Intel® Media SDK 2017 for Embedded Linux*
Release Notes for Version
IntelMediaSDK2017R2.1forEmbeddedLinux

Overview

Intel® Media Software Development Kit 2017 for Embedded Linux* (Intel Media SDK) is an SDK for optimizing datacenter and embedded media applications for Linux operating systems to utilize Intel® HD Graphics hardware acceleration capabilities.

The package includes the following components:

- SDK, version **7.0.16073710**
- Samples, version **8.1.24.0**

This particular package was tested only on Yocto Project* and specifically targets its operation environment.

This document covers product features, system requirements and known limitations. For installation procedures description please see the

`<unpack-folder>/mediasdk_getting_started_guide.pdf`.

This package is available at the Intel® Developer Zone for download as IntelMediaSDK2017R2.1forEmbeddedLinux.tar.gz, which corresponds to Maintenance Release 3.1.

The previous Intel® Developer Zone release, IntelMediaSDK2017R2forEmbeddedLinux.tar.gz, corresponds Maintenance Release 3.0.

What's New

Version 7.0.16073710

- SDK was tested with Graphics Driver version 16.7.3.64751.

- This is a production release.
- The size of the sliding window is limited to 1 second. WinBRCSize will be adjusted to FPS at encoder initialization. In order to minimize quality degradation during encoding with sliding window enabled, it is recommended to disable B-frames (GopRefDist=1) and set GopPicSize >= FPS to avoid the case where multiple I-frames are contained within the sliding window.
- When converting 30i content to 30p, the application is required to issue an MFXVideoVPP_Reset call at transition points. Failure to do so may result in incorrectly rendered frames.
- The MediaSDK manual (mediasdk-man.pdf) documents that a decoder parameter 'DecodedOrder' was deprecated. This parameter is still valid and supported for AVC decode only.
- Fixes added for user builds of the dispatcher and the samples in a non-X11 environment.

Version 7.0.16073704

- SDK was tested with Graphics Driver, version **16.7.62740**
- This is a Production release;
- Added support for QP map in CQP mode for HEVC encoder, extended sample to demonstrate proper mode of operations for supported codecs; for compatibility with AVC based applications granularity of the QP map should be 16x16, whereas actual CU QP granularity will be determined by encoder mode decisions and platform limitations;
- Enabled support for sliding window constraints in CBR and VBR modes of the AVC encoder (disabled by default); setting up non-zero `WinBRCMaxAvgKbps` or `WinBRCSize` fields in `ExtCodingOption3` structure will impose additional constrains on BRC algorithm resulting in better target and GOP-based bitrate accuracy at cost of quality degradation;
- Removed 6K resolution limitation for several VPP filters;
- Fixed performance regression in the video compositing of the equally-sized windows;

Known limitations:

- HEVC encoder will not insert suffix SEI's with enabled HDR (default mode for CBR and VBR); set `mfxExtCodingOption.VuiNalHrdParameters` to OFF to disable HRD compliance;
- Using opaque memory for 1:N pipelines may cause occasional deadlocks; using GPU memory is recommended for such workflows;
- Using opaque memory with transcoding pipeline and closing/initializing VPP component may cause MFX_ERR_DEVICE_BUSY error; using GPU memory is recommended for such workflows;

Version 7.0.16073696

- SDK was tested with Graphics Driver, version **16.7.62740**

- This is a Production release;
- New MediaSDK API 1.24 was introduced, adding support for non-uniform video-wall compositing by explicit tiling (for detailed description, refer to MediaSDK manual):
    - `TileId` field to `mfxVPPCompInputStream` structure;
    - `NumTiles` field to the `mfxExtVPPComposite` structure;
- Added deltaQP based ROI support for HEVC encoder working in CBR or VBR modes; due to implementation constraints, HEVC ROI size will be extended to minimal enclosing rectangle with top-left corner and width/height to be aligned to 32 pixels;
- Samples are now demonstrate the usage model for per-frame MB QP map for the AVC encoder;

Known limitations:

- While encoding the interlaced content with AVC encoder the same ROI submitted during EncodeFrameAsync will be used to encode for both top and bottom fields of the frame;

