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

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 2016 – Graphics driver, version 15.36.4371
- Intel® Media Server Studio 2016 – Software Development Kit, version 7.0.0000.327
- Intel® Media Server Studio 2016 – Screen Capture plug-in, version 1.17.2.14
- Intel® Media Server Studio 2016 – Advanced AVC Encode plug-in, version 1.17.5.34
- Intel® Media Server Studio 2016 – Samples are not a part of this package, but they can be downloaded from Intel(R) Media Server Studio 2015 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 1.17.

Following fixes, improvements and features were added:

- H.264 Encode:

- Improved AVC encode quality issues for BRefType MFX_B_REF_PYRAMID.
- Fixed support of MaxFrameSize option in MFXVideoENCODE_Reset function.
- Fixed crash in AVC Encode for QVBR rate control method when MaxKbps not provided.
- Fixed skip frame support in AVC encoder
- Added support for AVC Encode to natively consume ARGB surfaces (captured from screen/game) and YUY2 surfaces – from mjpeg decode or deinterlace output to optimize performance.
-

- H.264 decode:
  - Improved robustness and recovery for corrupted content
- Mpeg2 decode
  - Improved robustness and recovery for corrupted content
  - Fixed issue with SPS/PPS export through GetVideoParam
  - Fixed issue with CropX/CropY parameters update in Mpeg2 decode
  - Fixed issue in Mpeg2 decode with run-to-run wrong reporting of picture structure
  -
- VPP
  - YV12 input support for d3d11 VPP returned back.
  - Enabled right handling of VPP reset for ProCamp and other filters.
- VP8 decode
  - fixed issue with wrong picture structure reporting.
- Fixed issues with for user plugin and compatibility with Media Server Studio Professional edition plugins.

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`"

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 E3-128x 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

- Microsoft Windows Server* 2012 R2, 64-bit Microsoft Windows* 8.1 for server/embedded usage. 64-bit Microsoft Windows* 7 for development purpose only.
- Microsoft Visual C++* 2005 with Service Pack 1, or later version of Microsoft Visual C++.

Note: Other combinations of Microsoft Windows Server 2012 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 2016 – SDK installs under C:\Program Files\Intel\Intel® Media Server Studio 2016\– this is referenced as <sdk-install-dir> in the remainder of this document.

`<sdk-install-dir>\media_studio_2015_sdk_release_notes.pdf`	Intel® Media Server Studio 2016 – Driver, SDK Release Notes (this file)
`<sdk-install-dir>\Software Development Kit\bin\x64`	Intel® Media Server Studio 2016 – SDK Dynamic Library, software implementation: `libmfxsw64.dll` for Intel® 64 architecture **Note:** Hardware implementation of Intel® Media Server Studio 2016 – SDK Dynamic Library `libmfxhw64.dll` is packed and installed with Intel® Media Server Studio 2016 – Graphics Driver (default location is C:\Program Files\Intel\Media SDK)
`<sdk-install-dir>\Software Development Kit\bin\22d62c07e672408fbb4cc20ed7a053e4`	Intel® Media Server Studio 2016 – Screen Capture plug-in: • Screen Capture plug-in `mfxplugin64_screen_capture.dll`

	• Configuration file `plugin.cfg`

`<sdk-install-dir>\Software Development Kit\bin\588f1185d47b42968dea377bb5d0dcb4`	Intel® Media Server Studio 2016 – 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 2016 – SDK documentation
`<sdk-install-dir>\ Software Development Kit\include`	External Intel® Media Server Studio 2015 – SDK headers: • Type definitions in `mfxdefs.h` • Structure definitions in `mfxstructures.h` • Function definitions in C in `mfxvideo.h` • C++ wrapper of the SDK functions in `mfxvideo++.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` • Extensions for JPEG*/Motion JPEG Video coding options `mfxjpeg.h`
`<sdk-install-dir>\Software Development Kit\lib\x64`	Intel® Media Server Studio 2016 – SDK Static Dispatcher Library `libmfx.lib`
`<sdk-install-dir>\Software Development Kit\igfx_s3dcontrol`	Intel® Media Server Studio 2016 – SDK Stereoscopic 3D API for Intel Iris and HD Graphics hardware, includes: • S3D API definitions `igfx_s3dcontrol.h` • Static S3D Control Library `igfx_s3dcontrol.lib` • 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 `mediasdk_tracer`. 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`	Source code of Intel® Media Server Studio 2016 – SDK Dispatcher

Documentation

You can find more information on how to use Intel® Media Server Studio 2015 - SDK in the following documentation:

- `<sdk-install-dir>\Software Development Kit\doc\mediasdk-man.pdf`

  "Intel® Media Server Studio - SDK Reference Manual" describes the SDK API.

