Intel® Media Server Studio 2016 – Driver, SDK for Linux* Release Notes

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 2016 – Graphics Driver, version 16.4.4.47109
- Intel® Media Server Studio 2016 – SDK, version 6.0.16043361.361
- Intel® Media Server Studio 2016 – Samples are not a part of this package. The latest version of samples package (with all samples binaries and corresponding source code) could be downloaded from Intel® Media Server Studio 2016 Support.

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

What's New

Intel® Media Server Studio 2016 – SDK (hereinafter referred to as "SDK"):

Version 6.0.16043361.361:

- Bug fixes:
  - Performance:
    - Improved CPU utilization on multiple workloads running in parallel by fixing i915 KMD "Thundering Herd" problem (all the KMD waiters for the task completion were always waked up competing for the resources).
  - H.264 decoder:
    - Fixed decoder hang issue when decoder is initialized with the last found SPS header in the payload but the input bitstream position stays at the first found SPS.
    - Fixed hang issue when [SPS frame_width_in_mbs/frame_height_in_mbs * 16] exceeds max value for unsigned short [mfx.FrameInfo.Width/Height].
    - Fixed decoder failure while proceeding value of active references count in slice header, which exceed allowable range described in standard.
  - MPEG-2 decoder:
    - Robustness was improved. The following cases were fixed:
      - Width or height in sequence header is nulll.
      - Chroma value in sequence header is not valid.
      - Decoder hangs on bitstream with corrupted start code of sequence extension.
      - Decoder hangs on bitstream with corrupted picture structure type.
      - Segmentation faults and MFX_ERR_UNDEFINED_BEHAVIOR in decoder.
    - Synchronization on driver tasks was fixed.
    - Issue with CropH and CropW that were updated a frame later was fixed.
    - Resolution decrease in bitstream processing is fixed. The following issues were resolved:
      - Corruptions after resolution decrease change;
      - Corruptions when MFXVideoDECODE_Reset is called with resolutions that are smaller than the resolutions decoder is initialized with.
      - MFX_ERR_INCOMPATIBLE_VIDEO_PARAM after resolution decrease, now MFXVideoDECODE_DecodeFrameAsync returns MFX_WRN_VIDEO_PARAM_CHANGED on resolution decrease according to API.
    - Issue with incorrect picture structure reported from run to run was fixed.
    - Segmentation fault in GetVideoParam() calling to retrieve the SPS/PPS using an mfxExtBuffer of MFX_EXTBUFF_CODING_OPTION_SPSPPS was fixed.
    - Issue when MFXVideoDECODE_Init returned MFX_ERR_NONE on unsupported bit-streams with resolution higher than 2096x2096 was fixed.
  - H.264 encoder:

        - Robustness of MaxSliceSice feature was improved: fixed re-pack of slices when on difficult scene number of slices overflowed HW capabilities.
        - Quality drops at significant resolution increase were fixed. Following limitation isn't actual anymore "Significant increase of resolution by Reset function (CIF->FHD, or SD->FHD) may cause quality drop".
        - MBQP feature was enabled for Intel ® microarchitecture code name Broadwell. Following limitation isn't actual anymore "MBQP feature is supported only on Haswell".
        - Management of Decoded picture buffer was optimized for B-pyramid. This allows to optimally use DPB of size 4 for interlaced 7 B-frames B-pyramid. Following limitation isn't actual anymore "Encoder uses non-optimal reference pictures for 7 B-frame pyramid for interlaced coding with resolutions 1080i and higher".
    - VPP:
        - Fixed very rare floating visual corruption (~1/100K runs) on video pre-processing filters and transcoding from mpeg2/avc to mpeg2/avc on Broadwell.
        - Implemented VPP Reset functionality for denoise, detail, procamp, resolution change filters.
    - MJPEG decoder:
        - Fixed memory leak.
    - MJPEG encoder:
        - Fixed memory leak.
    - MPEG-2 encoder
        - Corrected error reporting when unsupported High Profile requested by user. Current version of HW MPEG-2 Encoder support up to MP@HL.
        - Fixed an error specific to handling resolution 194x232.
        - Fixed the last picture GOP type selection in case of strict GOP.
        - Fixed GOP timecode insertion in SPS header.
        - Added minor robustness checks for invalid opaque surfaces.

