

Intel® Media Server Studio 2017 – Driver, SDK for Linux* Release Notes

[Overview](#)

[What's New](#)

[Features](#)

[System Requirements](#)

[Package Contents](#)

[Installation Folders](#)

[Documentation](#)

[Known Limitations](#)

[Legal Information](#)

[Attributions](#)

Overview

For the most up to date version please refer the Intel® Media Server Studio Support [documentation page](#).

The Intel® Media Server Studio – Driver, SDK for Linux* provide software development tools and libraries needed to develop enterprise grade media solutions on Intel® Server Products. The studio is designed for optimizing datacenter and embedded media applications for Linux server operating systems to utilize Intel® Iris™ and Intel® HD Graphics hardware acceleration capabilities.

The package includes the following components:

- Intel® Media Server Studio 2017 – Graphics Driver, version 16.5.55964
- Intel® Media Server Studio 2017 – SDK, version 7.0.16053533
- Intel® Media Server Studio 2017 – Samples Binaries, version 7.0.16053497. The latest version of samples package (with all samples binaries and corresponding source code) could be downloaded from [Intel® Media Server Studio Support](#).

This document covers product features, system requirements and known limitations. For installation procedures description please see the `<sdk-extract-folder>/media_server_studio_sdk_getting_started_guide.pdf`.

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

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

Copyright © 2015-2016, Intel Corporation

In this document we will refer to processor families by their former codenames for the sake of readability:

- Intel® Xeon® E3-1200/1500 v5 Family and 6th Generation Intel Core™ Processors will be referred to as "Skylake"
- Intel® Xeon® E3-1200 v4 Family and 5th Generation Intel Core™ Processors will be referred to as "Broadwell"

What's New

Intel® Media Server Studio 2017 – SDK (hereinafter referred to as "SDK"):

Version 7.0.16053533:

- OS and hardware support:
 - CentOS* 7.2. The release supports the latest kernel 3.10.327.13.1.x86_64 as well as original kernel of CentOS 7.2
 - Set of patches for open source components on top of 4.4.0 kernel suitable for other variants of Linux* OSES

New features:

- **SDK API 1.19**

Please note that not all APIs from 1.19 version are supported, consult Known Limitations->API section for details

- **Skylake platforms support**

Skylake architecture introduces a number of power efficiency features over its predecessors in addition to more powerful graphics. Please view the Skylake Performance whitepaper (in the [Intel Media Server Studio Documentation webpage](#)) to understand how to extract the best performance from this platform on media transcode use-cases. The whitepaper will present the key efficiency features introduced in Skylake, best known methods on how to leverage them to extract performance for media transcode workloads spanning multiple resolutions, and use-cases.

- **GPU hang reporting and recovery**

Starting from version 2017 Intel® Media Server Studio supports reporting of GPU hangs occurred during SDK operations. This new feature allows application to establish proper GPU hang recovery procedure without the need for additional monitoring of the system (e.g. w/o checking dmesg logs). The feature is supported only on Linux* OS.

GPU hang reporting procedure: if GPU hang occurred during HW operation, SDK returns `status_MFX_ERR_GPU_HANG` from any `SyncOperation()` call which synchronizes SDK workload affected by hang. In addition SDK rejects to accept new frames and returns `MFX_ERR_GPU_HANG` status from any subsequent call of `EncodeFrameAsync()`, `DecodeFrameAsync()`, `RunFrameVPPAsync()`. It's available for H.264, H.265, MPEG2-Video decoders and encoders.

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

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

GPU hang recovery procedure: if application got `MFX_ERR_GPU_HANG` status from any SDK component, it should:

1. [Optional] call `MFXVideoCORE_SyncOperation()` for all submitted and non-completed tasks. It is possible that independent or preceding frames have been proceeded successfully.
2. Re-initialize all SDK components and sessions on the system (close all components, close all sessions, initialize sessions and components back).

In general case GPU hang recovery procedure doesn't require re-creation of acceleration device (vaDisplay for Linux) and re-allocation of input/output surfaces in video memory (see exceptions in "Related limitations").

Informative: usually it takes SDK a few seconds to detect and report GPU hang. During this time all `SyncOperation()` calls for tasks affected by GPU hang will return status `MFX_WRN_IN_EXECUTION`. SDK will report the hang with status `MFX_ERR_GPU_HANG` only after GPU hang is detected and HW recovery mechanism is started by driver.

Related limitations:

- GPU hang reporting is supported for Linux only.
- Processing (VPP) SDK component doesn't support GPU hang reporting. In most of cases `MFX_ERR_DEVICE_FAILED` will be returned from VPP `SyncOperation()` after GPU hang. Also GPU hang occurred during VPP operation may be caught by Encoder (as Encoder shares HW processing units with VPP). If Encoder returned `MFX_ERR_GPU_HANG` in pipeline with VPP, it's recommended to follow full HW error recovery procedure described in "Hardware Device Error Handling" section of SDK documentation.
- SDK decoders may buffer decoded frames. Decoder `SyncOperation()` call may return previously buffered frame with `MFX_ERR_NONE` status at the time when GPU hang was already reported by underlying HW for one of next frames. This means that even for synchronous pipeline application may get `MFX_ERR_GPU_HANG` from `DecodeFrameAsync()` earlier than from `SyncOperation()`.
- HEVC Decoder doesn't properly report hardware errors and GPU hangs from `SyncOperation`. In case of such error `SyncOperation` returns `MFX_ERR_ABORTED`. After this, the proper error (`MFX_ERR_DEVICE_FAILED` or `MFX_ERR_GPU_HANG`) is returned from subsequent calls of `DecodeFrameAsync`.
- Application should treat `MFX_ERR_ABORTED` status returned from `MFXVideoCORE_SyncOperation()` as `MFX_ERR_DEVICE_FAILED` and run recovery procedure as described in Hardware Device Error Handling section of the manual
- **Considering above limitations we recommend to simplify application logic and process `MFX_ERR_DEVICE_FAILED`, `MFX_ERR_GPU_HANG` and `MFX_ERR_ABORTED` uniformly using the full reset procedure described in "Hardware Device Error Handling" of SDK manual. (I.e. recreate all resources: acceleration device, frames memory, SDK sessions, SDK components).**

