

Intel® Media Server Studio 2017 R3 – Driver, SDK for Windows Server* Release Notes

[Overview](#)

[What's New](#)

[System Requirements](#)

[Installation Folders](#)

[Documentation](#)

[Known Limitations](#)

[Other Limitations](#)

[Legal Information](#)

Overview

The **Intel® Media Server Studio – Driver, SDK for Windows Server*** 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 Windows 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, versions: 15.40.4549 for Intel® Xeon® E3-1200 v4 and 5th Generation Intel® Core™, 15.45.4639 for Intel® Xeon® E3-1500 v5 and 6th Generation Intel® Core™
- Intel® Media Server Studio 2017 – Software Development Kit, version 7.0.0000.487
- Intel® Media Server Studio 2017 – Screen Capture plug-in, version 1.23.3.164
- Intel® Media Server Studio 2017 – Advanced AVC Encode plug-in, version 1.23.6.164
- Intel® Media Server Studio 2017 – Samples are not a part of this package, but they can be accessed through [Intel\(R\) Media Server Studio Support](#).

This document covers product features, system requirements and known limitations.

What's New

The **Intel® Media Server Studio – SDK for Windows Server*** (further referred to as the SDK) introduces API version 1.23.

Following fixes, improvements and features were added:

- API updates comparing to 1.20:
 - mfxcamera.h: mfxExtCamTotalColorControl structure added to control YUV->YUV conversion filter.
 - mfxcamera.h: mfxExtCamCscYuvRgb structure added to control conversion from RGB to YUV color format.
 - mfxplugin++.h: QueryPlatform function added to get current platform codename in application via plugin.
 - mfxstructures.h: MFX_REFRESH_NO, MFX_REFRESH_VERTICAL, MFX_REFRESH_HORIZONTAL, MFX_REFRESH_SLICE added for intra refresh support.
 - mfxstructures.h: mfxExtCodingOption3::RepartitionCheckEnable added to control AVC encoder attempts to predict from small partitions.
 - mfxstructures.h: mfxExtCodingOption3::AdaptiveMaxFrameSize added to control max size of P and B frames.
 - mfxstructures.h: mfxExtCodingOption3::EnableMBForceIntra and mfxExtMBForceIntra structure added for force specified macroblocks to be encoded as intra.
 - mfxstructures.h: mfxExtCodingOption3::LowDelayBRC added to specifies frame size tolerance for rate controls: VBR, QVBR, VCM.
 - mfxstructures.h: mfxExtCodingOption3::BRCPanicMode added to control panic mode in MPEG2 encoders.
 - mfxstructures.h: mfxExtEncoderROI::ROI Mode, mfxExtEncoderROI::ROI::DeltaQP and MFX_ROI_MODE_PRIORITY, MFX_ROI_MODE_QP_DELTA added to control QP adjustment for ROIs.
 - mfxstructures.h: mfxExtDecVideoProcessing structure added to control decoder to resize output frames via fixed function resize engine.

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

- Common
 - Added Control Flow Guard for driver version 15.45.4639.
- HW AVC Encode:
 - Improved dynamic intra refresh for driver version 15.45.4639.
 - Slice based intra refresh added for driver version 15.45.4639.

- Low delay BRC and sliding window BRC support added for driver version 15.45.4639.
- Max frame size I/P support added for driver version 15.45.4639.
- Explicit weighted P prediction and weighted B prediction support added for driver version 15.45.4639.
- Fade detection support added as preview for driver version 15.45.4639.
- Control RGB input color space via mfxExtVideoSignalInfo added for driver version 15.45.4639.
- Added support AdaptiveMaxFrameSize for driver version 15.45.4639.
- Forced intra based on ROIs support added for driver version 15.45.4639.
- Scene change quality improvement added for driver version 15.45.4639.
- HW HEVC Encode:
 - Max frame size BRC constraints added for driver version 15.45.4639.
- HW VPP:
 - Fixed CM resources memory leak.
- Media RAW Accelerator Plug-in:
 - Added support total color conversion for driver version 15.45.4639.
 - Added support control color space conversion from RGB to YUV for driver version 15.45.4639.

