsupported in current implementation.
- HW HEVC decode plugin is limited to 4096x2304 resolution and doesn't implement SW fallback for higher resolutions 5th generation Intel® Core and Intel® Xeon E3 v4. You may use software implementation of HEVC decode from Intel® Media Server Studio 2015 – Professional Edition to support higher resolutions. On 6th Generation Intel Core and Xeon E3 v5 it is limited to 8192x8192 resolution.
- Interlaced decoding is supported only through separate field output(half frame size), and no specific reporting of such behavior implemented in current version, if App need interlace decode support, it can detect interlaced sequence through: get SEI through `GetPayload`, parse SEI message to get `pic_struct` value for SEI semantics(See D.2.3 section of High efficiency video coding).

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

Page 11 of 20

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

Copyright © 2007-2017, Intel Corporation

H.264 decode:

- 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.

MPEG-2 encode limitations:

- 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`.
- MPEG-2 encode may produce not bit to bit result from run to run on 5th generation Intel® Core and Intel® Xeon E3 v4. Run to run variation doesn't affect visual quality.
- `mfxExtCodingOption3::BRCPanicMode` not supported for windows.

HEVC encode limitations:

- Supports maximum resolution 4096x2172.
- Supports only CQP, CBR, VBR, ICQ and AVBR rate control methods, as well as `MFx_RATECONTROL_LA_EXT` with lookahead plugin.
- Query max macroblock per second report not supported.
- Next limitation for encoded bitstream:
 - No tiles support;
 - No SAO;
 - No P slice;
 - Progressive only.
- QP Offset for HEVC is 1 by default.
- HW Accelerated HEVC encode supports next extended buffer ids, with listed limitations:
 - `MFx_EXTBUFF_CODING_OPTION`;
 - `MFx_EXTBUFF_OPAQUE_SURFACE_ALLOCATION`;
 - `MFx_EXTBUFF_HEVC_REFLISTS`;
 - `MFx_EXTBUFF_HEVC_REFLIST_CTRL`;
 - `MFx_EXTBUFF_CODING_OPTION_VPS`;
 - `MFx_EXTBUFF_ENCODED_FRAME_INFO`;
 - `MFx_EXTBUFF_VIDEO_SIGNAL_INFO`;
 - `MFx_EXTBUFF_CODING_OPTION2`;
 - Supported:
 - `MBBRC`;
 - `BRefType`;
 - `NumMbPerSlice`;
 - `DisableDeblockingIdc`;
 - `RepeatPPS`;
 - `NumMbPerSlice`;
 - `MaxFrameSize`.
 - `MFx_EXTBUFF_CODING_OPTION3`;
 - Supported:
 - `PRefType`;
 - `EnableQPOffset`;
 - `QPOffset`.

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

Page 12 of 20

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

Copyright © 2007-2017, Intel Corporation

- MFX_EXTBUFF_CODING_OPTION_SPSPPS;
- MFX_EXTBUFF_HEVC_TEMPORAL_LAYERS;
- MFX_EXTBUFF_ENCODER_RESET_OPTION;
- MFX_EXTBUFF_HEVC_PARAMS;
 - GeneralConstraintFlags: not supported.
- Encoder with CBR rate control and height bitrate may produce stream with HRD overflows.
- MaxFrameSize violation may happen in case of low bit rate and specific content. To avoid MaxFrameSize violation recommend to set $\text{MaxFrameSize} > 5 * \text{Bitrate} / \text{FPS} / 8$.
- Encoder with CBR rate control, big resolution and small max bitrate may fail.
- Supports maximum payloads 160 byte per frame.

H.264 AVC/MVC encode limitations:

- Specific QSV-FF encode limitations:
 - Only I and P frame types supported, so no API related to B frames encoding supported.
 - CQP, CBR, VBR and QVBR rate control methods only.
 - Supported QP values range: 10 – 51.
 - Only progressive encoding supported.
 - Maximum number of reference frames equal to 3.
 - Encoding quality can be worse than legacy encode with similar parameters.
 - Encoding bitrates higher than 43 Mbps not supported in current implementation.
 - MaxSliceSize feature can produce slices non-compliant to specified value, it is content dependent how often slice size overflow can occur, so need to be tested according to usage and requirement if such violations acceptable or not.
 - QSV-FF encode not utilizing render engine, except next 2 cases:
 - Target Usage equal to 1 and NumRefFrame equal to 3.
 - ARGB used as input.
 - Not supported `mfxExtCodingOption3::AdaptiveMaxFrameSize`.
- MaxQP with MaxFrameSize – not working, MaxQP will have more priority and MaxFrameSize will be violated.
- 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 in software implementation 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.
- RefPicMarkRepSEI syntax is not supported by MVC encoder.
- Known limitations for H.264 Multiple-Segment Encoding:

- Hardcoded HRD parameters: `bit_rate_scale = 0`, `cpb_size_scale = 3`.
 - 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).
 - 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).
 - Encoded `time_scale`, `num_units_in_tick` could be both multiplied by 2 if the `time_scale` from original SPS is odd.
 - Conflicts between SPS/PPS and `mfxVideoParam` for parameters that are not covered by SPS/PPS could lead to change of parameters in SPS/PPS.
- Target usage 7 of H.264/MVC encoders in software implementation is known to have a non-monotonic quality versus bitrate dependency.
 - MVC encoder ignores any user SEI messages for the dependent view.
 - The look ahead bitrate control mode may produce non HRD compliant encoded streams.
 - `mfxExtCodingOption2::LookAheadDS` currently supports only `MFX_LOOKAHEAD_DS_OFF` and `MFX_LOOKAHEAD_DS_2x`, `MFX_LOOKAHEAD_DS_4x` will give the same result as `MFX_LOOKAHEAD_DS_2x`. `MFX_LOOKAHEAD_DS_OFF` is the default value for target usage 1 and 2. `MFX_LOOKAHEAD_DS_2x` is the default value for target usages 3 – 7.
 - H.264 and MVC encoders may not obey the minimum compression ratio required by the Blu-Ray*/AVCHD* specifications when the requirement is stronger than in H.264 standard.
 - The value reported via `mfxExtEncoderCapability::MBPerSec` may be bigger than the actual maximum processing rate of the encoder.
 - To change encoding parameters on the fly with `Reset()` function w/o IDR insertion application should drain all the buffered surfaces from encoder. Otherwise encoder may demonstrate undefined behavior after `Reset`.
 - Call of `Reset` which starts new sequence (inserts IDR) will drop HRD conformance over the inserted IDR (CPB removal counter will be set to 0 in the IDR Picture Timing SEI).
 - HRD violations are possible in specific scenarios (e.g. massive frame skipping).
 - Encoder prohibits increase of DPB size (`NumRefFrame`) via `Reset` function even if new size is lower than initialization value. `Reset` function will return `MFX_ERR_INCOMPATIBLE_VIDEO_PARAM` on any attempt to increase `NumRefFrame`.
 - Target usage `MFX_TARGETUSAGE_BEST_SPEED` may produce better objective quality than `MFX_TARGETUSAGE_BALANCED`.
 - Encoder may not insert PCM macroblocks when required. Encoding of specific (complex) content with huge bitrate (which makes encoder insert many PCM MBs) may cause a GPU hang on Haswell.

- Reset function isn't supported for LookAhead BRC modes. Reset function doesn't return an error when called together with LA BRC. Result of such Reset call is undefined.
- B-pyramid isn't supported together with HRD compliant Look Ahead BRC (MFX_RATECONTROL_LA_HRD) and Look Ahead BRC with sliding window control (MFX_RATECONTROL_LA and WinBRCTMaxAvgKbps and WinBRCSIZE).
- Careful memory/resource planning is needed when using Look Ahead BRC due to storage of pre-analyzed frames. 1:N and N:N transcoding use cases are especially demanding for memory.
- When external Look Ahead BRC is used in transcoding pipeline which includes Frame Rate conversion (FRC), FRC in the pipeline should take place before external Look Ahead. Otherwise encoder may return error MFX_ERR_UNDEFINED_BEHAVIOR from EncodeFrameAsync calls.
- Trellis option can be enabled only on lower target usages, on some of those it is enabled by default but can be switched off. Exact implementation details are hidden and may change with time and between platforms. Use of Query function to retrieve actual support is strongly recommended.
- SkipFrame feature has the following limitations:
 - If GOP has only P frames, arbitrary P can be skipped. When skipped, it is made non-reference.
 - If GOP has B frames, only non-reference B can be skipped.
- Some fails can happen during encoding in case of reinitialization of encoder in one session with DirectX11.
- On specific content and height bitrate encoder doesn't always re-code MB as PCM when its size exceed allowed by standard.
- Encoding using VBR and dynamically adjusted the MaxFrameSize on a per-frame basis algorithm may leads to HRD buffer underflow and quality artifacts.
- Max bit rate violations are possible in case of using VBR and specific payload.
- For RGB input in case of unspecified mfxExtVideoSignalInfo::MatrixCoefficients or not equals 1 will be used BT.601, if mfxExtVideoSignalInfo::MatrixCoefficients = 1 - BT.709. Actual list of supported conversions may depend from version of driver. On 7th Generation Intel® Core™ and driver 15.46 mfxExtVideoSignalInfo::MatrixCoefficients will be ignored.
- For HW BRC: supported only WinBRCSIZE <=1 sec.

