

# Intel® Media Server Studio 2018 R1 – Driver, SDK for Linux\* Release Notes

[Overview](#)

[What's New](#)

[System Requirements](#)

[Features](#)

[GPU Hang Reporting And Recovery](#)

[Package Contents](#)

[Installation Folders](#)

[Documentation](#)

[Known Limitations](#)

[Legal Information](#)

[Attributions](#)

## Overview

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

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

The package includes the following components:

- Intel® Media Server Studio 2018 R1 – Graphics Driver, version 16.8.69752
- Intel® Media Server Studio 2018 R1 – SDK, version 7.0.16083982
- Intel® Media Server Studio 2018 R1 – Samples Binaries, version 8.2.25.982. The latest version of samples package (with all samples binaries and corresponding source code) could be downloaded from [Intel® Media Server Studio Support](#).

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

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

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

## What's New

Intel® Media Server Studio 2018 R1 – SDK (hereinafter referred to as "SDK"):

**Version 7.0.16083982 / Graphics Driver Version 16.8.69752 Hotfix:**

**Bug fixes:**

- **Misc:**
  - [40532] Fixed an issue that could cause an application to hang or run very slowly when ring buffers were oversubscribed.

## Version 7.0.16083982 / Graphics Driver Version 16.8.69021:

### New features:

- **OS and hardware support**
  - CentOS 7.4. The release supports original 3.10.0-693.11.6.el7.x86\_64 kernel of CentOS 7.4.
  - The release contains a set of patches for open source components on top of 4.4.111 kernel suitable for other variants of Linux OSes.
- **SDK API 1.25**
  - Added *mfExtCodingOption3::EnableNalUnitType* and *mfEncodeCtrl::MfxNalUnitType* fields.
  - *mfExtMultiFrameParam*, *mfExtMultiFrameControl* Multi-Frame encode APIs support added in this release for evaluation purposes only. See the Known Limitations section for details.
  - Added *mfExtBRC*, *mfBRCTFrameParam*, *mfBRCTFrameCtrl*, *mfBRCTFrameStatus* External BRC definitions.
- **HEVC encode**
  - [36866] Enabled setting NAL unit types from application in the encoded order mode.
  - [36347] Enabled interlaced content encoding support. Please see the *<sdk-install-folder>/doc/mediasdk\_hevc\_interlace\_whitepaper.pdf* for more details.
- **H.264 encode**
  - [36670] Enabled P/I field encoding (first field encoded as P field, second as I field) support for Skylake and Broadwell platforms. User might configure it to achieve better quality improvement for scenarios when there is scene change in-between the two fields in the same frame.
- **VPP**
  - [36655] Enable deinterlacer to support mixed picture structure. This allows VPP DI to handle content that alternates between Progressive/Interlaced frame, Top Field First (TFF) and Bottom Field First (BFF). See Known Limitations section for details.
- **Misc:**
  - In this release media kernels for VPP, H.264 and HEVC encoders were adjusted to improve subjective quality and robustness of corresponding components. This may lead to nonbitexact output comparing to previous Intel® Media Server Studio releases.
  - SDK, hardware implementation, migrated to 2.0 version of Video Acceleration API library (hereinafter referred to as "VA-API"). The SDK API remains backward compatible. However VA-API distributed in this package doesn't provide *libva\*.so.1* libraries and provides *libva\*.so.2* instead. If application has a direct dependency from *libva\*.so.1* libraries, then application should be re-compiled with dependency from *libva\*.so.2* or symlinks *libva\*.so*.
  - New version of Metrics Monitor tool based on i915 PMU API to query metrics. i915 PMU API allows to collect precise engines utilization tracking instead of sampling estimations. Uses of i915 PMU API lead to reduced CPU usage overhead in case of precise engines utilization path. Manual can be found in separate document *<sdk-install-dir>/tools/metrics\_monitor/doc/metricsmon-man.pdf* "Intel® Media Server Studio - Metrics Monitor (v1.1.2) Reference Manual".
  - Two implementations of new bit-rate controls introduced in this release: one of them is implemented at samples level and can be enabled through *mfExtCodingOption2::ExtBRC=MFX\_CODINGOPTION\_ON* **together with callback functions** passed to SDK library through *mfExtBRC* extended version. The second implementation is made inside of the library and identical

to the samples one. The reason of having two identical implementations at different stack levels is getting access to new bitrate control without application recompilation. Use `mfxExtCodingOption2::ExtBRC=MFX_CODINGOPTION_ON` flag **without** `mfxExtBRC` extended buffer to enable SDK in-built bit-rate control. Both implementations are CPU based and support CBR and VBR modes. Please follow "mfxExtBRC" section of the SDK manual for details.

## Bug fixes:

- **HEVC encode**

- [36579] Fixed a low visual quality issue with per-LCU QP feature.
- [36902] Fixed a CABAC issue when coding in the skip mode encoder put an extra bit (aligned to a byte in some cases) at the end of slices .