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

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

Page 3 of 27

Copyright © 2015-2016, Intel Corporation

- **Support of unlimited number of surfaces in decoder work/output pool** (previously max 128) for H.264, H.265 and MPEG2-Video decoders.
- **H.265 hardware accelerated decoder** was enabled on Skylake. The maximum supported resolution for H.265 decoder is 8192x8192
- **H.265 hardware accelerated encoder** was enabled on Skylake. The maximum supported resolution is 4096x2176
- **VPP**
 - New Advanced Deinterlacing mode with a scene change detector (MFX_DEINTERLACING_ADVANCED_SCD) was introduced. When the scene change detector identifies a large change between adjacent frames or fields it will trigger a single-field deinterlace process (MFX_DEINTERLACING_BOB) instead of the typical multi-field advanced deinterlace algorithm. Designed to work with output frame rates half of input field rates, i.e. 60i->30p, or 50i->25p.
 - MFXVideoVPP_GetVideoParam now reports enabled filters parameters. Please check Reference Manual document for usage details.
 - Latency for multiple MFXVideoVPP_Query/MFXVideoVPP_QueryIOSurf calls was reduced.
 - VPP allows to choose scaling algorithm. By default scaling quality mode will be used
- **MJPEG encoder:** hardware acceleration was enabled on Skylake
- **MPEG2 encoder:** MBQP feature was fully enabled
- **VP8 decoder:** support for opaque memory type was added
- **USER:** 2 new functions for user-defined interface were added (please see <sdk-install-folder>/doc/mediasdkusr-man.pdf for details):
 - mfxStatus (*QueryPlatform) (mfxHDL pthis, mfxPlatform *platform)
 - mfxStatus MFXVideoUSER_GetPlugin(mfxSession session, mfxU32 type, mfxPlugin *par)
- **Metrics Monitor:** support for Skylake was added

Features

Intel® Media Server Studio 2017 – SDK included in this package implements SDK API 1.19 and contains the following components:

Component	Supported features	Limitations
H.265 decoder	Supported Profiles: <ul style="list-style-type: none"> • Main 	Maximum supported resolution: 8192x8192

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

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

Page 4 of 27

Copyright © 2015-2016, Intel Corporation

H.265 encoder	Supported Profiles: <ul style="list-style-type: none"> • Main Supported BRC methods: <ul style="list-style-type: none"> • Constant QP (CQP) 	Maximum supported resolution: 4096x2176
H.264 decoder	Supported Profiles: <ul style="list-style-type: none"> • Baseline • Main • High 	Maximum supported resolution: 4096x2304
H.264 encoder	Supported Profiles: <ul style="list-style-type: none"> • Baseline • Main • High Supported BRC methods: <ul style="list-style-type: none"> • Constant QP (CQP) • Constant Bit Rate (CBR) • Variable Bit Rate (VBR) • Look Ahead (LA) 	Maximum supported resolution: 4096x4096
MPEG-2 decoder	Supported Profiles: <ul style="list-style-type: none"> • Simple • Main • High 	Maximum supported resolution: 2048x2048
MPEG-2 encoder	Supported Profiles: <ul style="list-style-type: none"> • Simple • Main Supported BRC methods: <ul style="list-style-type: none"> • Constant QP (CQP) • Constant Bit Rate (CBR) • Variable Bit Rate (VBR) 	Maximum supported resolution: 1920x1088
VC1 decoder	Supported Profiles: <ul style="list-style-type: none"> • Simple • Main • Advanced 	Maximum supported resolution: 1920x1088
MJPEG encoder	Supported Profiles: <ul style="list-style-type: none"> • Baseline mode, 8bit 	Maximum supported resolution: per ISO/IEC 14495-1 and system

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

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

Page 5 of 27

Copyright © 2015-2016, Intel Corporation

		memory limitations
MJPEG decoder	Supported Profiles: <ul style="list-style-type: none"> • Baseline mode, 8bit 	Maximum supported resolution: per ISO/IEC 14495-1 and system memory limitations
VP8 decoder	Supported Profiles: <ul style="list-style-type: none"> • Version 0 mode, 8bit 	Maximum supported resolution: 1920x1088
Video Pre Processing (VPP)	Supported Algorithms: <ul style="list-style-type: none"> • Color Conversion • Scaling • De-Interlacing (Advanced motion-compensated, BOB) • De-noising • Frame Rate Conversion • Composition • Alpha Blending • Sharpness • PROCAMP • Rotation 	Maximum supported resolution: 4096x4096

Common for all components: minimum supported resolution is 32x32, frame width/height must be a multiple of 32.

NOTE: Please use `Query` functions to check feature availability on any given machine at runtime. Availability of features depends on hardware capabilities as well as driver version.

Please see the Intel® Media Server Studio 2017 - SDK Reference Manual for details "`<sdk-install-folder>/doc/mediasdk-man.pdf`"

System Requirements

Hardware

Intel® Media Server Studio – SDK supports the following platforms with the integrated graphics:

- Intel® Xeon® E3-1200 v4 Family with C226 chipset
- Intel® Xeon® E3-1200 and E3-1500 v5 Family with C236 chipset
- 5th Generation Intel® Core™

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

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

Page 6 of 27

Copyright © 2015-2016, Intel Corporation

- 6th Generation Intel® Core™
- Note: chipset must have processor graphics enabled; make sure to check the datasheet.
 - Having a C226/C236 chipset is necessary but not sufficient. Make sure to consult with specific platform or board vendor regarding processor graphics being supported. Check Media Server Studio website for the list of “Known OEM/ODM Functional Platforms”:
<https://software.intel.com/en-us/intel-media-server-studio/details>

Additionally, for Intel® Xeon® E5 v4 and v5 processors, support of software-only (CPU) HEVC decode and encode, select video pre-processing (Color Space Conversion, Scaling), and virtualization (KVM*, Xen*) is available.

Software

- CentOS 7.2 (1511) of 64-bit architecture its default or latest kernels. **Installation process applies many changes to the kernel, graphics driver, libdrm and libva graphics stack. These changes would need to be reverted to request OS vendor support.**
- Generic OS install uses kernel 4.4.0 from www.kernel.org.

Package Contents

Intel® Media Server Studio 2017 – Driver, SDK for Linux* package includes the following components, where <id> is Graphics Driver version:

Component	Description
CentOS/intel-ocl-<id>.el7.x86_64.rpm CentOS/intel-ocl-devel-<id>.el7.x86_64.rpm	Intel® Media Server Studio - OpenCL™ Driver packages
CentOS/intel-linux-media-<id>.el7.x86_64.rpm	Intel® Media Server Studio – Driver & SDK runtime package.
CentOS/intel-linux-media-devel-<id>.el7.x86_64.rpm	Intel® Media Server Studio – Driver & SDK development package.
CentOS/kmod-ukmd-<id>.el7.x86_64.rpm CentOS/ukmd-kmod-<id>.el7.src.rpm	ukmd kernel module(s)
CentOS/intel-i915-firmware-<id>.el7.x86_64.rpm	Firmware binaries for Skylake support
CentOS/libdrm*-<id>.el7.x86_64.rpm CentOS/drm-utils*-<id>.el7.x86_64.rpm	Direct Rendering Manager runtime library runtime, development, etc.