Version 6.0.16043321.321:

- H.264 decode
    - Extended robustness testing didn't show any problems with corrupted content
    - Calling of GetVideoParam() with an mfxExtBuffer of MFX_EXTBUFF_CODING_OPTION_SPSPPS may cause a segmentation fault. Not a real bug fix, it was wrong documented in previous release notes
- Mpeg 2 decode
    - Decoding robustness was seriously improved after disabling SW fallback, and fixes in parsing picture headers
    - Sequence headers are skipped if resolution exceeds maximum supported values (2096x2096) or level is invalid. It affects DecodeHeader and DecodeFrameAsync behavior. The changes happened in 16.4.3
- VPP
    - Fixed issue with not updated picstruct in output surface for ADI30->60.
    - Wrong documented in 16.4.3 limitations: "VPP Detail filter supports only progressive input." VPP Detail supports both interlaced and progressive content
- AVC encoder
    - Following BRC methods became functional: HRD compliant Look Ahead BRC and Look Ahead BRC with sliding window control (video to video memory copy required for them was fixed).

Following limitation isn't actual anymore: "HRD compliant Look Ahead BRC (MFX_RATECONTROL_LA_HRD) and Look Ahead BRC with sliding window control (MFX_RATECONTROL_LA and WinBRCMaxAvgKbps and WinBRCSize) aren't functional"

- Resolving conflict between H264 level and frame_mbs_only_flag was fixed.

Following limitation isn't actual anymore: "When PicStruct is set to MFX_PICSTRUCT_UNKNOWN at initialization, AVC encoder may produce streams with frame_mbs_only_flag = 0 and level lower than 2.1 or higher than 4.1. This is a regression compared to Intel Media Server Studio 2015 R6".

- Insertion of emulation prevention 0x03 bytes to slice header was fixed.

Following limitation isn't actual anymore: "Encoder may not insert emulation prevention 0x03 bytes to slice header".

Version 6.0.16043300.300:

- SDK API changed to 1.17
- Using MediaSDK shared object is allowed from dynamic shared object (please see in readme-dispatcher document)
- H.264 encoder:
  - Support of Intra Refresh for Temporal Scalability was added
  - Possibility to set HRD parameters for CQP BRC mode was added
  - Control over deblocking (disable_deblocking_filter_idc values 0,1,2) is enabled for H264 encoder
  - Control over trellis quantization for P and B frame types was enabled
- VPP:
  - Improved quality of VPP resize for interlaced streams
  - Enabled support of VPP PROCAMP (brightness, contrast, hue, saturation) filter (preview)
  - Introduced new Advanced deinterlacing filter mode without using reference frames
  - VPP interpolated FRC was deprecated and replaced by simple arbitrary FRC
- VP8 hybrid encode was deprecated and excluded from this drop
- Bug fixes:
  - H.264 decoder:
    - Improved robustness of decoder, fixed all known segment faults and incorrect reporting of error codes in decoding corrupted streams
    - Fixed top/bottom field SEI data reporting in decoded output
  - H.264 encoder:
    - Fixed Intra-only coding for B-fields which were reordered over IDR
    - Fixed VQ qlitch with using closed GOP structure
    - Reference list control was optimized for interlaced B-pyramid case
  - VPP:
    - Regression with VPP fields copy filter was fixed
    - Advanced Deinterlace filter: Improved quality of deinterlacing for scrolling text and blurred output for 25i->50p case
    - VPP Composition: Fixed issue with setting background color
    - Enabled support of GPU copy for pipelines with VPP Composition
  - MPEG-2 decoder: Fixed segmentation faults by decoding corrupted streams
  - MJPEG decoder: Fixed memory leak issue in MJPEG HW decoder and improved memory consumption
  - MJPEG encoder: Support of OPAQ memory of HW MJPEG decoder/encoder was implemented.