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

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

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

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

Please see the Screen Capture plug-in Reference Manual for details

"`<install-folder>\doc\mediasdkscreen-cap-man.pdf`"

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

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

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

System Requirements

Hardware

The following Intel platforms with processor graphics are supported:

- Intel® Xeon® Processor E3-1200 v5, Intel® Xeon® Processor E3-1500 v5 series.
- 6th Generation Intel Core™ Processors with Intel Iris™ Pro Graphics, Intel Iris™ Graphics or Intel HD Graphics (580, 540, 530).
- Intel® Xeon® Processor E3-1200 v4 series
- 5th Generation Intel Core™ Processors with Intel Iris™ Pro Graphics, Intel Iris™ Graphics or Intel HD Graphics (5500, 6000, 6100, 6200).
- Note: chipset must have processor graphics enabled; make sure to check the datasheet.
 - Only C226 Server Chipset is supported for Xeon E3 v4 platforms.
 - Only C236 Server Chipset is supported for Xeon E3 v5 platforms.
 - 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>

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

- Intel 64 architecture processor supporting the Intel® Streaming SIMD Extensions 3 (SSE3).

Software

- Microsoft Windows Server* 2012 R2, 64-bit Microsoft Windows* 8.1 for server/embedded usage, Microsoft Windows* 10 and updates (OS builds 10240 – codename Threshold 1; 10586 – codename Threshold 2). 64-bit Microsoft Windows* 7 for development purpose only.
- Microsoft Windows Server* 2016 (driver implements WDDM2.0 only). Graphics driver supports the following Operating Systems:
 - Microsoft Windows Server* 2012 R2 64-bit;
 - Microsoft Windows Server* 2012 64-bit;
 - Microsoft Windows Server* 2008 R2 64-bit.

- Microsoft Visual C++* 2005 with Service Pack 1, or later version of Microsoft Visual C++.

Note: Other combinations of Microsoft Windows Server* 2012/2016 and Intel Core™/Xeon based platforms may function. But please be aware that such combinations are neither validated nor supported server platforms by Intel Media Server Studio – SDK for Windows Server*. These software drivers are generic versions and can be used for general purposes. However, computer original equipment manufacturers (OEMs) may have altered the features, incorporated customizations, or made other changes to the software or software packaging they provide. To avoid any potential installation incompatibilities on your OEM system, Intel recommends that you check with your OEM and use the software provided by your system manufacturer. Intel and the computer original equipment manufacturer (OEM) may not provide technical support for some or all issues that could arise from the usage of this generic version of software drivers.

Installation Folders

Intel® Media Server Studio 2017– SDK installs under C:\Program Files\Intel\Intel(R) Media Server Studio 2017 R3\ – this is referenced as <sdk-install-dir> in the remainder of this document.

<sdk-install-dir>\Documentation for Essentials Edition\media_server_studio_sdk_release_notes.pdf	Intel® Media Server Studio 2017 – Driver, SDK for Windows Server* Release Notes (this file)
<sdk-install-dir>\Documentation for Essentials Edition\media_server_studio_essentials_release_notes.pdf	Intel® Media Server Studio 2017 – Essentials Edition for Windows* Server Release Notes
<sdk-install-dir>\Software Development Kit\bin\x64	Intel® Media Server Studio 2017 – SDK Dynamic Library, software implementation: libmfxsw64.dll for Intel® 64 architecture Note: Hardware implementation of Intel® Media Server Studio 2017 – SDK Dynamic Library libmfxhw64.dll is packed and installed with Intel® Media Server Studio 2017– Graphics Driver (default location is C:\Program Files\Intel\Media SDK)
<sdk-install-dir>\Software Development Kit\bin\x64\22d62c07e672408fbb4cc2	Intel® Media Server Studio 2017 – Screen Capture plug-in:

0ed7a053e4	• Screen Capture plug-in mfxplugin64_screen_capture.dll • Configuration file plugin.cfg
<sdk-install-dir>\Software Development Kit\bin\x64\588f1185d47b42968dea377bb5d0dcb4	Intel® Media Server Studio 2017 – Advanced AVC Encode plug-in: • Advanced AVC Encode plug-in (implements 1:N Look Ahead optimization) mfxplugin64_h264la_hw.dll • Configuration file plugin.cfg
<sdk-install-dir>\ Software Development Kit\doc	Intel® Media Server Studio 2017 – SDK documentation
<sdk-install-dir>\ Software Development Kit\include	External Intel® Media Server Studio 2017 – SDK headers: • Type definitions in mfxdefs.h • Structure for video definitions in mfxstructures.h • Structure for audio definitions in mfxastructures.h • Common structures definition in mfxcommon.h • Functions for work with sessions definitions in mfxsession.h • Function for video definitions in C in mfxvideo.h • C++ wrapper of the SDK functions for video in mfxvideo++.h • Function for audio definitions in C in mfxaudio.h • C++ wrapper of the SDK functions for audio in mfxaudio++.h • Extensions for Multi-view Video Coding options mfxmvc.h • Extensions for User-Defined Functions