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

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

Page 7 of 27

Copyright © 2015-2016, Intel Corporation

CentOS/libva*-<id>.el7.x86_64.rpm CentOS/libva-utils*-<id>.el7.x86_64.rpm	Video Acceleration (VA) API runtime, development, etc.
CentOS/install_scripts_centos_<id>.tar.gz	Intel® Media Server Studio – Driver & SDK installation scripts.
CentOS/MediaSamples_Linux_bin-<id>.tar.gz	Intel® Media Server Studio – Samples package (binaries only).
Generic/intel-linux-media_generic_<id>_64bit.tar.gz Generic/intel-opencl-<id>.tar.xz Generic/intel-opencl-devel-<id>.tar.xz	Intel® Media Server Studio – Driver & SDK & OpenCL package for Generic OS.
intel-linux-media-patches-<ID>.tar.gz	Optional tar ball may include additional patches for open source components. All details please see in README file within the tar ball.
media_server_studio_sdk_release_notes.pdf media_server_studio_sdk_getting_started_guide.pdf	Intel® Media Server Studio – SDK documentation: this file, Getting Started Guide.
intel-opencl-16.5-release-notes.pdf intel-opencl-16.5-installation.pdf	OpenCL™ Driver Release Notes and Installation Guide.

Installation Folders

Intel® Media Server Studio – SDK installs under /opt/intel/mediasdk – this is referenced as <sdk-install-dir> in the remainder of this document.

Component	Description
<sdk-install-dir>/lib64	Intel® Media Server Studio – SDK Dynamic Library, hardware implementation libmfxhw64-p.so.* software implementation libmfxsw64-p.so.*
<sdk-install-dir>/doc	Intel® Media Server Studio – SDK documentation

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

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

Page 8 of 27

Copyright © 2015-2016, Intel Corporation

<code><sdk-install-dir>/include</code>	<p>External Intel® Media Server Studio – SDK headers:</p> <ul style="list-style-type: none"> • Structure definitions in <code>mfxstructures.h</code>, <code>mfxastructures.h</code>, <code>mfxvstructures.h</code> and <code>mfxcommon.h</code> • Audio function definitions in C in <code>mfxaudio.h</code> • C++ wrapper for Media SDK audio functions in <code>mfxaudio++.h</code> • Type definitions in <code>mfxdefs.h</code> • <code>mfxVideoENC</code> functions definitions <code>mfxenc.h</code> • <code>mfxVideoPAK</code> functions definitions <code>mfxpak.h</code> • Extensions for Motion JPEG Video coding options <code>mfxjpeg.h</code> • Extensions for standalone Look Ahead algorithm <code>mfxla.h</code> • Extensions for Multi-view Video Coding options <code>mfxmvc.h</code> • Extensions for User-Defined Functions <code>mfxplugin.h</code> • C++ wrapper for User-Defined Functions <code>mfxplugin++.h</code> • Session management function definitions in <code>mfxsession.h</code> • Function definitions in C in <code>mfxvideo.h</code> • C++ wrapper of the SDK functions in <code>mfxvideo++.h</code> • VP8 Extension definition <code>mfxvp8.h</code>
<code><sdk-install-dir>/lib/lin_x64</code>	<p>Intel® Media Server Studio – SDK Static Dispatcher Library:</p> <p><code>libmfx.a</code></p>
<code><sdk-install-dir>/plugins</code>	<p>Intel® Media Server Studio – SDK plug-ins:</p> <ul style="list-style-type: none"> • Advanced AVC Encode plug-in (implements 1:N Look Ahead optimization) <code>libmfx_h264la_hw64.so</code> • VP8 Decode plug-in <code>libmfx_vp8d_hw64.so</code> • HEVC Hardware Decode Plug-in <code>libmfx_hevcd_hw64.so</code> • HEVC Hardware Encode Plug-in <code>libmfx_hevce_hw64.so</code> • Configuration file <code>plugins.cfg</code>

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

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

Page 9 of 27

Copyright © 2015-2016, Intel Corporation

<code><sdk-install-dir>/opensource/mfx_dispatcher</code>	Source code for the Intel® Media Server Studio – SDK Dispatcher
<code><sdk-install-dir>/tools/tracer</code>	SDK Tracer – tool for dumping API level logging information

Intel® Media Server Studio 2017 – Graphics Driver installs in the following locations:

Component	Description
<code><sdk-install-dir>/lib64</code>	Intel® Media Server Studio – Graphics Driver
<code><sdk-install-dir>/opensource/libdrm</code>	Source code for Direct Rendering Manager runtime library
<code><sdk-install-dir>/opensource/libva</code>	Source code for Video Acceleration (VA) API
<code><sdk-install-dir>/opensource/patches/kmd</code>	Source code of Intel® Media Server Studio – Graphics Driver, Kernel Mode Driver (KMD)
<code>/usr/include</code>	Direct Rendering Manager runtime library, Video Acceleration (VA) API includes
<code>/usr/lib64</code>	Direct Rendering Manager runtime library, Video Acceleration (VA) API libraries
<code>/usr/bin</code>	Direct Rendering Manager runtime library, Video Acceleration (VA) API utilities.

OpenCL™ Driver installs in the following locations:

Component	Description
<code>/opt/intel/ocl/include/CL</code>	OpenCL™ Driver includes
<code>/opt/intel/ocl</code>	OpenCL™ Driver libraries

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

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

Page 10 of 27

Copyright © 2015-2016, Intel Corporation

You could find more information about OpenCL™ Driver in corresponding Release Notes in `<sdk-extract-folder>/intel-ocl-16.5-release-notes.pdf`.

Documentation

For the most up to date versions please refer the Intel® Media Server Studio Support [documentation page](#).

You can find more information on how to use Intel® Media Server Studio - SDK in the following documentation:

- `<sdk-install-folder>/doc/mediasdk-man.pdf`
"Intel Media Server Studio - SDK Reference Manual" describes the Intel Media SDK API.
- `<sdk-install-folder>/doc/mediasdkusr-man.pdf`
"Intel Media Server Studio - SDK Extensions for User-Defined Functions" describes an API extension (aka plug-ins API) that allows seamless integration of user-defined functions in SDK pipelines.
- `<sdk-install-folder>/doc/mediasdkjpeg-man.pdf`
"Intel® Media Server Studio - SDK Reference Manual for JPEG*/Motion JPEG" describes SDK API for JPEG* processing.
- `<sdk-install-folder>/doc/mediasdkvp8-man.pdf`
"Intel® Media Server Studio - SDK Reference Manual for VP8*" describes SDK extension to support VP8* video codec.