- **H.264 encode**

- [37798] Encoder didn't insert emulation prevention bytes for slice headers. The issue is fixed.
- [37968] Fixed *Query/Init* functions a return status when in CQP or LA BRC modes encoder was initialized with disabled BRC panic mode. The right status is `MFX_ERR_UNSUPPORTED`.

- **MPEG-2 encode:**

- [37967] Fixed *Query/Init* functions a return status when in CQP mode encoder was initialized with disabled BRC panic mode. The right status is `MFX_ERR_UNSUPPORTED`.

- **H.264 decode:**

- [37755] Fixed segmentation faults on streams with corrupted PPS slice group information.
- [39887] Fixed incorrect "Corruption" flag reporting on streams with ref pic list modification syntax.

- **HEVC decode**

- [37169, 37184] Fixed an invalid decoder behavior when decoder returned `MFX_ERR_UNKNOWN` if RPL contained an LTR frame missing in DPB.
- [37168] Fixed a GPU hang issue on corrupted streams when `log2_min_luma_coding_block_size`, `log2_max_luma_coding_block_size` syntax exceeded valid ranges.
- [36224] Fixed decoder behavior when decoder returned `MFX_ERR_UNDEFINED_BEHAVIOR` on streams when `pps_pic_parameter_set_id` syntax exceeded valid ranges.
- [37211] HEVC decoder freeze fixed, it was caused due to DPB overflow on streams with RPS corruption.

- **VPP:**

- [37007] Fixed resize issue when VPP produced corrupted output (erroneous blank black "bottom field" with erroneous compressed "top field") for input containing mixed interlace and progressive frames.
- [28637] Fixed bit-exactness issue for VPP scaling interlaced fields with oversized surface. Prior to the fix, use of an input or output surface with an oversized height would result in bit exact mismatch compared to actual height, as equalization in the scaling algorithm used surface height. The fix corrects this issue by using actual crop height when oversize surface is in use.
- Following limitation is not present anymore "[27481] 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`".

- **Performance:**

- Fixed performance drop on workloads which uses SDK Sessions Joining API (`MFXJoinSession`). Improvements are related to the changes in components scheduling policy.
- CPU usage has significant improvement in comparison to Media Server Studio 2017 R3 due to SDK scheduler changes. In average (it depends on particular

workload): 11% for 1:N, 10 % for 1:N Predefined, 10% for N:N, 23% for N:N Multiprocess, 6% for Composition, 17% for FEI workloads.

- **Misc**

- [35703] Usage of non-dispatched AVX instructions in hardware and software libraries was removed.
- Following limitation is not present anymore "[35341] Dispatcher CMakeLists.txt misses following function:

```
function( get_mfx_version mfx_version_major mfx_version_minor )
  file(STRINGS /opt/intel/mediasdk/include/mfxvideo.h major REGEX
"#define MFX_VERSION_MAJOR")
  file(STRINGS /opt/intel/mediasdk/include/mfxvideo.h minor REGEX
"#define MFX_VERSION_MINOR")
  string(REPLACE "#define MFX_VERSION_MAJOR " " " major ${major})
  string(REPLACE "#define MFX_VERSION_MINOR " " " minor ${minor})
  set(${mfx_version_major} ${major} PARENT_SCOPE)
  set(${mfx_version_minor} ${minor} PARENT_SCOPE)
endfunction()
```

## System Requirements

### Hardware

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

- Intel® Xeon® E3-1200 v4 Family with C226 chipset
- Intel® Xeon® E3-1200 and E3-1500 v5 Family with C236 chipset
- 5th Generation Intel® Core™
- 6th Generation Intel® Core™
- Note: chipset must have processor graphics enabled; make sure to check the datasheet.
  - Having a C226/C236 chipset is necessary but not sufficient. Make sure to consult with specific platform or board vendor regarding processor graphics being supported. Check Media Server Studio website for the list of “Known OEM/ODM Functional Platforms”:  
<https://software.intel.com/en-us/intel-media-server-studio/details>

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

### Software

- **CentOS 7.4 (1708)** of 64 - bit architecture its default or latest kernels. **Installation process applies many changes to the kernel, graphics driver, libdrm and libva graphics stack. These changes would need to be reverted to request OS vendor support.**
- Future CentOS 7.4 releases may break Intel Media Server Studio since their version is too new - the kernel cannot work with our kernel mode driver release, libdrm will prevent our patched version libdrm to be installed. So system should be kept as CentOS 7.4 level.
- Installation script handles incompatible CentOS upgrades by utilizing "yum --releaserver=7.4.1708".