    - Fixed GPU hang in multithread encode test usage cases
    - Fixed mfxtracer hang after MFXInitEx

Version 6.0.16043166.166

- SDK API version increased to 1.16 to adopt new enhancements.
- Support of dynamic SDK calls from dynamic library implemented in dispatcher
- Added control to turn on/off GPU copy (for frame data copy between video and system memory)
- Frame/surface corruption detection and reporting for new CPU models implemented
- Support of opaque memory type in VP8 GPU-accelerated encoder added
- Enabled 8K x 8K HW support in MJPEG decoder
- Bug fixes:
    - MPEG-2 encoder: fixed issue with corruption when encoder initialized with unknown picture structure
    - MPEG-2 and H.264 (AVC) decoders: fixed failures with certain corrupted streams, improved decoder robustness
    - VP8 GPU-accelerated encoder: fixed visual corruption issue with big bitrates
    - VPP Resize: quality improved for interlaced content
    - VPP Advanced Deinterlace: performance improved
    - VPP Composition: GPU hang with N:1 transcoding fixed
    - VPP Interpolated FRC: fixed issue with end of stream processing

Version 6.0.16043138.138

- Bug fixes
    - MPEG2 encoder: fixed issue with insertion of VideoSignalInfo header at wrong position; improved encoder robustness.
    - H.264 encoder: fixed MBQP for interlaced encode; H.264 LA was fixed for default GOPRefDist in the case of no B-pyramid.
    - MPEG2 decoder: fixed issue of dropping a frame when aspect ratio changed in the stream.
    - H.264 decoder: fixed issue with SEI order when using GetPayload() function.
    - VPP: fixed issue in VPP impacting ability to enable Deinterlacing and Denoising filters. VPP FieldCopy filter issue (MFX_WRN_FILTER_SKIPPED at Init) was fixed.
    - Performance regression of VPP ADI was fixed.
    - Opaque memory handling issues in pipelines with VC1 decoder and plugins (VP8, HEVC) were fixed.
    - Memory leaks in GPU copy and MJPEG decoder were fixed.
    - Samples: incorrect handling of surfaces for ADI 30i->60p was fixed in sample_vpp. OpenCL™ rotation plug-in sample regression fixed.
    - Corruption of X11 Gnome* terminal was fixed.
- Enabled support of Interpolated FRC in sample_vpp.
- Improved latency due to new synchronization scheme for major components by using "spin free" blocking call of libva.
- Support for ftrace API to improve Intel® VTune™ Amplifier XE granularity and provide information on pipeline components.
- New version of metrics_monitor tool supporting new CPU architecture was included into release.

Version 6.0.16042112.112 (This is Beta release)

- SDK API version changed to 1.15 to adopt new enhancements.
- Enabled support of CPU models with former codename "Broadwell".

- HW accelerated decoding of VP8 streams by decoder plugin was added.
- Pure SW library libmfx64sw.so was added.
- MPEG2 encoder. Fixed the issue with correct placement of SignalInfo header.
- MPEG2 encoder. Support of Skip Frames encoding (CQP mode) was implemented
- MPEG2 encoder. Fixed MBQP issue.
- Small performance/latency improvements for 1,2 sessions by tuned synchronization
- New threading control API and implementation are introduced in this drop
- Issue with 4K LA encoding was fixed

Version 5.0.16043100.100

- SDK API version changed to 1.14.
- Implemented support of MPEG2 MBQP
- Implemented HRD compliancy for CQP mode of H.264 encoder
- Implemented Frame Type report for H.264 decoded surface
- Improved memory consumption to increase transcoding capacity
- New SDK API level tracer was implemented (Preview)
- Fixed Intra-coded P and B fields for B-pyramid in H.264 encoder
- Fixed panic mode for H.264 LA BRC
- Fixed YV12 color conversion for MJPEG SW encoder
- Fixed VPP Interpolated FRC
- Fixed regression in VPP ADI with default filter setting for 30i->60p
- Enabled VPP sharpness (detail) filter

Version 5.0.1604374.74 Hot Fix

- Hot fix for SDK for enabling back default deinterlacing when VPP received interlaced source and progressive output without configuration of deinterlacing in extended buffer.