Known Limitations

This release is subject to the following known limitations:

- **Changes in behavior compared to Media Server Studio 2016 R1 - SDK (version 6.0.16043361.361):**
 - H.264 encoder:
 - H.264 encoder on Broadwell may produce worse objective quality for interlace encoding using B-pyramid. Average quality difference on big stream set is about 0.5% PSNR BDRATE.
 - VPP:
 - Resize, color space conversion, detail and denoise filters may produce not bitexact results compared to previous release. The difference doesn't affect visual quality
 - GPU copying during surface exchange between system and video memory is disabled by default. To enable GPU copy please set `mfxInitParam::GPUCopy = MFX_GPUCOPY_ON`

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

- **Hardware Device Error Handling**

- Application should treat `MFX_ERR_ABORTED` status returned from `MFXVideoCORE_SyncOperation()` as `MFX_ERR_DEVICE_FAILED` and run recovery procedure as described in Hardware Device Error Handling section of the manual

- **API:**

Intel® Media Server Studio - SDK API is designed for a range of products. A particular product release may support only a subset of the features of the declared API version. This release has the following API limitations:

- Only the following features among those introduced in API 1.7 are supported:
 - `RateControlMethod::MFX_RATECONTROL_LA`
 - `mfxExtCodingOption2::LookAheadDepth`
 - `mfxExtCodingOption2::MBBRC`
 - `mfxExtCodingOption2::Trellis`
- Only the following features among those introduced in API 1.8 are supported:
 - `mfxVideoCodecPlugin`
 - `mfxExtVPPComposite`
 - `mfxExtVPPDeinterlacing`
 - `mfxExtCodingOption2::LookAheadDS, RepeatPPS, BRefType`
 - `mfxHandleType::MFX_HANDLE_VA_DISPLAY`
 - `mfxImpl::MFX_IMPL_VIA_VAAPI, mfxIMPL::MFX_IMPL_AUDIO`
 - `CodecFormatFourCC::MFX_CODEC_HEVC, CodecLevel::HEVC level and tier definitions, CodecProfile::HEVC profile definitions`
 - `BRefControl`
 - `mfxFrameData::PitchHigh, PitchLow`
- Only the following features among those introduced in API 1.9 are supported:
 - `mfxExtVPPComposite, mfxVPPCompInputStream::LumaKeyEnable, LumaKeyMin, LumaKeyMax, GlobalAlphaEnable, GlobalAlpha, PixelAlphaEnable`
 - `mfxExtAVCRefLists`
 - `mfxExtAVCEncodedFrameInfo::secondFieldOffset`
 - `mfxExtCodingOption2::SkipFrame, supported for AVC and MPEG2 Encode`
 - `ColorFourCC::MFX_FOURCC_P010, MFX_FOURCC_A2RGB10`
 - `mfxExtCodingOption2::MaxSliceSize`
- Only the following features among those introduced in API 1.10 are supported:
 - `MFXVideoENC class of functions`
 - `mfxENCInput`
 - `mfxENCOutput`

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

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

Page 12 of 27

Copyright © 2015-2016, Intel Corporation

- mfxExtLAControl
- mfxExtLAFrameStatistics
- RateControlMethod::MFX_RATECONTROL_LA_EXT
- mfxExtCodingOption2::BufferingPeriodSEI and enum {MFX_BPSEI_DEFAULT, MFX_BPSEI_IFRAME}
- Only the following features among those introduced in API 1.11 are supported:
 - mfxExtCodingOption3::WinBRCTMaxAvgKbps, WinBRCTMaxAvgKbps
 - mfxFrameData:: NumExtParam, ExtParam
 - mfxExtVPPFieldProcessing, enum VPPFieldProcessingMode, enum PicType
 - RateControlMethod::MFX_RATECONTROL_LA_HRD
 - ExtendedBufferID::MFX_EXTBUFF_CODING_OPTION3, MFX_EXTBUFF_VPP_FIELD_PROCESSING
- Only the following features among those introduced in API 1.12 and 1.13 are supported:
 - mfxExtCodingOption2::UseRawRef
 - mfxExtCodingOption3::DirecrBiasAdjustment, GlobalMotionBiasAdjustment, MVCostScalingFactor, MBDisableSkipMap, EnableMBQP.
 - mfxExtChromaLocInfo
 - mfxExtMBDisableSkipMap
 - mfxExtMBQP
 - MFXVideoUSER_LoadByPath()
 - For mfxExtVPPDeinterlacing extended buffer only DeinterlacingModes MFX_DEINTERLACING_BOB and MFX_DEINTERLACING_ADVANCED are supported
- Only the following features among those introduced in API 1.14 are supported:
 - mfxExtDecodedFrameInfo
 - mfxExtTimeCode
- Only the following features among those introduced in API 1.15 are supported:
 - mfxExtThreadsParam and MFX_EXTBUF_THREADS_PARAM enumerator
- Only the following features among those introduced in API 1.16 are supported:
 - GPUCopy enumerator and control in mfxInitParam structure.
 - mfxFrameAllocRequest.AllocId
 - mfxInfoMFX.MaxDecFrameBuffering
- Only the following features among those introduced in API 1.17 are supported:
 - mfxVideoParam.AllocId **field**
 - DeinterlacingMode::MFX_DEINTERLACIF_ADVANCED_NOREF **enumerator**

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

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

Page 13 of 27

Copyright © 2015-2016, Intel Corporation

- deprecated `MFXVideoCORE_SetBufferAllocator()` function, `mfxBufferAllocator` structure
- deprecated fields:
 - `mfxCodingOption.EndOfSequence`
 - `mfxCodingOption.EndOfStream`
 - `mfxExtCodingOption2.ExtBRC`
- deprecated `MFX_EXTBUFF_VPP_PICTSTRUCT_DETECTION` enumerator
- Only the following features among those introduced in API 1.19 are supported:
 - `MFXVideoCORE_QueryPlatform()` function
 - `MFXVideoENC_GetVideoParam()` function
 - `mfxPlatform` structure
 - `mfxPayload.CtrlFlags` field
 - `mfxExtVPPScaling` structure
 - `mfxStatus (*QueryPlatform)(mfxHDL pthis, mfxPlatform *platform)` for USER interfaces
 - `mfxStatus MFXVideoUSER_GetPlugin(mfxSession session, mfxU32 type, mfxPlugin *par)`

NOTE: Other options may be not supported. Please use Query functions to check feature availability on any given machine at runtime. Availability of features depends on hardware capabilities as well as driver version.

- **Performance:**

- Advanced De-Interlacing provides better quality but might be slower than BOB DI in some cases. This is especially affects N:N multi-transcoding sessions. API control `mfxExtVPPDeinterlacing` provides application control of de-interlacing method.
- The product was fully validated only with the default values of `mfxExtThreadsParam`. Executing application or initializing the SDK library internal threads under real time scheduling policies (`SCHED_FIFO` or `SCHED_RR`) with specific Priority levels may lead to significantly increased latency, increased total processing time and/or increased CPU usage.
- HEVC decoder and encoder tasks can be executed only on VDBOX0 and UMD/KMD will schedule such tasks accordingly.
- Media SDK Sessions Joining API (`MFXJoinSession`) is not recommended to use since significant performance drops can be observed on specific components and use cases. To reduce internal Media SDK threads number consider to use `mfxExtThreadsParam::NumThread` API.
- To get better performance results and performance/resources tradeoff on the N:N multi-transcoding scenarios with the significant number of parallel sessions ($N > 4$) it is recommended to consider synchronous pipeline implementations and initialize components with `mfxVideoParam::AsyncDepth=1`. In case of small number of parallel sessions asynchronous pipelines may give better results, it is recommended to consider `mfxVideoParam::AsyncDepth=2, 3`.