	mfplugin.h • C++ wrapper for User-Defined Functions <code>mfplugin++.h</code> • Extensions for ENC Video coding options <code>mfenc.h</code> • Extensions for PAK Video coding options <code>mfpack.h</code> • Extensions for JPEG*/Motion JPEG Video coding options <code>mfjpeg.h</code> • Extensions for LookAhead Video coding options <code>mfsla.h</code> • Extensions for RAW Accelerator <code>mfxcamera.h</code> • Extensions for VP8 Video coding options <code>mfvp8.h</code> • <code>mfstructures.h</code>
<sdk-install-dir>\Software Development Kit\lib\x64	Intel® Media Server Studio 2017 – SDK Static Dispatcher Libraries <code>libmfx.lib</code> and <code>libmfx_vs2015.lib</code> for Microsoft* Visual Studio 2015
<sdk-install-dir>\Software Development Kit\igfx_s3dcontrol	Intel® Media Server Studio 2017 – SDK Stereoscopic 3D API for Intel Iris, Intel® Iris Pro and HD Graphics hardware, includes: • S3D API definitions <code>\include\igfx_s3dcontrol.h</code> • Static S3D Control Library <code>\lib\x64\igfx_s3dcontrol.lib</code> • Displaying S3D with Intel® HD Graphics Developers Guide
<sdk-install-dir>\Software Development Kit\tools	Contains the following tools in binary form: • SDK Tracer in folder <code>mediasdk_tracer</code>. This utility performs runtime recording of Intel SDK API calls and parameters to a log file.

	• SDK System Analyzer in folder mediasdk_sys_analyzer. This utility analyzes the system and reports back the SDK related capabilities, graphics driver and components status.
<sdk-install-dir>\Software Development Kit\opensource\mfx_dispatch	Source code of Intel® Media Server Studio 2017 – SDK Dispatcher
<sdk-install-dir>\Software Development Kit\samples	The link for downloading Samples packages.

Documentation

You can find more information on how to use Intel® Media Server Studio 2017 – SDK in the following documentation:

- <sdk-install-dir>\Software Development Kit\doc\mediasdk-man.pdf
“Intel® Media Server Studio – SDK Developer Reference” describes the SDK API.
- <sdk-install-dir>\Software Development Kit\doc\mediasdkusr-man.pdf
“Intel® Media Server Studio – 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-dir>\Software Development Kit\doc\mediasdkjpeg-man.pdf
“Intel® Media Server Studio – SDK Developer Reference for JPEG*/Motion JPEG” describes SDK API for JPEG* processing.
- <sdk-install-dir>\Software Development Kit\doc\mediasdkmvc-man.pdf
“Intel® Media Server Studio – SDK Developer Reference for Multi-view Video Coding” describes the SDK extension to support Multi-view Video Coding (MVC).
- <sdk-install-dir>\Software Development Kit\doc\mediasdk-distrib.pdf
“Intel® Media Server Studio – Media SDK Library Distribution and Dispatching Process”.
- <sdk-install-dir>\Software Development Kit\doc\mediasdkscreen-cap-man.pdf
“Intel® Media Server Studio – Screen Capture Decoder Developer Reference”
- <sdk-install-dir>\Software Development Kit\doc\media-raw-accelerator-man.pdf
“Intel® Media Server Studio – Media RAW Accelerator Developer Reference”