Version 5.0.1604373.73

- Multi-threading issue for Intel® Media Server Studio 2015 – HEVC GPU Accelerated Encode plugin was fixed.

Version 5.0.1604368.68

- New features:
    - SDK API changed to version 1.13
    - Performance of H.264 Encode Configurable Slice Size feature for Video Conferencing scenarios was improved.
    - GPU accelerated copy for CPU-GPU memory interexchange was enabled.
    - VPP Advanced De-interlace algorithm was enhanced to use motion information which leads to improved quality of de-interlace.
- The following MPEG2 Decode issues were fixed:
    - MPEG2 Decode may fall in infinite MFX_WRN_DEVICE_BUSY loop on certain severely corrupted streams. Now it will return MFX_ERR_UNDEFINED_BEHAVIOR and application is expected to close and re-initialize the decode component.
    - MPEG2 Decode doesn’t report downsize resolution change. Fix: now reports via MFX_WRN_VIDEO_PARAM_CHANGED as per API specification.
    - Timestamps pass-through from input to output doesn’t work properly in COMPLETE_FRAME mode.

- The following VPP Composition and Alpha Blending issues were fixed:
    - VPP composition uses the first stream as background in destination surface. Fix: now customizable background color is used as per API specification. **Note: a new issue appeared as a side effect of this fix, please check Known Limitations section for details.**
- The following H.264 encoder improvements:
    - Several quality improvements modes for H.264 encoder were added
    - Bitrate control MFX_RATECONTROL_LA with sliding window has been improved
    - Buffering Period SEI insertion control for AVC Encode was implemented.

Version 5.0.1604246.46

- This is Pre-Release version.
- New features:
    - Support of Render Nodes was added.
    - VP8 Hybrid Encoder was added.

Version 5.0.1604227.27

- This is Pre-Release version.
- OS and hardware support:
    - Ubuntu* 12.04 LTS and SUSE* Linux* Enterprise Server 11 support discontinued.
    - CentOS* 7.0, SUSE* Linux* Enterprise Server 12 support was added.
    - Set of patches for open source components on top of 3.14.5 kernel suitable for other variants of Linux OSes
    - This release is for Intel® 4th Generation Core/Xeon E3-128X(L) v3 processors
- New features:
    - New packages structure including per-component packages suitable for using default package managers
    - OpenCL™ 1.2 for GPU runtime and development packages, please see details in respective release notes in intel-opencl-1.2-16.4.tar.gz.
    - JPEG/MJPEG HW Decoder was enabled.
    - H.264 Encode Configurable Slice Size feature for Video Conferencing scenarios was added.
    - VPP Field Copy filter was added.

Features

Intel® Media Server Studio 2016 – SDK included in this package implements SDK API 1.17 and contains the following components:

Component	Supported features	Limitations
H.264 decoder	Supported Profiles: • Baseline • Main • High	Maximum supported resolution: 4096x2304
H.264 encoder	Supported Profiles: • Baseline • Main • High Supported BRC methods: • Constant QP (CQP) • Constant Bit Rate (CBR) • Variable Bit Rate (VBR) • Look Ahead (LA)	Maximum supported resolution: 4096x4096
MPEG-2 decoder	Supported Profiles: • Simple • Main • High	Maximum supported resolution: 2096x2096
MPEG-2 encoder	Supported Profiles: • Simple • Main Supported BRC methods: • Constant QP (CQP) • Constant Bit Rate (CBR) • Variable Bit Rate (VBR)	Maximum supported resolution: 1920x1088
VC1 decoder	Supported Profiles: • Simple • Main • Advanced	Maximum supported resolution: 1920x1088
MJPEG encoder (SW only)	Supported Profiles: • Baseline mode, 8bit	Maximum supported resolution: per ISO/IEC 14495-1 and system memory limitations