- **HEVC decode:**

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

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

Page 14 of 27

Copyright © 2015-2016, Intel Corporation

- Decoder may cause GPU hang if SPS which changed bit length of syntax elements is completely lost from the bitstream.
- Decoder doesn't support output of field pair interleaved in one frame buffer. During decoding of interlaced stream every field is returned in separate frame buffer with `PicStruct = MFX_PICSTRUCT_PROGRESSIVE`. Polarity of returned field may be obtained from [H265: Picture timing SEI message] using [H265: Table D.2 – Interpretation of pic_struct] process. Application may use `[MFXVideoDECODE_GetPayload]` function to retrieve this SEI message. This function extracts SEI messages from the bitstream into the provided `[mfxPayload]` structure and returns the time stamp for this message (in units of 90 KHz) which can be used to relate this message with the certain frame (`mfxFrameSurface1::Data::TimeStamp`). Note, the decoder internally buffers the messages, so application may call this function multiply times to retrieve all buffered messages. If application uses external timestamps (`mfxBitStream::TimeStamp`), no more than one frame at once should be submitted via `[MFXVideoDECODE_DecodeFrameAsync]`.
- Function `GetPayload()` returns SEI with inserted emulation prevention bytes. If these SEI messages will be sent as Payloads to MSDK HEVC encoder, emulation bytes will be duplicated.
- Function `GetPayload()` returns incorrect timestamps for SEI suffixes messages, these timestamps are equal to the timestamp of the following frame.
- Decoder doesn't properly report hardware errors and GPU hangs from `SyncOperation`. In case of such error `SyncOperation` returns `MFX_ERR_ABORTED`. After this, the proper error (`MFX_ERR_DEVICE_FAILED` or `MFX_ERR_GPU_HANG`) is returned from subsequent calls of `DecodeFrameAsync`.
- When more than one frame is passed to `DecodeFrameAsync` in same input buffer, and external timestamp is set for this input, decoder will use provided timestamp to calculate PTS for 1st frame in the buffer. Decoder will not request additional timestamps for rest of frames in the buffer, and output PTS for these frames will be incorrect.
- "Corrupted" flag in `mfxFrameData` is set by Decoder based on last slice only. I.e., if there are corruptions in previous slices, and no corruptions in last slice, "Corrupted" flag will be set by Decoder to 0.
- Decoder doesn't clear the `[VA_STATUS_ERROR_DECODING_ERROR]` error status for function `[vaSyncSurface]` for the certain output surface. So if application will call this function for frame which caused HW decoding error, `[VA_STATUS_ERROR_DECODING_ERROR]` may be returned. External frame allocator which is present in MSDK sample package calls `vaSyncSurface` in method `[vaapiFrameAllocator::LockFrame]` and converts `[VA_STATUS_ERROR_DECODING_ERROR]` returned from `vaSyncSurface` to `[MFX_ERR_UNKNOWN]`. This `vaSyncSurface` call in allocator is redundant. Applications which use sample code in their allocators need to remove `vaSyncSurface` call to avoid unexpected `MFX_ER_UNKNOWN` status returned from allocator.
- If Decoder is called with flag `MFX_BITSTREAM_EOS` set and incomplete (or absent frame), it will return `MFX_ERR_MORE_DATA` status instead of returning the buffered frames.
- Decoder doesn't properly support API functions `GetDecodeStat()` and `SetSkipMode()`. Both functions always return status `MFX_ERR_NONE`. `GetDecodeStat()` doesn't modify output `mfxDecodeStat` structure. `SetSkipMode()` function doesn't affect decoder operation.

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

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

Page 15 of 27

Copyright © 2015-2016, Intel Corporation

- Decoder may cause GPU hang for frame which starts with highly corrupted slice segments (which can't be decoded and have to be skipped by Decoder), and first decodable slice segment header in the frame (which follows this series of corrupted slice segments) has flag [dependent_slice_segment_flag] set to 1.
- **HEVC encode:**
 - Only following parameters from MSDK API extended buffers are supported by Encoder:
 - mfxExtHEVCParam
 - PicWidthInLumaSamples
 - PicHeightInLumaSamples
 - mfxExtHEVCTiles
 - NumTileRows
 - NumTileColumns
 - mfxExtCodingOption
 - VuiNalHrdParameters
 - NalHrdConformance
 - AUDelimiter
 - mfxExtCodingOption2
 - IntRefType
 - IntRefCycleSize
 - IntRefQPDelta
 - BRefType
 - NumMbPerSlice
 - DisableDeblockingIdc
 - RepeatPPS
 - mfxExtCodingOption3
 - PRefType
 - IntRefCycleDist
 - EnableQPOffset
 - QPOffset
 - NumRefActiveP
 - NumRefActiveBL0
 - NumRefActiveBL1
 - mfxExtEncoderCapability
 - MBPerSec

If any other parameter is set in extended buffer and passed to functions Query, QueryIOSurf, Init or Reset, Encoder will reset it to 0 and will return status MFX_WRN_INCOMPATIBLE_VIDEO_PARAM.

 - Encoder cannot accept Prefix SEI payloads with total size over 256 bytes. As a result of bigger payload error MFX_ERR_DEVICE_FAILED will be returned from SyncOperation().
 - Default GOP structure is different for CQP (7 B-frame B-pyramid) and CBR/VBR (3 frame B-pyramid).
 - Encoder cannot encode regular P-frames. Low-delay B-frames are coded instead. All parameters targeted to P-frames (e.g. -GopRefDist, -QPP) are applied to these low-delay B frames (new). Similarly, if FrameType = MFX_FRAMETYPE_P is passed in encoded order, it's coded as low-delay B-frame.
 - Encoder doesn't support Intra Refresh.