Known Limitations

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

API

- The following APIs are not supported by the software implementation of the SDK library:
 - `mfxExtEncoderCapability`, `mfxExtEncoderResetOption`, `mfxExtAVCEncodedFrameInfo`;
 - `MFX_RATECONTROL_LA`, `MFX_RATECONTROL_ICQ`, `MFX_RATECONTROL_LA_ICQ`, `MFX_RATECONTROL_VCM`, `MFX_RATECONTROL_QVBR`, `MFX_RATECONTROL_LA_HRD` and any options related to these BRC modes;
 - `mfxExtVPPComposite`, `mfxExtVPPDeinterlacing`;
 - `mfxExtAVCRefListCtrl::ApplyLongTermIdx`, `LongTermIdx`;
 - `mfxExtEncoderROI`;
 - `mfxExtCodingOption2::Trellis`, `MBBRC`, `ExtBRC`, `RepeatPPS`, `BRefType`, `AdaptiveI`, `AdaptiveB`, `LookAheadDS`;
 - `mfxInfoMFX::ICQQuality`;
 - `mfxEncodeCtrl::SkipFrame`;
 - `MFX_PLUGINID_HEVCE_HW`, `MFX_EXTBUFF_HEVC_PARAM`, `mfxExtHEVCParam`;
 - `CreateAccelerationDevice`;
 - `MFXInitEx`, `MFXDoWork`, `mfxInitParam`, `mfxExtThreadsParam`;
 - `mfxInfoMFX::LowPower`;
 - `MFX_EXTBUFF_DECODED_FRAME`, `mfxExtDecodedFrameInfo`, `MFX_EXTBUFF_TIME_CODE`, `mfxExtTimeCode`;
 - `mfxExtCodingOption3`;
 - `mfxExtPredWeightTable`;
 - `mfxExtDirtyRect`;
 - `mfxExtMovingRect`;
 - `mfxInitParam::GPUCopy`;
 - `mfxInfoMFX::MaxDecFrameBuffering`;
 - `mfxVideoParam::AllocId`;
 - `MFX_MEMTYPE_EXPORT_FRAME`;
 - `mfxExtCodingOptionVPS`;
 - `mfxExtVPPRotation`;
 - `mfxExtVPPSignalInfo`;
 - `mfxExtVPPMirroring`;
 - `MFXVideoCORE_QueryPlatform`;
 - `mfxPayload::CtrlFlags`;

- mfxFrameData::MemType;
 - mfxExtVPPScaling;
 - mfxExtVPPColorFill;
 - mfxExtEncodedSlicesInfo;
 - MFX_MEMTYPE_SHARED_RESOURCE;
 - mfxCoreInterface::QueryPlatform;
 - mfxExtSceneChange;
 - mfxExtVPPColorFill;
 - mfxExtHEVCRegion;
 - mfxExtVPPFieldProcessing;
 - MFX_DEINTERLACING_ADVANCED_SCD;
 - MFX_DEINTERLACING_FIELD_WEAVING;
 - MFX_REFRESH_NO, MFX_REFRESH_VERTICAL, MFX_REFRESH_HORIZONTAL, MFX_REFRESH_SLICE;
 - MFXCoreInterface::QueryPlatform;
 - mfxExtDecVideoProcessing;
 - mfxExtMBForceIntra;
- The below APIs are not supported by the hardware implementation of SDK Library in this release. For other APIs not mentioned in this list make sure to call `Query` functions to check the actual support on a particular platform as it may vary.
 - mfxExtCodingOption2::AdaptiveI, AdaptiveB, UseRawRef;
 - mfxExtAVCEncodedFrameInfo::MAD, BRCPanicMode, QP;
 - MFX_PLUGINID_VP8D_HW;
 - mfxExtChromaLocInfo;
 - mfxExtMBQP;
 - MFXInitEx, MFXDoWork, mfxInitParam, mfxExtThreadsParam;
 - mfxExtDirtyRect;
 - mfxExtMoveRect;
 - mfxExtCodingOption3::ScenarioInfo, ContentInfo, RepartitionCheckEnable EnableMBQP, DirectBiasAdjustment, GlobalMotionBiasAdjustment;
 - mfxExtCodingOptionVPS;
 - mfxExtEncodedSlicesInfo;
 - mfxExtSceneChange;
 - mfxExtHEVCRegion;
 - mfxExtVPPFieldProcessing;
 - MFX_PICSTRUCT_FIELD_SINGLE, MFX_PICSTRUCT_FIELD_TOP, MFX_PICSTRUCT_FIELD_BOTTOM, MFX_PICSTRUCT_FIELD_PAIRED_PREV, MFX_PICSTRUCT_FIELD_PAIRED_NEXT;

- o `mfxExtEncoderROI::ROIMode`.

Functional

Common/general limitations:

- The SDK dispatcher `libmfx.lib` is best used with a standard DLL entry point (as recommended by Microsoft*) when used in a DLL application such as a Microsoft DirectShow* filter. The DLL entry point setting can be found under the `Link > Advanced` compiler options. Non-standard entry points can be used, but are not recommended.
- Loading of SDK dynamic libraries `libmfxsw64.dll` and `libmfxhw64.dll` not through the dispatcher is unsafe.
- Using the software implementation of SDK in parallel with Intel® Threading Building Blocks could impact performance.
- The number of internal tasks in hardware implementation is limited to 1024. This imposes a related limitation on the number of SDK sessions which depends on the number of components in a session and the asynchronous depth of each component: each component (DECODE, ENCODE or VPP) requires one task for synchronous operation and N tasks for asynchronous operation with depth N.
- This release supports only 64-bit Microsoft* Windows* applications.
- Microsoft DirectX* 11.1 is the only supported acceleration infrastructure (due to headless mode requirement).
- On Windows 10 bitstreams produced by any encoder can be not bit exact from time to time, but no visible difference.
- Recommended to use system memory allocated with pointer alignment to 64, otherwise result can contain corruptions and sometimes pipeline can be broken.