- `<sdk-install-dir>\Software Development Kit\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-dir>\Software Development Kit\doc\mediasdkjpeg-man.pdf`

  "Intel® Media Server Studio - SDK Reference Manual 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 Extensions 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 - SDK Library Distribution and Dispatching Process".

- `<sdk-install-dir>\Software Development Kit\doc\mediasdkscreencap-man.pdf`

  "Intel® Media Server Studio - SDK Screen Capture Decoder Reference Manual"

Known Limitations

The Intel® Media Server Studio - SDK libraries 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`
- 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
- mfxExtCodingOption2:: EnableMBQP, DirectBiasAdjustment, GlobalMotionBiasAdjustment
- mfxExtChromaLocInfo
- mfxExtMBQP
- MFX_PLUGINID_HEVCE_HW, MFX_EXTBUFF_HEVC_PARAM, mfxExtHEVCParam
- MFXInitEx, MFXDoWork, mfxInitParam, mfxExtThreadsParam
- mfxInfoMFX::LowPower
- mfxInitParam::GPUCopy
- mfxExtPredWeightTable
- mfxExtDirtyRect
- mfxExtMovingRect
- mfxExtCodingOption3::ScenarioInfo
- mfxExtCodingOption3::ContentInfo
- MFX_MEMTYPE_EXPORT_FRAME
- mfxExtCodingOptionVPS
- mfxExtVPPRotation
- mfxExtCodingOption3::FadeDetection
- mfxExtVPPVideoSignalInfo

Functional

- 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.
- 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.

- 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.
- `RefPicMarkRepSEI` syntax is not supported by MVC encoder.
- If the MPEG-2 Video encoder `mfxVideoParam::mfxInfoMFX::CodecProfile` is initialized to 0, then the stream will be encoded as `MFX_PROFILE_MPEG2_MAIN`. Additionally if the MPEG-2 Video encoder `mfxVideoParam::mfxInfoMFX::CodecLevel` is initialized to `0`, then the stream will be encoded as `MFX_LEVEL_MPEG2_MAIN`.
- `MFX_FRCALGM_DISTRIBUTED_TIMESTAMP` is unsupported by InverseTelecine and Deinterlace (60i->60p) VPP filters.
- 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.
- Target usage 7 of H.264/MVC encoders in software implementation is known to have a non-monotonic quality vs. bitrate dependency.
- MPEG-2 Video, VC-1 and MVC decoders are not optimized for low delay of output frames.
- MVC encoder ignores any user SEI messages for the dependent view.
- `MFX_CORRUPTION_ABSENT_TOP_FIELD`, `MFX_CORRUPTION_ABSENT_BOTTOM_FIELD`, `MFX_BITSTREAM_EOS` are not supported by VC-1, MPEG-2 Video and JPEG decoders.
- VPP in software implementation always uses simple FRC algorithm based on repeat/drop frames and ignores `MFX_FRCALGM_FRAME_INTERPOLATION` flag.
- The feature set of JPEG decoder/encoder is limited to the following:
  - Baseline mode only
    - DCT based
    - 8-bit samples
    - sequential

        - loadable 2 AC and 2 DC Huffman tables
        - 3 loadable quantization matrixes
        - interleaved and non-interleaved scans
        - single and multiple scans
    - No extended, lossless and hierarchical modes
        - no 12-bit samples
        - no progressive
        - no arithmetic coding
        - no 4 AC and 4 DC Huffman tables
- H.264 encoder and decoder in software implementation are known to be a little bit slower compared with Intel® Media SDK 2012 R2.
- The output AVC and MVC streams contain SPS and PPS headers before IDR frames only.
- Encoders and VPP don't support `mfxExtVppAuxData::PicStruct`.
- VPP scaling in software implementation may produce slightly blurred frames for RGB32 interlaced content.
- JPEG decoder does not set `Corrupted` flag of `mfxFrameData` structure, and does not accept `MFX_BITSTREAM_EOS` as `DataFlag` of `mfxBitstream` structure.
- 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.
- The look ahead bitrate control mode may produce non HRD compliant encoded streams.
- `VPP::Query` in software implementation mistakenly indicates support for `MFX_FRCALGM_FRAME_INTERPOLATION` while it is actually not available.
- 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.
- `VPP::Reset` does not apply dynamic changes made to extended buffers such as `mfxExtVPPProcAmp`. The current workaround is to call `VPP::Close` directly followed by `VPP::Init` with the new configuration.
- 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.
- `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.