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

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

Page 16 of 27

Copyright © 2015-2016, Intel Corporation

- Support of Bitrate Control is limited. There could be quality problems for encoding with CBR/VBR (e.g. low coding quality for CBR and B-pyramid). ICQ, VCM, QVBR BRC methods aren't supported. LookAhead BRC modes aren't supported.
- Encoder doesn't support "Region Of Interest" feature.
- Encoder doesn't support interlaced encoding.
- Encoder may produce non-bit exact streams from run-to-run. Run to run difference doesn't affect visual quality
- MBBRC option doesn't take effect. For CBR/VBR macroblock-based BRC is always turned on for TargetUsage 1, 2, and always turned off for TargetUsages 3-7.
- Parameter NumMBPerSlice is treated as number of coding units of maximum size supported by underlying HW. For Skylake it's 32x32. May vary for future platforms.
- Encoder doesn't support insertion of HRD information to coded stream. At the same time Encoder tries to keep HRD conformance for given BRC and HRD parameters. E.g. padding will be inserted for CBR BRC mode.
- Padding for CBR is done by series of trailing_zero_8bits syntax element.
- For TargetUsage 7 encoder produces on-par or worse RD quality than AVC encoder for non-linear types of motion (e.g. rotation, shaking). Work around is to set multiple references using NumRefActiveP, NumRefActiveBL0, NumRefActiveBL1.
- If application needs to disable deblocking, it needs to always set respective value of DisableDeblockingIdc if extended buffer MfxExtCodingOption2 is sent to encoder in runtime. If value isn't set (equals to 0), and buffer is sent to EncodeFrameAsync, zero value will be applied to current frame, and deblocking will be turned on.
- Attempt to create HEVC encoder on Broadwell will cause segmentation fault in Init(), Query() and QueryIOSurf() functions.
- Encoder does not support resolution 16x16. MFX_ERR_DEVICE_FAILED will be returned on attempt to create encoder with such small resolutions from MSDK Init() function. Meanwhile Query() function will return MFX_ERR_NONE.
- Values of PicWidthInLumaSamples, PicHeightInLumaSamples must be multiple of 16.
- For very complex content BufferSizeInKB reported by GetVideoParam() may not be enough to hold coded bitstream. If it happen, HEVC encoder will return MFX_ERR_NOT_ENOUGH_BUFFER in runtime.
- If in EncodedOrder mfxEncodeCtrl.FrameType is set to MFX_FRAMETYPE_P | MFX_FRAMETYPE_REF and mfxExtHEVCRefLists contains non-empty L1 list, output frame will contain visual artifacts (as well as all frames which use this frame for reference).
- If in EncodedOrder application will send to Encoder equal FrameOrders for subsequent frames (e.g. all zeroes, or some other value), Encoder may hang inside EncodeFrameAsync call.
- Encoder cannot change frame rate via Reset() call (new value will be ignored). But if frame rate is changed along with bitrate by same Reset() call, encoder will apply new value.

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

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

Page 17 of 27

Copyright © 2015-2016, Intel Corporation

- Encoder may return MDX_ERR_DEVICE fail in case of very high bitrates or low QP when compression rate is about 1. Is it required to call reset and do re-encode with lower bitrate or higher QP.
- For CQP encoding of very complex content with low QP BufferSizeInKB reported by GetVideoParam() may not be enough to hold coded bitstream. If it happens, HEVC encoder returns MFX_ERR_NOT_ENOUGH_BUFFER from SyncOperation. Application may workaround it by allocation of bigger buffer than MSDK reported in BufferSizeInKB. Following value is enough: <frame size in pixels> * 3.
- **H.264 decode:**
 - The H.264 decoder may leave Corrupted flag as 0 in case of minor corruption in macroblock bitstream data.
 - Decoder returns MFX_ERR_UNSUPPORTED for streams which cannot be processed by hardware, software fallback was removed.
 - For decoders it's not guaranteed that GPU hang will be timely reported from SyncOperation (there could be several frames delay). MFX_ERR_GPU_HANG will be timely reported from DecodeFrameAsync call.
 - Function GetPayload() returns SEI with inserted emulation prevention bytes. If these SEI messages will be sent as Payloads to MSDK HEVC encoder, emulation bytes will be duplicated
- **H.264 encode:**
 - Reporting of per-frame QP via mfxExtAVCEncodedFrameInfo isn't supported by Encoder (zero QPs are always returned).
 - Encoder doesn't support use of MaxFrameSize and MaxSliceSize together. If MaxSliceSize is set, MaxFrameSize is ignored.
 - GOP structure may be non-optimal at the place of forced IDR insertion. E.g. B-frame could be reordered over forced IDR and encoded with only backward references (new).
 - Reset function isn't supported for LookAhead BRC modes (except MaxSliceSize mode). Reset returns MFX_ERR_INVALID_VIDEO_PARAM for LA BRC.
 - LookAhead BRC modes (including MaxSliceSize mode) don't support CAVLC coding. CAVLC will be switched to CABAC for LA BRC.
 - Downscale factor 2 of LookAhead BRC modes may produce better objective coding quality than scale factor 1
 - 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).
 - Encoder may produce non-bit exact streams from run to run on Broadwell and Skylake. Run to run difference doesn't affect visual quality.
 - Chroma artefacts are possible at very low bitrates (when QP is close to 51).

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

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

Page 18 of 27

Copyright © 2015-2016, Intel Corporation

- Support for encoded frame info (mfxExtAVCEncodedFrameInfo) is limited with SecondFieldOffset for interlace coding, and FrameOrder, LongTermIdx UsedRefListL0/L1 for progressive coding.
- 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.
- Usual Look Ahead BRC may generate non HRD-compliant streams.
- Look Ahead BRC (MFX_RATECONTROL_LA_HRD) may not give quality gain for B-pyramid case (or give very little gain in comparison with regular B-frame configuration)
- 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.
- MBBRC option is enabled by default on lower target usages but can be switched off. Exact implementation details are hidden and may change with time and between platforms, so using 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.
- MBQP mode is applicable only for CQP BRC and can be set in value range 1-51.
- In very rare cases Encoder may produce GPU hang for very specific input content (static flat texture with small portion of very complex blocks - like white noise).
- Encoder doesn't release Locked counters of input surfaces in Close() function.
- Encoder doesn't support WeightedPrediction and FadeDetection features.
- For very complex content BufferSizeInKB reported by GetVideoParam() may not be enough to hold coded bitstream. If it happen, AVC encoder will cut bytes that don't fit to output buffer.
- If application needs to disable deblocking, it needs to always set respective value of DisableDeblockingIdc if extended buffer MfxExtCodingOption2 is sent to encoder in runtime. If value isn't set (equals to 0), and buffer is sent to EncodeFrameAsync, zero value will be applied to current frame, and deblocking will be turned on.
- Turning ON Trellis for B-frames may slightly worsen objective quality of coded streams in terms of BDPSNR.