JPEG encode, decode:

- The feature set of JPEG decoder/encoder is limited to the following:
 - o 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.
 - o No extended, lossless and hierarchical modes
 - no 12-bit samples;
 - no progressive;
 - no arithmetic coding;

- no 4 AC and 4 DC Huffman tables.
- JPEG decoder does not set `Corrupted` flag of `mfxFrameData` structure, and does not accept `MXF_BITSTREAM_EOS` as `DataFlag` of `mfxBitstream` structure.
- Software library implementation can break pipeline in stress multisession encoding and complex threading graphs scenarios when two or more encoders sharing the same `mfxFrameSurface1` object and memory handling implemented through `MemId`.
- Encoder with GPU copy may produce stream witch not bit to bit with stream encoded without GPU copy.

HEVC decode:

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

H.264 decode:

- H.264 decoder may consume more than 1 frame from the input bitstream and then propagate same timestamp to all of the consumed frames. If accurate time stamp handling is required the application has to make sure that it doesn't store more than one-frame wise data in the input bitstream.

MPEG-2 encode limitations:

- If the MPEG-2 Video encoder `mfxVideoParam::mfxInfoMFX::CodecProfile` is initialized to 0, then the stream will be encoded as `MXF_PROFILE_MPEG2_MAIN`. Additionally if the MPEG-2 Video encoder `mfxVideoParam::mfxInfoMFX::CodecLevel` is initialized to 0, then the stream will be encoded as `MXF_LEVEL_MPEG2_MAIN`.
- MPEG-2 encode may produce not bit to bit result from run to run on 5th generation Intel® Core and Intel® Xeon E3 v4. Run to run variation doesn't affect visual quality.
- `mfxExtCodingOption3::BRCPanicMode` not supported for windows.

HEVC encode limitations:

- Supports maximum resolution 4096x2172.
- Supports only CQP, CBR, VBR, ICQ and AVBR rate control methods, as well as `MXF_RATECONTROL_LA_EXT` with lookahead plugin.
- Query max macroblock per second report not supported.

- Next limitation for encoded bitstream:
 - No tiles support;
 - No SAO;
 - No P slice;
 - Progressive only.
- QP Offset for HEVC is 1 by default.
- HW Accelerated HEVC encode supports next extended buffer ids, with listed limitations:
 - MFX_EXTBUFF_CODING_OPTION;
 - MFX_EXTBUFF_OPAQUE_SURFACE_ALLOCATION;
 - MFX_EXTBUFF_HEVC_REFLISTS;
 - MFX_EXTBUFF_HEVC_REFLIST_CTRL;
 - MFX_EXTBUFF_CODING_OPTION_VPS;
 - MFX_EXTBUFF_ENCODED_FRAME_INFO;
 - MFX_EXTBUFF_VIDEO_SIGNAL_INFO;
 - MFX_EXTBUFF_CODING_OPTION2;
 - Supported:
 - MBBRC;
 - BRefType;
 - NumMbPerSlice;
 - DisableDeblockingIdc;
 - RepeatPPS;
 - NumMbPerSlice;
 - MaxFrameSize.
 - MFX_EXTBUFF_CODING_OPTION3;
 - Supported:
 - PRefType;
 - EnableQPOffset;
 - QPOffset.
 - MFX_EXTBUFF_CODING_OPTION_SPSPPS;
 - MFX_EXTBUFF_HEVC_TEMPORAL_LAYERS;
 - MFX_EXTBUFF_ENCODER_RESET_OPTION;
 - MFX_EXTBUFF_HEVC_PARAMS;
 - GeneralConstraintFlags: not supported.
- Encoder with CBR rate control and height bitrate may produce stream with HRD overflows.
- MaxFrameSize violation may happen.
- Encoder with CBR rate control, big resolution and small max bitrate may fail.
- Supports maximum payloads 160 byte per frame.