MJPEG decoder	Supported Profiles: • Baseline mode, 8bit	Maximum supported resolution: per ISO/IEC 14495-1 and system memory limitations
VP8 decoder	Supported Profiles: • Version 0 mode, 8bit	Maximum supported resolution: 1920x1088
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	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 2016 - SDK Reference Manual for details "`<sdk-install-folder>/doc/mediasdk-man.pdf`"

System Requirements

Hardware

The following Intel platforms with processor graphics are supported:

- Intel® Xeon® Processor E3-128x v4
- 5th Generation Intel® Core™ Processors with Intel Iris™ Pro Graphics, Intel Iris™ Graphics or Intel HD Graphics (5500, 6000, 6100, 6200).
- Intel® Xeon® Processor E3-128x v3
- 4th Generation Intel® Core™ Processors with Intel Iris™ Pro Graphics, Intel Iris Graphics or Intel HD Graphics 4200+ Series (chipset compatibility is usually not an issue for Core™ processors.)

- Note: chipset must have processor graphics enabled; make sure to check the datasheet.
    - Only C226 Server Chipset is supported
    - Having a C226 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

The following Intel platforms are supported for SW (CPU) only processing:

- Intel® Xeon® Processor E5 v3 and v4

Note: SW only processing includes HEVC decode & encode (part of Media Server Studio – Professional Edition), selected Video Pre-Processing (SCS, Scaling, DI), Virtualization (KVM*, Xen*)

The following Intel platforms are not supported:

- Intel® Core processors earlier than 4th Generation, Intel® Celeron®, Intel® Pentium® and Intel® Atom™

Software

- CentOS 7.1 (1503) of 64-bit architecture its default 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.**
- Xf86-video-intel driver (needed only for if local rendering is required with the LibVA X11 backed). Recommended version: 2.20.10, http://cgit.freedesktop.org/xorg/driver/xf86-video-intel/snapshot/xf86-video-intel-2.20.10.tar.gz
- Generic OS install uses kernel 3.14.5 from www.kernel.org.

Package Contents

Intel® Media Server Studio 2016 – Driver, SDK for Linux* package includes the following components, where `<id>` is Graphics Driver version:

Component	Description

`CentOS/intel-opencl-1.2-<id>.el7.x86_64.rpm`
`CentOS/intel-opencl-1.2-devel-<id>.el7.x86_64.rpm`
`CentOS/intel-opencl-1.2-<id>.tar.gz`
	Intel® Media Server Studio 2016 OpenCL™ packages

CentOS/intel-linux-media-<id>.el7.x86_64.rpm	Intel® Media Server Studio 2016 – Driver & SDK runtime package.
CentOS/intel-linux-media-devel-<id>.el7.x86_64.rpm	Intel® Media Server Studio 2016 – Driver & SDK development package.
CentOS/libdrm*-<id>.el7.x86_64.rpm CentOS/drm-utils*-<id>.el7.x86_64.rpm	Direct Rendering Manager runtime library runtime, development, etc.
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 2016 – Driver & SDK installation scripts.
CentOS /intel-linux-media-samples-<id>.el7.x86_64.rpm	Intel® Media Server Studio 2016 – Samples package (binaries only).
Generic/intel-linux-media-ocl_generic_<id>_64bit.tar.gz	Intel® Media Server Studio 2016 – Driver & SDK 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 EULA.pdf redist.txt site_license_materials.txt third_party_programs.txt	Intel® Media Server Studio 2016 – SDK documentation: this file, Getting Started Guide, EULA, EULA's accompanying files.