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

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

Page 19 of 27

Copyright © 2015-2016, Intel Corporation

- Encoder cannot change frame rate via Reset() call (new value will be ignored). But if frame rate is changed along with bitrate with same Reset() call, encoder will apply new value.
- Encoder cannot accept SEI payloads with total size over ~1130 bytes (the exact number may slightly vary depending on Encoder configuration). As a result of bigger payload error MFX_ERR_DEVICE_FAILED will be returned from SyncOperation().
- Parameters MaxQP and MaxFrameSize may contradict each other. If for particular frame MaxQP doesn't give BRC to reach MaxFrameSize, MaxQP is obeyed, and MaxFrameSize is violated.
- Encoder may produce stream with HRD violation if application skips frames using SkipFrame interface.
- Encoder may produce visual artifacts at B-frames when encoding with mixed progressive/interlaced picture structure. Work around is not to use B-frames together with mixed picture structure.
- Encoder may produce significantly higher bitrates than requested when coding with frequent alternation of progressive/interlaced picture structure (e.g. about 40% bigger bitrate when picture structure is alternated for every other frame). Behavior in such stress conditions may significantly differ between current release and previous release (MSS2016 R1) (e.g. resulting stream may have tenth of percent of BDRATE difference with either better or worse quality)
- When I/P field pair is encoded at the beginning of new GOP, P field in the pair may use frames from previous GOP as reference. So I/P field pairs can't be used as clean random access points, unlike IDR/P field pairs.
- Parameters that are part of PPS header in AVC standard (e.g. weighted prediction) can't be correctly applied on per-frame level if RepeatPPS option is set to OFF during encoder initialization.
- Encoder implements "force-to-skip" functionality: Inter frame is coded w/o coefficients and motion vectors if there is a risk of HRD underflow. But this feature doesn't guarantee HRD compliant encoding. In rare cases for very low bitrates and very complex content HRD underflow may happen (e.g. 500 kbps 1080p and white noise content). It's actual for both Skylake and Broadwell.
- Rolling Intra refresh with MB-rows doesn't work correctly - all MBs below current Intra region in the frame are forced to Intra as well. Intra refresh with MB-columns could be used as work around.
- Encoder produces different results for synchronous and asynchronous pipelines for TFF interlace encoding with TU7 on Broadwell. Visual quality isn't affected by this difference.
- For encoding with temporal layers together with multislice, only first slice in a picture has SVC prefix NALu prior to slice header (there are number of prefix NALus equal to number of slices).
- On Broadwell Encoder may produce visual quality drops when motion/scene change comes after long simple static scene. E.g. in screen capture when window is popping up after several seconds of still picture.
- Encoder may produce different output if cases when size of input surface is bigger than frame resolution, and when size of surface equal to frame resolution. Difference doesn't affect objective quality, and isn't visible by eye.

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

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

Page 20 of 27

Copyright © 2015-2016, Intel Corporation

- H.264 encoder may produce lower objective quality on Skylake in comparison with Broadwell. Average quality difference on big stream set is about 1% PSNR BDRATE.
 - When number of slices is controlled by parameters NumSlice, NumSliceI, NumSliceP, NumSliceB, resulting slices are always aligned to MB-row. In addition all slices have equal sizes (except last one). Such alignment may cause change of slice number set by application. Application should use NumMBPerSlice parameter to get slices of arbitrary MB size.
 - Following restrictions are applied to reference lists set via mfxExtAVCRefLists for interlaced encoding. List should start from reference field with same polarity as current one. Fields in reference lists should alternate while it's possible to pick field of alternative parity. If listed restrictions are violated, coded field may contain visual artifacts, or be encoded with Intra macroblocks only.
 - Enabling or disabling of Rolling Intra Refresh by Reset() call will lead to insertion of IDR. E.g. if Encoder is initialized with IntRefType = 0 (disabled refresh), and IntRefType = 1 is sent to Reset call, IDR will be inserted right after Reset, and only then first Intra refresh cycle will be started.
 - For CQP encoding of very complex content with low QP BufferSizeInKB reported by GetVideoParam() may not be enough to hold coded bitstream. If it happens, AVC encoder truncates coded bitstream to fit to the buffer provided by application and returns MFX_ERR_NONE from respective SyncOperation call. Application may workaround it by allocation of bigger buffer than MSDK reported in BufferSizeInKB. Following value is enough: <frame size in pixels> * 3.
- **MPEG-2 decode:**
 - Decoder does not support bitstreams with resolution bigger than 2048x2048. 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 (2048x2048) or level/chroma are invalid. It affects MFXVideoDECODE_DecodeFrameAsync.
 - For decoders it's not guaranteed that GPU hang will be timely reported from SyncOperation (there could be several frames delay). MFX_ERR_GPU_HANG will be timely reported from DecodeFrameAsync call.
 - If incoming bitstream is corrupted decoder may skip corrupted slices or entire frames. Corruptions in bitstream does not influence on decoding of next frames in bitstream that are not corrupted.
 - Decoder returns MFX_ERR_UNSUPPORTED for streams which cannot be processed by hardware, software fallback was removed.
 - Decoder may return MFX_ERR_NONE but a zero sync point on corrupted streams. Application should explicitly check for non-zero sync point to get decoded valid and skip frames with zero sync points.
 - **MPEG-2 encode:**
 - Encoder may produce non-bit exact streams. Run to run difference doesn't affect visual quality.

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

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

Page 21 of 27

Copyright © 2015-2016, Intel Corporation

- The MPEG-2 encoder may produce output that under-runs the MPEG-2 video buffer verifier model (VBV) on some streams. We suggest the following guideline of the parameter values to be followed to keep VBV compliance.
 - MPEG2 buffer usage is mainly restricted by the number of bits used for I frame. The minimum size of each 16x16 blocks of intra frame at highest QP is about 50 bits. The minimum initial buffer fullness (InitVBVBufferFullnessInBit) should be at least twice the size of the initial I frame, and the minimum buffer size (vbv_buffer_size) should be twice of the initial buffer fullness (4 times of the initial I frame).
- MBQP mode is applicable only for CQP BRC and can be set in value range 1-122.
- SkipFrame feature has the following limitations:
 - works only with CQP BRC mode
 - only MFX_SKIPFRAME_INSERT_DUMMY is supported
 - If GOP has only P frames, arbitrary P can be skipped. When skipped, it is made non-reference.
- Software implementation of MPEG-2 Encoder may erroneously insert a duplicated field in interlace field encode mode on specific content.
- Setting too low bitrate for MPEG-2 Encoder may produce mosaic visual artifacts on complex content with fast motion or scene changes. For example bitrate 5.6 Mbps is too low for 1080@25p, increasing bitrate to 8.5 Mbps produce much better quality stream.
- Resetting MPEG-2 Encoder with new aspect ratio may return MFX_ERR_INCOMPATIBLE_VIDEO_PARAM, workaround - explicitly close and re-initialize encoder.
- Contrary to SDK Reference Manual (mediasdkman.pdf) MPEG-2 Encoder may allocate surfaces on Reset call.
- Encoder doesn't return error on attempt to initialize it with some unsupported parameters (e.g. WeightedPrediction, FadeDetection and so on). Encoder ignores such parameters and returns MFX_ERR_NONE from Query/Init/Reset functions.
- Once Encoder is initialized with some resolution, Encoder doesn't return error status on attempt to submit frame of bigger resolution to EncodeFrameAsync. Frame is accepted, consequence may be corrupted output, segmentation fault or some other undefined behavior.
- **JPEG/MJPEG decode and encode** support only the below feature set:
 - Baseline mode only
 - DCT based
 - 8-bit samples
 - sequential
 - loadable 2 AC and 2 DC Huffman tables
 - 2 loadable quantization matrixes
 - interleaved and non-interleaved scans
 - single and multiple scans