Version 7.0. 16073682

- SDK was tested with Graphics Driver, version **16.7.60601**
- This is a Hot Fix release;
- Fixed: VPP reset operation unlocks all frames for ADI deinterlacer;
- Fixed: VC1d can decode single frame streams to arbitrary surface allocated by external allocator;

Version 7.0. 16073669

- SDK was tested with Graphics Driver, version **16.7.60601**
- This is a Hot Fix release;
- Fixed M/JPEG encoder operation in asynchronous mode;

Version 7.0. 16073664

- SDK was tested with Graphics Driver, version **16.7.60601**
- New MediaSDK API 1.22 was introduced, adding
    - support for deltaQP based ROI encoding in CBR/VBR mode for AVC; refer to `mfxExtEncoderROI` buffer description in the API manual;
    - support for direct pipe scaling for AVC progressive content; refer to `mfxExtDecVideoProcessing` buffer description in the API manual;
    - support for fields weaving and fields splitting; Fields weaving is enabled by setting `MFX_PICSTRUCT_FIELD_SINGLE` as input `PicStruct` value and setting double height in output VPP initialization parameters. Fields splitting is enabled by setting `MFX_PICSTRUCT_FIELD_SINGLE` as output `PicStruct` value

and setting half-double height in output VPP initialization parameters. The opposite PicStruct value in initialization parameters should

  - Either be set to `MFX_PICSTRUCT_FIELD_TFF` or `MFX_PICSTRUCT_FIELD_BFF` at initialization time. If TFF/BBF pattern change is required, it should be done by calling VPP Reset function with new parameters.
  - Or be set to `MFX_PICSTRUCT_UNKNOWN`. In this case, VPP chooses a real TFF/BFF pattern based on a `PicStruct` value in input (for field splitting) or output (for field weaving) `mfxFrameSurface` passed to `RunFrameVPPAsync` function.

- The H264 decoder stops skipping IDR frames with nonzero frame number. Such frames had been considered as invalid. Instead of skipping these frames the decoder begins to handle them as not an LTR frames.
- Behavior of GetPayload function of the MPEG2 decoder have changed. The decoder does not remove payloads from an internal cache when buffer provided from application is not enough to store a payload. Application is able to retrieve this payload by providing a bigger buffer.
- Maximum number of streams supported by VPP composition filter was increased to 72 channel;
- VPP filters support frames up to 8Kx8K resolution; VPP composition filter is now uses `VA_BLEND_PREMULTIPLIED_ALPHA` mode when compositing surfaces with `PixelAlphaEnable` set to 1, while `LumaKeyEnable` set to 0;
- Fixed issue with intensity compensation handling for interlaced VC1 content;
- Fixed issue with surface deallocation logic for ADI VPP during Reset operation;
- Samples versioning system was modified, new version 1.0.0.0 corresponds to 7.0.16073667 in the old notation;
- Due to AVC BRC update, tradeoffs between compression quality and rate were adjusted; apparent decrease of video quality as measured by objective VQM (ex. PSNR) are compensated by corresponding increase of compression rate and vice versa; this is expected behavior;

Known Limitations

- M/JPEG encoder should be used in synchronous mode of operations (`AsyncDepth=1`) to avoid potential segmentation faults in the library;
- Using more than 128 surfaces per session for VC1/VP8/VP9 decoding may sometimes result in erroneous output and is not recommended;
- BRC with enabled ROI’s operates in “best effort” mode, stressing it with extreme deltaQP values or very low bitrates may cause target bitrate mismatch;
- Using fields weaving and fields splitting functionality may cause minor memory leak; this issue will be addressed in the next release;
- Corrupted MPEG2 streams may cause temporary GPU hangs resulting in discontinuities during playback or possible pipeline termination with various error codes, such events are recorded in dmesg log;

- This is Production release
- SDK was tested with Graphics Driver, version 16.7.57168
- This release features general decoding stability improvements on corrupted HEVC content, add support for per frame MaxFrameSize control to AVC encoder BRC, smoother rendering during video playback in samples; remove 128 surfaces per session limitation for H.264/HEVC/MPEG2/MJPEG/VP8 decoders

Known Limitations

- Setting aperture size to 512Mb in BIOS (default = 256Mb) may cause system instability and not recommended;
- Concurrent graphical memory operations in parallel with HEVC encoder under specific conditions may cause temporary GPU hangs resulting in discontinuities during playback or possible pipeline termination with various error codes, such events are recorded in dmesg log;