Installation Folders

Intel® Media Server Studio 2016 – 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 2016 – SDK Dynamic Library, hardware implementation libmfxhw64-p.so.* software implementation libmfxsw64-p.so.*
<sdk-install-dir>/doc	Intel® Media Server Studio 2016 – SDK documentation
<sdk-install-dir>/include	External Intel® Media Server Studio 2016 – 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_x64	Intel® Media Server Studio 2016 – SDK Static Dispatcher Library: libmfx.a
<sdk-install-dir>/plugins	Intel® Media Server Studio 2016 – SDK plug-ins: • Advanced AVC Encode plug-in (implements 1:N Look Ahead optimization)

	libmfx_h264la_hw64.so • VP8 Decode plug-in libmfx_vp8d_hw64.so • Configuration file plugins.cfg
<sdk-install-dir>/opensource/mfx_dispatch	Source code for the Intel® Media Server Studio 2016 – SDK Dispatcher
<sdk-install-dir>/tools/drmserver	Intel® Media Server Studio - SDK DRM Authentication Server
<sdk-install-dir>/tools/metrics_monitor	Metrics Monitor – tool for monitoring GPU metrics
<sdk-install-dir>/tools/tracer	SDK Tracer – tool for dumping API level logging information

Intel® Media Server Studio 2016 – Graphics Driver installs in the following locations:

Component	Description
<sdk-install-dir>/lib64	Intel® Media Server Studio 2016 – Graphics Driver, SDK
<sdk-install-dir>/opensource/libdrm	Source code for Direct Rendering Manager runtime library
<sdk-install-dir>/opensource/libva	Source code for Video Acceleration (VA) API
<sdk-install-dir>/opensource/patches/kmd	Source code of Intel® Media Server Studio 2016 – Graphics Driver, Kernel Mode Driver (KMD)
/usr/include	Direct Rendering Manager runtime library, Video Acceleration (VA) API includes
/usr/lib64	Direct Rendering Manager runtime library, Video Acceleration (VA) API libraries
/usr/bin	Direct Rendering Manager runtime library, Video Acceleration (VA) API utilities.

OpenCL™ Driver installs in the following locations:

Component	Description
/opt/intel/opencl/include/CL	OpenCL™ Driver includes
/opt/intel/opencl	OpenCL™ Driver libraries
/etc/OpenCL/vendors	Configuration for Khronos OpenCL ICD library

You could find more information about OpenCL™ Driver in corresponding Release Notes in intel-opencl-1.2-16.4.tar.gz.

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 2016 - 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

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

- Intel® Xeon® Processor E3-128x v4 and 5th Generation Intel Core™ Processors will be referred to as "Broadwell"
- Intel® Xeon® Processor E3-128x v3/4th Generation Intel Core™ Processors will be referred to as "Haswell"

This release is subject to the following known limitations:

- **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`
    - `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

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.
  - Performance of N:N and 1:N multi-transcoding may not be optimal due to absence of Kernel Mode Driver scheduler required for the out-of-order execution of the tasks with satisfied dependencies. This effect may differ depending on the components used in the pipeline: the more components producing the tasks for the different GPU engines – the bigger the effect. **As a mitigation option**, it is recommended to design pipeline in a way for it to produce serialized tasks. One of the major things – consider to use AsyncDepth=1.
  - Performance of the pipeline containing 2 VPP components in 2 SDK sessions in a way that 1st session contains decoder and VPP, the 2nd session – VPP and encoder may be better than single-session pipeline with single VPP combining all filters. Considering such VPP split it is recommended to perform Dinterlacing, Detail Enhancement and Denoising in 1st VPP component (attached to decoder), and Scaling, FrameRate conversion in the 2nd VPP component (attached to encoder).
  - Performance on some use cases running multiple parallel workloads may be unstable in the sense that the same scenario executed few times in a row may achieve significantly better performance (noticed as up to 20%) compared to usual. This is mostly noticed on multiprocess use cases on Broadwell but described behavior seldom (and randomly) appears on Broadwell multisession use cases and on Haswell multisession/multiprocess use cases.
- **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 hang-up while proceeding corrupted streams with value of first_mb_in_slice in Slice header exceeds the number of 1079840751.

- Decoder may produce memory leak in function [MFXVideoDECODE_DecodeHeader] for streams containing corruptions in following elements of PPS header: pic_parameter_set_id, seq_parameter_set_id, slice groups syntax, num_ref_idx_lX_default_active_minus1. Leak happens if [MFXVideoDECODE_DecodeHeader] function doesn't find any PPS w/o listed corruptions between 2 SPS headers in the stream. Usually this function is being used few times per session at initialization or after drastic parameters change (resolution) or corruption recovery.