H.264 AVC/MVC encode limitations:

- Specific QSV-FF encode limitations:
 - Only I and P frame types supported, so no API related to B frames encoding supported.
 - CQP, CBR, VBR and QVBR rate control methods only.
 - Supported QP values range: 10 – 51.
 - Only progressive encoding supported.
 - Maximum number of reference frames equal to 3.
 - Encoding quality can be worse than legacy encode with similar parameters.

- Encoding bitrates higher than 43 Mbps not supported in current implementation.
 - MaxSliceSize feature can produce slices non-compliant to specified value, it is content dependent how often slice size overflow can occur, so need to be tested according to usage and requirement if such violations acceptable or not.
 - QSV-FF encode not utilizing render engine, except next 2 cases:
 - Target Usage equal to 1 and NumRefFrame equal to 3.
 - ARGB used as input.
 - Not supported `mfxExtCodingOption3::AdaptiveMaxFrameSize`.
- MaxQP with MaxFrameSize – not working, MaxQP will have more priority and MaxFrameSize will be violated.
- Frames for different views in single AU in MVC encoder must be provided to encoder in order specified by `mfxMVCViewDependency`.
- `MFx_EXTBUFF_AVC_REFLIST_CTRL` and `MFx_EXTBUFF_CODING_OPTION_SPSPPS` external buffers are not supported by MVC encoder.
- MVC encoder supports `MFx_PROFILE_AVC_STEREO_HIGH` only.
- H.264 encoder in software implementation doesn't support processing of `mfxExtPictureTimingSEI` template. During initialization 0xFFFF values will be reset to default values. In runtime 0xFFFF values will be put to bitstream as is.
- `RefPicMarkRepSEI` syntax is not supported by MVC encoder.
- Known limitations for H.264 Multiple-Segment Encoding:
 - Hardcoded HRD parameters: `bit_rate_scale = 0`, `cpb_size_scale = 3`.
 - Encoded `bit_rate_value_minus1`, `bit_rate_scale` represent BitRate from original SPS within precision of Kbps (maximum supported BitRate is $2^{16} - 1$ Kbps).
 - Encoded `cpb_size_value_minus1`, `cpb_size_scale` represent CpbSize from original SPS within precision of Kb (maximum supported CpbSize is $2^{16} - 1$ Kb).
 - Encoded `time_scale`, `num_units_in_tick` could be both multiplied by 2 if the `time_scale` from original SPS is odd.
 - Conflicts between SPS/PPS and `mfxVideoParam` for parameters that are not covered by SPS/PPS could lead to change of parameters in SPS/PPS.
- Target usage 7 of H.264/MVC encoders in software implementation is known to have a non-monotonic quality versus bitrate dependency.
- MVC encoder ignores any user SEI messages for the dependent view.
- The look ahead bitrate control mode may produce non HRD compliant encoded streams.
- `mfxExtCodingOption2::LookAheadDS` currently supports only `MFx_LOOKAHEAD_DS_OFF` and `MFx_LOOKAHEAD_DS_2x`, `MFx_LOOKAHEAD_DS_4x` will give the same result as `MFx_LOOKAHEAD_DS_2x`. `MFx_LOOKAHEAD_DS_OFF` is the default value for target usage 1 and 2. `MFx_LOOKAHEAD_DS_2x` is the default value for target usages 3 – 7.

- H.264 and MVC encoders may not obey the minimum compression ratio required by the Blu-Ray*/AVCHD* specifications when the requirement is stronger than in H.264 standard.
- The value reported via `mfxExtEncoderCapability::MBPerSec` may be bigger than the actual maximum processing rate of the encoder.
- To change encoding parameters on the fly with `Reset()` function w/o IDR insertion application should drain all the buffered surfaces from encoder. Otherwise encoder may demonstrate undefined behavior after `Reset`.
- Call of `Reset` which starts new sequence (inserts IDR) will drop HRD conformance over the inserted IDR (CPB removal counter will be set to 0 in the IDR Picture Timing SEI).
- HRD violations are possible in specific scenarios (e.g. massive frame skipping).
- Encoder prohibits increase of DPB size (`NumRefFrame`) via `Reset` function even if new size is lower than initialization value. `Reset` function will return `MXF_ERR_INCOMPATIBLE_VIDEO_PARAM` on any attempt to increase `NumRefFrame`.
- Target usage `MXF_TARGETUSAGE_BEST_SPEED` may produce better objective quality than `MXF_TARGETUSAGE_BALANCED`.
- Encoder may not insert PCM macroblocks when required. Encoding of specific (complex) content with huge bitrate (which makes encoder insert many PCM MBs) may cause a GPU hang on Haswell.
- `Reset` function isn't supported for LookAhead BRC modes. `Reset` function doesn't return an error when called together with LA BRC. Result of such `Reset` call is undefined.
- B-pyramid isn't supported together with HRD compliant Look Ahead BRC (`MXF_RATECONTROL_LA_HRD`) and Look Ahead BRC with sliding window control (`MXF_RATECONTROL_LA` and `WinBRCTMaxAvgKbps` and `WinBRCTSize`).
- 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 `MXF_ERR_UNDEFINED_BEHAVIOR` from `EncodeFrameAsync` calls.
- Trellis option can be enabled only on lower target usages, on some of those it is enabled by default but can be switched off. Exact implementation details are hidden and may change with time and between platforms. Use of `Query` function to retrieve actual support is strongly recommended.
- `SkipFrame` feature has the following limitations:
 - If GOP has only P frames, arbitrary P can be skipped. When skipped, it is made non-reference.
 - If GOP has B frames, only non-reference B can be skipped.
- Some fails can happen during encoding in case of reinitialization of encoder in one session with `DirectX11`.