MPEG-2 decode limitations:

- MPEG-2 Video decoder returns MFX_ERR_UNDEFINED_BEHAVIOR instead of MFX_ERR_MORE_DATA when part of sequence header is absent and MFX_BITSTREAM_COMPLETE_FRAME flag is set.
- Decoder does not support bitstreams with resolution bigger than 2096x2096. MFXVideoDECODE_Init returns MFX_ERR_UNSUPPORTED on such bitstreams.
- Decoder does not support MPEG-1 bitstreams. It is interpreted as corrupted MPEG-2 bitstream. MFXVideoDECODE_Init returns MFX_ERR_NONE and

MFXVideoDECODE_DecodeFrameAsync returns MFX_ERR_MORE_DATA until valid MPEG-2 bitstream is found.

- Sequence headers are skipped if resolution exceeds maximum supported values (2096x2096) or level/chroma are invalid. It affects MFXVideoDECODE_DecodeFrameAsync.
- mfxDecodeStat.NumFrame that is returned from GetDecodeStat function is less by 1 than actual count of decoded frames if GetDecodeStat is called after buffered frames are returned.

VC1 decode limitations:

- VC1 decoder can't set corruption flag in P,B-Frames if previous I-Frame was missed.

VPP component has next limitations:

- Composition has issues after reset with input system memory usage, input system memory not recommended for use with composition if resetting composition pipeline required.
- MFX_FRCALGM_DISTRIBUTED_TIMESTAMP is unsupported by InverseTelecine and Deinterlace (60i->60p) VPP filters.
- mfxExtVppAuxData is deprecated.
- VPP::Query in software implementation mistakenly indicates support for MFX_FRCALGM_FRAME_INTERPOLATION while it is actually not available.
- VPP scaling in software implementation may produce slightly blurred frames for RGB32 interlaced content.
- VPP Image Stabilization and FRC Interpolated filters will be deprecated and not recommended for use due to result image quality and product quality associated with these filters.
- When De-interlace and FRC filters used in pipeline MFX_DEINTERLACING_BOB will be used by default, if not specified explicitly to use MFX_DEINTERLACING_ADVANCED.
- MFX_FOURCC_YV12 supported only via software fallback with DirectX* 11.1 interface.
- VPP in software implementation always uses simple FRC algorithm based on repeat/drop frames and ignores MFX_FRCALGM_FRAME_INTERPOLATION flag.
- ADI may produce color artefacts on frame preceding a scene change.
- ADI may produce color artefacts in case of harmonic motion (repeated pattern and motion magnitude is the same as the periodic of repeated pattern).
- Multiple VPP filters being combined in one session may produce output that is not bit-exact with the output from the same VPP filters that are split by separate sessions, but the difference does not affect visual quality.
- Frames with interlaced content must have CropH multiple of 4. Otherwise, VPP may produce color artefacts on the bottom lines.
- De-interlacing is supported for NV12, YUY2 formats only.

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

- Once enabled at the Init stage VPP de-interlacing for BOB mode is not disabled automatically if application provides input frames with picstruct set to MFX_PICSTRUCT_PROGRESSIVE.
- 1st frame is doubled during 30i->60p de-interlacing for MFX_DEINTERLACING_BOB and MFX_DEINTERLACING_ADVANCED_NOREF modes.
- MFXVideoVPP_GetVideoParam does not update values in attached extended buffers, except VPP_DO_USE.
- MFXVideoVPP_Reset could return error if additional memory allocation is required based on provided video parameters (changed type of color or frame rate conversion for example). The application should close VPP component and then re-initialize it in this case. See MFXVideoVPP_Reset in SDK API Reference Manual for more details about possible return statuses.
- Multiple RGB4 surfaces with PixelAlphaEnable enabled could give poor visual quality after composition on the same area with overlapping.
- Deprecated MFX_EXTBUFF_VPP_PICTSTRUCT_DETECTION enumerator.
- VPP Mirroring in pipeline with video to video memory in/output patterns can be supported only without any other filters, in/out memory patterns containing system memory is supported with any other legal filters combination.
- ADI with doubled framerate – there is an issue when first 3 frames copied instead of 2.
- On 6th generation Intel Core and Xeon E3 v5 Temporal Denoise enabled – this can affect some result of denoise on scene change and after reset with changed denoise factor.
- mfxExtVPPSignalInfo has to be attached to surface with NV12 format only when color conversion done.
- Rotate + Detail/ProcAmp – in combination affect each other, thus result can differ comparing running filters separately and in one call.