- **H.264 encode:**
  - 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 Broadwell. Run to run difference doesn't affect visual quality.
  - Encoder may produce non-bit exact streams for interlaced coding only on Haswell. 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. Encoding of specific (complex) content with huge bitrate (which makes encoder insert many PCM MBs) may cause a GPU hang on Haswell.
  - Encoder doesn't process recovery after a GPU hang correctly. It returns MFX_ERR_NONE from SyncOperation with DataLength = 0 in corresponding mfxBitstream. If application continues encoding procedure as usual, MFX_ERR_UNKNOWN may be returned by SyncOperation at certain iteration.
  - On Broadwell encoder doesn't insert repeated (skipped) frames to avoid HRD underflow. On Haswell it does, which gives better HRD robustness. For Broadwell HRD violations are possible for very complex content at very low bitrates.
  - Usual Look Ahead BRC may generate non HRD-compliant streams.
  - Reset function isn't supported for LookAhead BRC modes. Reset function doesn't return an error when called together with LA BRC. Result of such Reset call is undefined.

- B-pyramid isn't supported together with HRD compliant Look Ahead BRC (MFX_RATECONTROL_LA_HRD) and Look Ahead BRC with sliding window control (MFX_RATECONTROL_LA and WinBRCMaxAvgKbps and WinBRCSize).
- Careful memory/resource planning is needed when using Look Ahead BRC due to storage of pre-analyzed frames. 1:N and N:N transcoding use cases are especially demanding for memory.
- When external Look Ahead BRC is used in transcoding pipeline which includes Frame Rate conversion (FRC), FRC in the pipeline should take place before external Look Ahead. Otherwise encoder may return error MFX_ERR_UNDEFINED_BEHAVIOR from EncodeFrameAsync calls.
- On Broadwell, encoding mode with limitation on maximum slice (MaxSliceSize) may produce very big latency (hundreds milliseconds) for the first few frames in the stream.
- Trellis option can be enabled only on lower target usages, on some of those it is enabled by default but can be switched off. Exact implementation details are hidden and may change with time and between platforms. Use of `Query` function to retrieve actual support is strongly recommended.
- 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.
- Dynamic change of MaxFrameSize isn't supported. Encoder keep using initial value of MaxFrameSize after Reset() or per-frame update.

- **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.
  - The MPEG-2 decoder may keep Corrupted flag as 0 in case of minor corruption in macroblock bitstream data.

- mfxDecodeStat.NumFrame that is returned from GetDecodeStat function is less by 1 than actual count of decoded frames if GetDecodeStat is called after buffered frames are returned.

- **MPEG-2 encode:**
  - Encoder may produce non-bit exact streams. Run to run difference doesn't affect visual quality.
  - 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 bufer 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.
  - MBQP feature is supported only on Haswell.
  - 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.
- **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.
    - Embedded picture rotation is not supported yet.
- **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.
    - Once enabled at the Init stage VPP de-interlacing for BOB mode is not disabled automatically if application provides input frames with picstruct set to MFX_PICSTRUCT_PROGRESSIVE.
    - 1st frame is doubled during 30i->60p de-interlacing for MFX_DEINTERLACING_BOB and MFX_DEINTERLACING_ADVANCED_NOREF modes
    - MFXVideoVPP_GetVideoParam does not update values in attached extended buffers, except VPP_DO_USE
    - MFXVideoVPP_Reset could return error if additional memory allocation is required based on provided video parameters (changed type of color or frame rate conversion for example). The application should close VPP component and then re-initialize it in this case. See MFXVideoVPP_Reset in SDK API Reference Manual for more details about possible return statuses.
    - Multiple RGB4 surfaces with PixelAlphaEnable enabled could give poor visual quality after composition on the same area with overlapping.
    - In case of YV12->RGB4 color space conversion output RGB4 surface has alpha channel filed with random values instead of 0xFF.

- 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.

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

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