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

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

Copyright © 2015-2016, Intel Corporation

Page 22 of 27

- No extended, lossless and hierarchical modes
 - no 12-bit samples
 - no progressive
 - no arithmetic coding
 - no 4 AC and 4 DC Huffman tables
- JPEG/MJPEG hardware accelerated decoder supports resolutions ≤ 8192 . In case of bigger resolution fallback to SW will be notified via MFX_WRN_PARTIAL_ACCELERATION from Init/Query/QueryIOSurf functions.
- Decoder supports YUV2 output format and multi-scan pictures only by SW fallback.
- Hardware accelerated encoder (available on Skylake only) supports only NV12, RGB32 input color formats with SW fallback in other cases.
- MJPEG Encoder doesn't encode correctly nv12 monochrome format input.
- Decoder doesn't support 4:4:4 chroma subsampling (chroma subsamplings 4:2:0, 4:2:2, 4:1:1 are supported).
- Decoder with rotation doesn't support rgb32 and yuy2 formats, it supports nv12 pixel format only.
- Decoder doesn't properly support output surfaces pool bigger than 128. MFX_ERR_UNSUPPORTED may be returned on attempt to use such big surface pool for decoder output.
- Decoder doesn't support GPU hang reporting.
- Decoder may produce non bitexact output from run to run without visual artifacts.
- **VC1 decoder:**
 - Decoder may cause GPU hangs and return status MFX_ERR_UNDEFINED_BEHAVIOR during decoding of corrupted content.
 - Decoder doesn't support GPU hang reporting.
- **VPP:**
 - 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.
 - Field copy processing cannot be used with any other VPP filters including resize and color conversion. In case field processing is requested, all other VPP filters are skipped without error/warning messages.
 - Frames with interlaced content must have CropH multiple of 4. Otherwise, VPP may produce color artefacts on the bottom lines.
 - VPP interpolated FRC was deprecated and replaced by simple arbitrary FRC. Setting FRC algorithm to MFX_FRCALGM_FRAME_INTERPOLATION in mfxExtVPPFrameRateConversion extended buffer will cause fallback to MFX_FRCALGM_PRESERVE_TIMESTAMP algorithm. No errors/warning are returning by MSDK in this case.
 - 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

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

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

Page 23 of 27

Copyright © 2015-2016, Intel Corporation

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.
- When composition is used for 8 or more channels AND at least one channel has GlobalAlphaEnable=true AND first channel has GlobalAlphaEnable=false => first channel is composed with artifacts looking like certain global alpha value was applied to it despite GlobalAlphaEnable is false.

Workaround: For the first stream, replace GlobalAlphaEnable=false with GlobalAlphaEnable=true plus GlobalAlpha=255. This combination is visually equal to GlobalAlphaEnable=false and works correctly.

- De-interlacing is supported for NV12, YUY2 formats only.
- 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.
- Every frame is doubled during 30i->60p de-interlacing for MFX_DEINTERLACING_BOB and MFX_DEINTERLACING_ADVANCED_NOREF modes
- The first frame is doubled for MFX_DEINTERLACING_ADI and MFX_DEINTERLACING_ADI_SCD.
- MFX_DEINTERLACING_ADI can produce color artifacts and out of order frames after scene change. MFX_DEINTERLACING_ADI_SCD should be used.
- MFX_DEINTERLACING_ADI_SCD is handling frame with scene change and two or four frames after with BOB algorithm producing frame from the second field (from bottom in case of TFF and from top in case of BFF). Number of frames which is processed by BOB depends on deinterlacing mode: two for 30i->30p and four for 30i->60p.
- MFX_DEINTERLACING_ADI and MFX_DEINTERLACING_ADI_SCD may produce color artefacts in case of harmonic motion (repeated pattern and motion magnitude is the same as the periodic of repeated pattern).
- Using of MFX_DEINTERLACING_ADVANCED_SCD for doubled framerate mode (e.g. 30i->60p) may lead to visual quality issues (artifacts, out of order frames) after scene change. MFX_DEINTERLACING_ADVANCED_NOREF should be used for these configurations.
- VPP doesn't support GPU hang reporting

Workaround: Instead of a dynamic switch between the modes, application should create separate VPP components and initialize them with required settings. To process frames application should chose a VPP instance based on parameters passed at initialization.
- Dynamic switch between field copy processing and frame copy processing modes causes a memory leak.
- VPP doesn't support standalone scene change detection filter and ignores MFX_EXTBUFF_VPP_SCENE_CHANGE, MFX_EXTBUFF_VPP_SCENE_ANALYSIS extended buffers

- **Misc:**

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

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

Page 24 of 27

Copyright © 2015-2016, Intel Corporation

- Only the following functionality from software implementation of the SDK is fully validated and has production quality: VPP scaling and color conversion. All the rest (decoders, encoder, other VPP filters) is provided for demo purpose only and is not guaranteed to have production quality.
- Due to specifics of GPU Copy implementation it is required to close/destroy SDK associated resources (including VADisplay and frame surfaces) only after MFXClose call.
- Using system memory as input/output with mfxInitParam::GPUCopy set to MFX_GPUCOPY_ON has restriction for system memory allocation to be aligned at 64 bytes.
- Encode quality may be different (non-bit exact) between CPU generations.
- In case of executable compiled with -rdynamic or SDK loaded by application plugin, Dispatcher linker symbols would conflict with SDK symbols. To mitigate symbol conflict application should:
 - link against dispatch_shared.a instead of libmfx.a
 - define MFX_DISPATCHER_EXPOSED_PREFIX before any SDK includes

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

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

Copyright © 2015-2016, Intel Corporation

Page 25 of 27

Legal Information

THIS DOCUMENT CONTAINS INFORMATION ON PRODUCTS IN THE DESIGN PHASE OF DEVELOPMENT.

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.

VP8 video codec is a high quality royalty free, open source codec deployed on millions of computers and devices worldwide. Implementations of VP8 CODECs, or VP8 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 SSSE3 instruction sets and other optimizations. Intel does not guarantee the

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

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

Page 26 of 27

Copyright © 2015-2016, Intel Corporation

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

Attributions

Safe C Library

Copyright (C) 2012, 2013 Cisco Systems

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

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

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

Copyright © 2015-2016, Intel Corporation