Version 7.0.16063541

- This is Production release
- SDK was tested with Graphics Driver, version 16.6.55849
- H264 encoder, HEVC (H265) encoder, MJPEG encoder are tested for this release
- VP9 decoder is introduced as a preview component

Version 7.0.16063528

- This is Production release
- SDK was tested with Graphics Driver, version 16.6.55737

Known Limitations

- H264 encoder, HEVC (H265) encoder, MJPEG encoder are not tested for this release

Version 7.0.16062456:

- This is Beta release
- SDK was tested with Graphics Driver, version 16.6.53604

New features:

- New MediaSDK API 1.19 was introduced
- MediaSDK is built against freedesktop.org libva staging branch
- Enabled support:
    - VP8 decoder
    - H.265 encoder
    - VPP filters: Rotation, Sharpness
- H.265 encode: support of SEI payload insertion was implemented

- H.264, H.264, MPEG2 codecs follow GPU hang recovery procedure
- VP8 decoder supports OPAQ memory
- VPP scaling filter allow to choose scaling algorithm
- 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.
- GPUcopy functionality is supported but disabled by default, please consult with SDK manual and samples readmes.

Bug fixes:

- Fixed several memory leaks in different components

H.265 encoder:

- Fixed the issue with transcoding scenarios when encoder may start using input surfaces at the time when they aren't finalized by decoder.
- Fixed the memory leak issue.
- Fixed the issue with support dynamic change of slice number via Reset call which lead GPU hang and stream corruption.

H.265 Decoder:

- extended support of 8K resolutions up to 8192x8192

H.264 encoder:

- Fixed the issue with encoding corrupted slices for frames from non-base layer when temporal scalability is used.

H.264 Decoder:

- Implemented ability to recover after broken MMCO syntax and after some corruptions in IDR slice headers.

Mpeg2 decoder:

- Fixed synchronization call sequence logic. This gave ~25 performance improvements and ~50% CPU reduction on particular use cases with system memory on the output

VPP:

- Usage of allocated surfaces with bigger resolution than specified per MFXInit was allowed.

Samples changes:

- Sample_vpp updated by adjusted test_vpp
- Rotation support was enabled
- I420 as destination color conversion was excluded from support

Version 6.0.16052410:

- This is Alpha 3 release
- SDK was tested with Graphics Driver, version 16.5.50228
- Enabled support:
  - VC1, MPEG2, MJPEG, H.265 decoders

    - H.264, MJPEG encoders
    - VPP filters: Composition, CSC, Denoise, FRC, PROCAMP (HSBC)
- Enabled ability to run VPP SFC resize filter
- Fixed the issue with background color during VPP composition
- Fixed issue with setenv command after root login
- Fixed issue with building samples by GCC 5.2 C++ compiler

Version 6.0.16051360.360:

- This is Pre-Release Alpha 2 version
- SDK was tested with Graphics Driver, version 16.5.46972
- MediaSDK was built against new libva in 46972 driver.
- Sample_decode:
    - fixed for VPP API, VPP cropping issues
    - Fix for Wayland mouse event lag/flickering
    - Performance mode was added

Version 6.0.16051301.301:

- This is Pre-Release version
- SDK API is 1.17
- SDK was tested with Graphics Driver, version 16.5.44864
- Intel® microarchitecture code name "Intel® AtomTM E3900 SoC Family" support
- H.264 decode and basic VPP functions (resize, color conversion, frame rate conversion, deinterlacing) support
- Enabled supports of rendering for Wayland compositor by samples
- Sample_decode received support of VPP resize filter

Detailed description of the new API features can be found in Intel® Media SDK Reference Manual "`<sdk-install_folder>/doc/mediasdk-man.pdf`"

For information on the USER class please see "`<sdk-install_folder>/doc/mediasdkusr-man.pdf`"

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

"`<sdk-install_folder>`" means the location where Media SDK installed, by default it is `/opt/intel/mediasdk`.