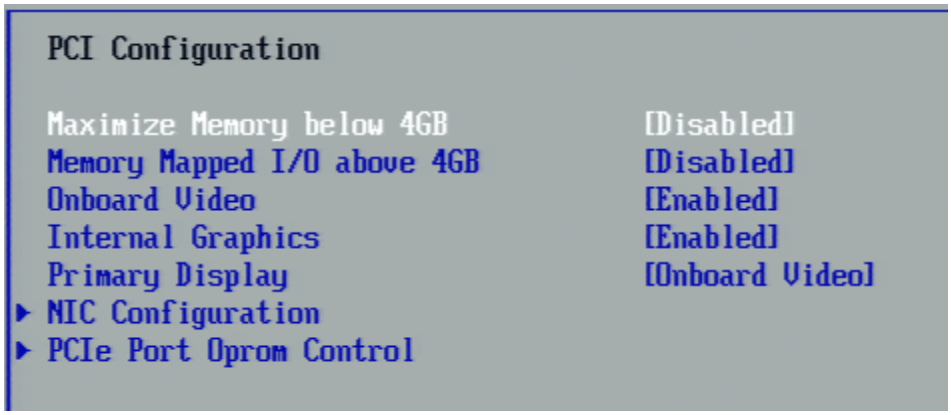
- The value reported via `mfxExtEncoderCapability::MBPerSec` may be bigger than the actual maximum processing rate of the encoder.
- This release supports only 64-bit Microsoft* Windows* applications.
- Microsoft DirectX* 11.1 is the only supported acceleration infrastructure (due to headless mode requirement).
- HW HEVC decode plugin is limited to 4096x2304 resolution and doesn't implement SW fallback for higher resolutions. You may use software implementation of HEVC decode from Intel® Media Server Studio 2015 - Professional Edition to support higher resolutions.
- 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.
- 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.
- Screen capture with scaling (i.e. when initialization resolution is not equal current display resolution) is supported only with NV12 fourcc format
- HEVC HW plugin MAIN10 profile supported only on 5th Generation Intel Core and Xeon E3 v4.
- 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
- 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.
- MFX_FOURCC_YV12 supported only via software fallback with DirectX* 11.1 interface.
- H264 decode 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.

- Mpeg2 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.
- H264 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.
- H264 Encode: 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).
- H264 Encode: HRD violations are possible in specific scenarios (e.g. massive frame skipping).
- H264 Encode: 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.
- H264 Encode: Target usage MFX_TARGETUSAGE_BEST_SPEED may produce better objective quality than MFX_TARGETUSAGE_BALANCED.
- H264 Encode: 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.
- H264 Encode: 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.
- H264 Encode: 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).
- H264 Encode: 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.
- H264 Encode: 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.
- H264 Encode: 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.
- H264 Encode: 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.

- MPEG2 Decode: Decoder does not support bitstreams with resolution bigger than 2096x2096. MFXVideoDECODE_Init returns MFX_ERR_UNSUPPORTED on such bitstreams.
- MPEG2 Decode: 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.
- MPEG2 Decode: Sequence headers are skipped if resolution exceeds maximum supported values (2096x2096) or level/chroma are invalid. It affects MFXVideoDECODE_DecodeFrameAsync.
- MPEG2 Decode: 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.
- VPP: ADI may produce color artefacts on frame preceding a scene change.
- VPP: ADI may produce color artefacts in case of harmonic motion (repeated pattern and motion magnitude is the same as the periodic of repeated pattern).
- VPP: Multiple VPP filters being combined in one session may produce output that is not bit-exact with the output from the same VPP filters that are split by separate sessions, but the difference does not affect visual quality.
- VPP: Frames with interlaced content must have CropH multiple of 4. Otherwise, VPP may produce color artefacts on the bottom lines.
- VPP: De-interlacing is supported for NV12, YUY2 formats only.
- VPP: 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.
- VPP: 1st frame is doubled during 30i->60p de-interlacing for MFX_DEINTERLACING_BOB and MFX_DEINTERLACING_ADVANCED_NOREF modes
- VPP: MFXVideoVPP_GetVideoParam does not update values in attached extended buffers, except VPP_DO_USE
- VPP: 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.
- VPP: Multiple RGB4 surfaces with PixelAlphaEnable enabled could give poor visual quality after composition on the same area with overlapping.
- VPP: deprecated `MFX_EXTBUFF_VPP_PICTSTRUCT_DETECTION` enumerator

Other Limitations

- For Intel® Server Systems R1304RPMSHOR/ R1208RPMSHOR:
    - 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