The Screen Capture have the following known limitations:

- Screen Capture plugin will use fallback to standard Microsoft* DirectX* Video Acceleration (DXVA) path in case when monitor is connected and application uses exclusive render mode or OPM mode.
- Screen capture with scaling (i.e. when initialization resolution is not equal current display resolution) is supported only with NV12 fourcc format.
- Screen Capture plugin loading recommended only with SDK HW library implementation API 1.15 or later.
- In case of DirectX 11 implementation, video memory type, and RGB4 surface format usage, the application frame allocator needs to allocate the surfaces using DXGI_FORMAT_AYUV format because OS runtime will block RGB surface allocation with BIND_DECODER flag and decoder output view. In any other configuration cases, e.g. DirectX 9 implementation, system or opaque memory type, or NV12 output format, special frame allocation is not needed.
- DXVA fallback functionality doesn't work for resolution change.

- DirtyRect detection feature will detect dirty areas in captured frame and coalesce areas to bigger ones until get 256 areas to feed into DirtyRect API buffer.
- Display Selection API supported only for Virtual Displays.
- Windows 10 acceleration currently unsupported, plugin will fall back to standard Microsoft functions, and performance drop can be observed.

The Media RAW Accelerator Plug-in has the following known limitations:

- Current version supports sessions only initialized as MFX_IMPL_HARDWARE, MFX_IMPL_HARDWARE_ANY, MFX_IMPL_VIA_D3D11, MFX_IMPL_VIA_D3D9.
- Software fallback implementation is not optimized thus performance is very poor.
- Software fallback can be not bit exact with GPU optimized version.
- Current version supports only system memory as input memory type.
- Current version supports only system memory as output memory type for MFX_FOURCC_ARGB16 and MFX_FOURCC_ABGR16 color types.
- Current version of plugin doesn't support opaque memory type.
- Additional Intel Media SDK session must be created to use traditional VPP features of Intel Media SDK HW Library.
- MFX_GAMMA_MODE_VALUE currently not supported.
- Current version supports only system memory output type for MFX_FOURCC_ARGB4 color type not supported with output Video memory type in case when total surface size is bigger than 6000x4000 on processors older than 6th generation Intel Core.
- For resolutions more than 6000x4000 result will be slightly different for padded content and non-padded content due to internal implementation.
- Application need to do MFXInit with API 1.10 version to have RAW Accelerator working on all available driver versions.
- MFX_EXTBUF_CAM_FORWARD_GAMMA_CORRECTION API is slower than MFX_EXTBUF_CAM_GAMMA_CORRECTION on platforms older than 6th generation Intel Core processors.
- Vignette correction filter support is limited to photo with 8192x8192 resolution of picture on 6th Generation Intel® Core™ platforms.
- For resolutions width bigger than 8192 and MFX_FOURCC_ARGB16 or MFX_FOURCC_ABGR16 color types plug-in will fall back to GPU acceleration instead of fixed function on 6th generation Core platforms.
- 3D look up table filter supported only on 6th generation Core platforms and will utilize render engine for acceleration.

Non zero values of CropY may cause RunFrameVPPAsync to fail with MFX_ERR_DEVICE_FAILED (-17 error code) only on 6th Generation Intel Core™ Processors based platforms.

The Audio library has the following known limitations and features:

- Loading of Audio library `libmfxaudiosw32.dll/libmfxaudiosw64.dll` not through the SDK Dispatcher is unsafe.
- Bitrate calculation routine in `MFXAudioENCODE_Init` may work incorrectly for some AAC High Efficiency (HE) profile streams. As a result `MFXAudioENCODE_Init` function returns an error.
- Bitrate values for AAC HE profile vary from 96 Kbit/s to 265 Kbit/s.
- The following profile and mode values are not supported:

- MFX_PROFILE_AAC_SSR
- MFX_PROFILE_AAC_ALS
- MFX_PROFILE_AAC_BSAC
- MFX_AUDIO_MP3_LFE_FILTER_ON always disabled
- AAC decoder may handle streams with multiple (more than 2) channels incorrectly.
- Audio has API version 1.15.

Decoding

- AAC, supported profiles:
 - Low Complexity (LC);
 - Long Term Prediction (LTP)
 - Parametric Stereo (PS)
 - Spectral Band Replication (SBR)/HE AAC v1
 - HE AAC v2 (SBR + PS)
- Mpeg audio, supported standards, profiles
 - MPEG-1 Audio Layer I, Layer II and Layer III (ISO*/IEC* 11172-3)
 - MPEG-2 Audio Layer I, Layer II and Layer III (ISO/IEC 13818-3)

Encoding

- AAC supported profiles:
 - Low Complexity (LC)
 - Spectral Band Replication (SBR) /HE AAC v1
 - Output formats for AAC – ADTS, RAW, ADIF
 - Stereo mode – LR, JOINT, MS

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,

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

Page 19 of 20

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

Copyright © 2007-2017, Intel Corporation

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