Features

Hardware accelerated Intel® Media SDK Library included in this package implements Intel Media SDK API 1.22 and contains the following components:

Component	Supported features	Limitations
H.265 decoder	Supported Profiles: • Main	Maximum supported resolution: 8192x8192
H.265 encoder	Supported Profiles: • Main Supported BRC methods: • Constant QP (CQP) • Constant Bit Rate (CBR) • Variable Bit Rate (VBR)	Maximum supported resolution: 4096x2304
H.264 decoder	Supported Profiles: • Baseline • Main • High	Maximum supported resolution: 3840x2160
H.264 encoder	Supported Profiles: • Constrained Baseline • Main • High Supported BRC methods: • Constant QP (CQP) • Constant Bit Rate (CBR) • Variable Bit Rate (VBR)	Maximum supported resolution: 4096x4096
MPEG-2 decoder	Supported Profiles: • Simple • Main	Maximum supported resolution: 1920x1088

	• High	
MJPEG decoder	Supported Profiles: • Baseline mode, 8bit	Maximum supported resolution: 8192x8192
MJPEG encoder	Supported Profiles: • Baseline mode, 8bit	Maximum supported resolution: 8192x8192
Video Pre Processing (VPP)	Supported Algorithms: • Color Conversion • Scaling • De-Interlacing (Advanced motion-compensated, BOB) • De-noising • Frame Rate Conversion • Composition • Alpha Blending • Sharpness • PROCAMP • Rotation	Maximum supported resolution: 8192x8192
VP8 decoder	Supported Profiles: • Version 0 mode, 8bit	Maximum supported resolution: 1920x1088
VP9 decoder	Supported Profiles: • Profile 0 (8-bit), 8bit	Maximum supported resolution: 4096x2304
VC1 decoder	Supported Profiles: • Simple • Main • Advanced	Maximum supported resolution: 1920x1088

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

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

System Requirements

Hardware
The following processor models are supported:

- Intel Atom® Processor E3900 Series, Intel® Celeron® Processor N3350, Intel® Pentium® Processor N4200
- Minimum 4 GB of DRAM is required, dual channel mode is preferred.

Software
This version of Media SDK is tested with:

- Yocto Project* Version 2.0.3 Jethro (https://www.yoctoproject.org/)
- Kernel version 4.1.42 (http://www.kernel.org)

Package Contents

Intel® Media SDK 2017 for Embedded Linux* package includes the following components, where `<id>` is package identifier:

Component	Description
`install_media.sh`	Optional Media SDK installation script.
`intel-linux-mediasdk-<id>.el7.x86_64.rpm`	Intel® Media SDK 2017 – SDK runtime package.
`intel-linux-mediasdk-devel-<id>.el7.x86_64.rpm`	Intel® Media SDK 2017 – SDK development package.
`intel-linux-mediasdk-<id>.yocto.x86_64.tgz`	Intel® Media SDK 2017 – combined SDK tar ball.
`intel-linux-mediasdk-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.

`mediasdk_release_notes.pdf`
`mediasdk_getting_started_guide.pdf`
`Intel(R)_MediaSDK_EULA.pdf`
`redist.txt`
`site_license_materials.txt`
`third_party_programs.txt`
	Intel® Media SDK 2017 documentation: this file, Getting Started Guide, EULA, EULA’s accompanying files.

Installation Folders

Intel® Media 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 SDK Dynamic Library, hardware implementation
	`libmfxhw64-p.so.*`
	software implementation
	`libmfxsw64-p.so.*`

`<sdk-install-dir>/doc`	Intel® Media SDK documentation
`<sdk-install-dir>/include`	External Intel® Media SDK headers: • Structure definitions in `mfxstructures.h`, `mfxastructures.h`, `mfxvstructures.h` and `mfxcommon.h` • Audio function definitions in C in `mfxaudio.h` • C++ wrapper for Media SDK audio functions in `mfxaudio++.h` • Type definitions in `mfxdefs.h` • `mfxVideoENC` functions definitions `mfxenc.h` • `mfxVideoPAK` functions definitions `mfxpak.h` • Extensions for Motion JPEG Video coding options `mfxjpeg.h` • Extensions for standalone Look Ahead algorithm `mfxla.h` • Extensions for Multi-view Video Coding options `mfxmvc.h` • Extensions for User-Defined Functions `mfxplugin.h` • C++ wrapper for User-Defined Functions `mfxplugin++.h` • Session management function definitions in `mfxsession.h` • Function definitions in C in `mfxvideo.h` • C++ wrapper of the SDK functions in `mfxvideo++.h` • VP8 Extension definition `mfxvp8.h`
`<sdk-install-dir>/lib/lin_`	Intel® Media SDK – Static Dispatcher Library:

`x64`	`libmfx.a`
`<sdk-install-dir>/plugins`	Intel® Media SDK – SDK plug-ins: • VP8 Decode plug-in `libmfx_vp8d_hw64.so` • VP9 Decode plug-in `libmfx_vp9d_hw64.so` • HEVC Hardware Decode Plug-in `libmfx_hevcd_hw64.so` • HEVC Hardware Encode Plug-in `libmfx_hevce_hw64.so` • Configuration file `plugins.cfg`
`<sdk-install-dir>/opensource/mfx_dispatch`	Source code for the Intel® Media SDK – SDK Dispatcher
`<sdk-install-dir>/tools/tracer`	SDK Tracer – tool for dumping API level logging information

`<sdk-install-dir>/builder`
`<sdk-install-dir>/samples`
	Samples

Documentation

For the most up to date versions please refer the Intel® Media SDK documentation page.

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

- `<sdk-install-folder>/doc/mediasdk-man.pdf`

  "SDK Developer Reference" describes the Intel Media SDK API.

- `<sdk-install-folder>/doc/mediasdkusr-man.pdf`

  "SDK Developer Reference: 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`

  "SDK Developer Reference for JPEG*/Motion JPEG" describes SDK API for JPEG* processing.

- `<sdk-install-folder>/doc/mediasdkvp8-man.pdf`

  "SDK Developer Reference for VP8*" describes SDK extension to support VP8* video codec.

Known Limitations

This release is subject to the following known limitations:

- LookAhead BRC as high memory consumption feature can't be recommended for usage with memory limited platform configurations.
- 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
- System stability is limited, unexpected system crashes were seen. To improve it, please make sure RC6 power state is off in BIOS and kernel (i915.enable_rc6=0)
- With enabled bound check on final bitstream buffer during encoding SDK can return MFX_DEVICE_ERROR in case of failed condition when encoded bitstream bigger than buffer. This especially may happen during CQP mode encoding, as WA please decrease the requested quality.
- PROCAMP filter does not work in the same pipeline with other filters. This requires 2 pass filtering – 1st pass PROCAMP filtering, 2nd pass all other filtering
- JPEG/MJPEG decode to RGB32 surface: Both frame width and height must be greater than 128 and smaller the 8000 pixels, total frame size should be smaller than aperture size defined in BIOS settings;
- MPEG2 decoder SW fallback is disabled to improve robustness on corrupted streams
- Broken HEVC streams may cause temporary GPU hangs resulting in discontinuities during playback or possible pipeline termination with various error codes, such events are recorded in dmesg log
- Dynamic resolution change is not supported by the HEVC encoder, attempt to change the resolution may cause GFX hangs, such events are recorded in dmesg log; as a workaround, if the resolution change is required, application is suggested to perform encoder re-initialization
- Deinterlacing mode MFX_DEINTERLACING_ADVANCED_NOREF is not supported in this release, the library will perform fallback to MFX_DEINTERLACING_BOB, it is recommended to use MFX_DEINTERLACING_ADVANCED assuming the processed stream do not have the scene changes on odd frames
- Both decoding and encoding of the higher resolution videos (720p+) in synchronous mode may produce temporarily drops in frame-to-frame performance making the pipeline unusable for demanding low-latency tasks
- Sample_mondello has dependency from libcamhal-1.0.0-144.4ad4bff.x86_64.rpm package
- **API:**

  Note: API version of this drop is 1.22

  MediaSDK 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
    - 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::WinBRCMaxAvgKbps, WinBRCMaxAvgKbps
    - 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
    - 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
- Only the following features among those introduced in API 1.20 are supported:
  - `PicStruct::MFX_PICSTRUCT_FIELD_SINGLE`
  - `PicStruct::MFX_PICSTRUCT_FIELD_TOP`
  - `PicStruct::MFX_PICSTRUCT_FIELD_BOTTOM`
  - `PicStruct::MFX_PICSTRUCT_FIELD_PAIRED_PREV`
  - `PicStruct::MFX_PICSTRUCT_FIELD_PAIRED_NEXT`
- Only the following features among those introduced in API 1.21 are supported:
  - none