- On specific content and height bitrate encoder doesn't always re-code MB as PCM when its size exceed allowed by standard.
- Encoding using VBR and dynamically adjusted the MaxFrameSize on a per-frame basis algorithm may leads to HRD buffer underflow and quality artifacts.
- Max bit rate violations are possible in case of using VBR and specific payload.

MPEG-2 decode limitations:

- MPEG-2 Video decoder returns `MFx_ERR_UNDEFINED_BEHAVIOR` instead of `MFx_ERR_MORE_DATA` when part of sequence header is absent and `MFx_BITSTREAM_COMPLETE_FRAME` flag is set.
- Decoder does not support bitstreams with resolution bigger than 2096x2096. `MFxVideoDECODE_Init` returns `MFx_ERR_UNSUPPORTED` on such bitstreams.
- Decoder does not support MPEG-1 bitstreams. It is interpreted as corrupted MPEG-2 bitstream. `MFxVideoDECODE_Init` returns `MFx_ERR_NONE` and `MFxVideoDECODE_DecodeFrameAsync` returns `MFx_ERR_MORE_DATA` until valid MPEG-2 bitstream is found.
- Sequence headers are skipped if resolution exceeds maximum supported values (2096x2096) or level/chroma are invalid. It affects `MFxVideoDECODE_DecodeFrameAsync`.
- `mfxDecodeStat.NumFrame` that is returned from `GetDecodeStat` function is less by 1 than actual count of decoded frames if `GetDecodeStat` is called after buffered frames are returned.

VC1 decode limitations:

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

VPP component has next limitations:

- Composition has issues after reset with input system memory usage, input system memory not recommended for use with composition if resetting composition pipeline required.
- `MFx_FRCALGM_DISTRIBUTED_TIMESTAMP` is unsupported by InverseTelecine and Deinterlace (60i->60p) VPP filters.
- `mfxExtVppAuxData` is deprecated.
- `VPP::Query` in software implementation mistakenly indicates support for `MFx_FRCALGM_FRAME_INTERPOLATION` while it is actually not available.
- VPP scaling in software implementation may produce slightly blurred frames for RGB32 interlaced content.
- VPP Image Stabilization and FRC Interpolated filters will be deprecated and not recommended for use due to result image quality and product quality associated with these filters.
- When De-interlace and FRC filters used in pipeline `MFx_DEINTERLACING_BOB` will be used by default, if not specified explicitly to use `MFx_DEINTERLACING_ADVANCED`.