NOTE: Other options may not be 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.
- **H.265 encode:**
  - 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() (New).
  - Encoder doesn’t support Intra Refresh.
  - Encoder doesn’t support HRD conformance for BRC modes (CBR, VBR); is case of significant divergence between target and output bitrates HRD information present in the stream may be erroneous, it is recommended to adjust target bitrates to the output bitrates and re-encode the stream if the decoder will relay on the HRD info
  - 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).
  - 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). Encoder does not support dynamic change of bitrate w/o IDR insertion. 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
  - Only TargetUsage 1, 4, 7 are supported. Encoder may return MFX_ERR_DEVICE_FAILED if TargetUsages 2, 3, 5, 6 are passed as input.
- **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.
  - Decoder may produce corrupted output in case when it’s feed by whole frames equipped with COMPLETE_BITSTREAM flag.
  - Decoder may produce GPU hangs during decoding of corrupted content.
- **H.264 encode:**
  - Reporting of per-frame QP via mfxExtAVCEncodedFrameInfo isn’t supported by Encoder (zero QPs are always returned) (new).
  - Encoder doesn’t support use of MaxFrameSize and MaxSliceSize together. If MaxSliceSize is set, MaxFrameSize is ignored (new).
  - 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. Reset returns MFX_ERR_INVALID_VIDEO_PARAM for LA BRC (new).
  - Encoder may produce duplicated frames in case of B-pyramid even if there is no risk of HRD underflow. It may significantly drop objective video quality of output.
  - 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
  - Encoder may produce significantly higher bitrate than requested (up to 1.6x) for very low bitrates (below 300 kbps for SD) and specific content (animation).
  - 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 on different platforms. Run to run difference doesn’t affect visual quality.
  - HRD violations are possible in specific scenarios (e.g. massive frame skipping).
  - Chroma artefacts are possible at very low bitrates (when QP is close to 51).

  - 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.
  - Encoder doesn't insert repeated (skipped) frames to avoid HRD underflow. HRD violations are possible for very complex content at very low bitrates.
  - 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.

- **MPEG-2 decode:**
  - 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.
  - Decoder has robustness issues with corruptions in headers (sequence header, picture header, extension headers) and may return unexpected statuses from MFXVideoDECODE_DecodeFrameAsync: MFX_ERR_UNDEFINED_BEHAVIOR, MFX_ERR_UNKNOWN or MFX_ERR_DEVICE_FAILED.

  - Sequence headers are skipped if resolution exceeds maximum supported values (2096x2096) or level/chroma are invalid. It affects MFXVideoDECODE_DecodeFrameAsync.
  - Decoder may cause GPU hangs, memory leaks and segmentation faults during decoding of corrupted content.

- **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
    - 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/MJPEG HW decoder supports only 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, multi-scan pictures only by SW fallback.
  - HW encoder supports only NV12, RGB32 input color formats with SW fallback in other cases.
  - Only up to two external quantization tables is allowed for encode
- **VPP:**
  - 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.
  - 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.
  - De-interlacing is supported for NV12, YUY2 formats only.

  - 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.
  - 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_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.
  - VPP rotation supports only NV12, RGB32 color formats.
  - Detail (sharpness) filter produce non bit-exact filtering comparing previous release
  - With new resize method enabled by default and providing better performance impact to quality is negative. Boxing and some color space conversion operations also affected by this change.
  - Color space conversion to RGB32 has worse quality in this drop
  - 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.

- **Misc:**
  - Software library is provided for demo/internal testing purposes only, it is not a product quality piece.
  - Due to specifics of GPU Copy implementation it is now required to close/destroy SDK associated resources (including VADisplay and frame surfaces) only after MFXClose call.
  - Encode quality may be different (non-bit exact) between CPU generations.
  - mfxExtThreadsParam::NumThread is not supported

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.

VP9 video codec is a high quality royalty free, open source codec deployed on millions of computers and devices worldwide. Implementations of VP9 CODECs, or VP9 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

Attributions

Safe C Library