- MFX_FOURCC_YV12 supported only via software fallback with DirectX* 11.1 interface.
- VPP in software implementation always uses simple FRC algorithm based on repeat/drop frames and ignores `MFX_FRCALGM_FRAME_INTERPOLATION` flag.
- ADI may produce color artefacts on frame preceding a scene change.
- ADI may produce color artefacts in case of harmonic motion (repeated pattern and motion magnitude is the same as the periodic of repeated pattern).
- Multiple VPP filters being combined in one session may produce output that is not bit-exact with the output from the same VPP filters that are split by separate sessions, but the difference does not affect visual quality.
- Frames with interlaced content must have CropH multiple of 4. Otherwise, VPP may produce color artefacts on the bottom lines.
- De-interlacing is supported for NV12, YUY2 formats only.
- Once enabled at the Init stage VPP de-interlacing for BOB mode is not disabled automatically if application provides input frames with picstruct set to `MFX_PICSTRUCT_PROGRESSIVE`.
- 1st frame is doubled during 30i->60p de-interlacing for `MFX_DEINTERLACING_BOB` and `MFX_DEINTERLACING_ADVANCED_NOREF` modes.
- `MFXVideoVPP_GetVideoParam` does not update values in attached extended buffers, except `VPP_DO_USE`.
- `MFXVideoVPP_Reset` could return error if additional memory allocation is required based on provided video parameters (changed type of color or frame rate conversion for example). The application should close VPP component and then re-initialize it in this case. See `MFXVideoVPP_Reset` in SDK API Reference Manual for more details about possible return statuses.
- Multiple RGB4 surfaces with `PixelAlphaEnable` enabled could give poor visual quality after composition on the same area with overlapping.
- Deprecated `MFX_EXTBUFF_VPP_PICTSTRUCT_DETECTION` enumerator.
- VPP Mirroring in pipeline with video to video memory in/output patterns can be supported only without any other filters, in/out memory patterns containing system memory is supported with any other legal filters combination.
- ADI with doubled framerate – there is an issue when first 3 frames copied instead of 2.
- On 6th generation Intel Core and Xeon E3 v5 Temporal Denoise enabled – this can affect some result of denoise on scene change and after reset with changed denoise factor.
- `mfxExtVPPSignalInfo` has to be attached to surface with NV12 format only when color conversion done.
- Rotate + Detail/ProcAmp – in combination affect each other, thus result can differ comparing running filters separately and in one call.

The Screen Capture have the following known limitations:

- Screen Capture plugin will use fallback to standard Microsoft* DirectX* Video Acceleration (DXVA) path in case when monitor is connected and application uses exclusive render mode or OPM mode.
- Screen capture with scaling (i.e. when initialization resolution is not equal current display resolution) is supported only with NV12 fourcc format.
- Screen Capture plugin loading recommended only with SDK HW library implementation API 1.15 or later.
- In case of DirectX 11 implementation, video memory type, and RGB4 surface format usage, the application frame allocator needs to allocate the surfaces using DXGI_FORMAT_AYUV format because OS runtime will block RGB surface allocation with BIND_DECODER flag and decoder output view. In any other configuration cases, e.g. DirectX 9 implementation, system or opaque memory type, or NV12 output format, special frame allocation is not needed.
- DXVA fallback functionality doesn't work for resolution change.
- DirtyRect detection feature will detect dirty areas in captured frame and coalesce areas to bigger ones until get 256 areas to feed into DirtyRect API buffer.
- Display Selection API supported only for Virtual Displays.
- Windows 10 acceleration currently unsupported, plugin will fall back to standard Microsoft functions, and performance drop can be observed.

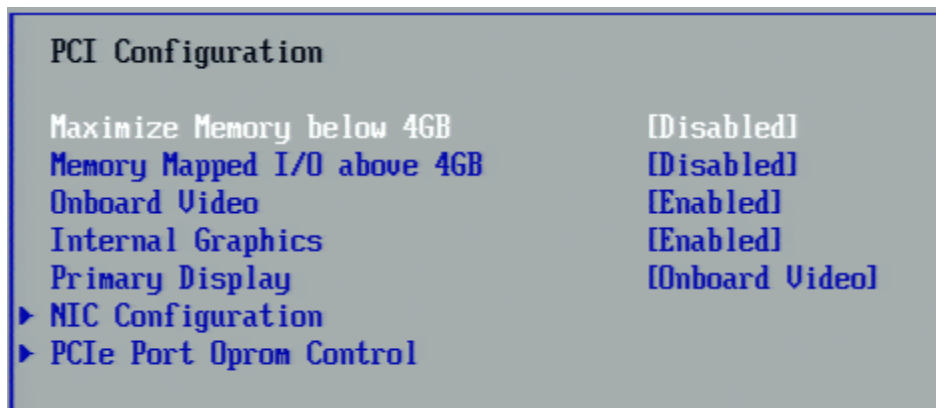
The Media RAW Accelerator Plug-in has the following known limitations:

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

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

Other Limitations

- For Intel® Server Systems R1304RPMISHOR/R1208RPMISHOR:
 - Headless mode is supported only with BIOS version 01.03.0004 or later. Download the Intel® Server Board S1200V3RPM Firmware Update Package for EFI at downloadcenter.intel.com ([link](#)).
- When Intel HD Graphics is not primary display and not connected to an actual display device make sure to manually enable Internal Graphics in BIOS, see the screenshot below for reference:



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.

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