**Important Note:** It appears that CentOS has a bug in mirrorlist backend which requires manual edit of `/etc/yum.repos.d/CentOS-Base.repo` file. Correct installation procedure requires commenting out mirrorlist lines and uncommenting baseurl lines. Like:

```
#mirrorlist=http://mirrorlist.centos.org/?
release=\$releasever&arch=\$basearch&repo=os&infra=\$infra
baseurl=http://mirror.centos.org/centos/\$releasever/os/\$basearch/
```

Installation script automatically perform check of "--releaserver" functionality and abort

installation if environment is not correctly configured.

- Generic OS install uses kernel 4.4.111 from [www.kernel.org](http://www.kernel.org) .

## Features

Intel® Media Server Studio 2018 – SDK included in this package implements SDK API 1.25 and contains the following components:

Component	Supported features	Limitations
H.265 decoder	Supported Profiles: <ul style="list-style-type: none"> <li>• Main</li> </ul>	Maximum supported resolution: 8192x8192
H.265 encoder	Supported Profiles: <ul style="list-style-type: none"> <li>• Main</li> </ul> Supported BRC methods: <ul style="list-style-type: none"> <li>• Constant QP (CQP)</li> <li>• Constant Bit Rate (CBR)</li> <li>• Variable Bit Rate (VBR)</li> <li>• Software BRC</li> </ul>	Maximum supported resolution: 4096x2176
H.264 decoder	Supported Profiles: <ul style="list-style-type: none"> <li>• Baseline</li> <li>• Main</li> <li>• High</li> </ul>	Maximum supported resolution: 4096x2304
H.264 encoder	Supported Profiles: <ul style="list-style-type: none"> <li>• Baseline</li> <li>• Main</li> <li>• High</li> </ul> Supported BRC methods: <ul style="list-style-type: none"> <li>• Constant QP (CQP)</li> <li>• Constant Bit Rate (CBR)</li> <li>• Variable Bit Rate (VBR)</li> <li>• Look Ahead (LA)</li> <li>• Software BRC</li> </ul>	Maximum supported resolution: 4096x4096
MPEG-2 decoder	Supported Profiles: <ul style="list-style-type: none"> <li>• Baseline</li> <li>• Main</li> <li>• High</li> </ul>	Maximum supported resolution: 2048x2048
MPEG-2 encoder	Supported Profiles: <ul style="list-style-type: none"> <li>• Simple</li> <li>• Main</li> </ul> Supported BRC methods: <ul style="list-style-type: none"> <li>• Constant QP (CQP)</li> <li>• Constant Bit Rate (CBR)</li> <li>• Variable Bit Rate (VBR)</li> </ul>	Maximum supported resolution: 1920x1088
VC1 decoder	Supported Profiles: <ul style="list-style-type: none"> <li>• Simple</li> <li>• Main</li> <li>• Advanced</li> </ul>	Maximum supported resolution: 1920x1088

Component	Supported features	Limitations
MJPEG encoder	Supported Profiles: <ul style="list-style-type: none"> <li>Baseline mode, 8bit</li> </ul>	Maximum supported resolution: per ISO/IEC 14495-1 and system memory limitations
MJPEG decoder	Supported Profiles: <ul style="list-style-type: none"> <li>Baseline mode, 8bit</li> </ul>	Maximum supported resolution: per ISO/IEC 14495-1 and system memory limitations
VP8 decoder	Supported Profiles: <ul style="list-style-type: none"> <li>Version 0 mode, 8bit</li> </ul>	Maximum supported resolution: 1920x1088
Video PreProcessing (VPP)	Supported Algorithms: <ul style="list-style-type: none"> <li>Color Conversion</li> <li>Scaling</li> <li>De-Interlacing(Advanced motion-compensated, BOB)</li> <li>De-noising</li> <li>Frame Rate Conversion</li> <li>Composition</li> <li>Alpha Blending</li> <li>Sharpness</li> <li>PROCAMP</li> <li>Rotation</li> </ul>	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 - SDK Reference Manual for details "[<sdk-install-folder>/doc/mediasdk-man.pdf](#)".

## GPU Hang Reporting And Recovery

Intel® Media Server Studio supports reporting of GPU hangs occurred during SDK operations. This feature allows application to establish proper GPU hang recovery procedure without the need for additional monitoring of the system (e.g. without checking dmesg logs).

In this release, recommended GPU hang recovery procedure was modified (see below).

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

**SDK-independent GPU hang reporting procedure:** there is recommended way how to monitor GPU hang incidents independently from SDK. There are the following uevent messages being sent via udev system when GPU hang is happened:

- "ERROR=1" – GPU hang happened
- "RESET=1" – GPU reset is started (GPU is being recovered after GPU hang)
- "ERROR=0" – GPU successfully recovered after GPU hang (reset procedure is completed)

To be able to receive these messages, application need to monitor uevents of drm subsystem (devtype is "drm\_minor"):

```
uevent_monitor = udev_monitor_new_from_netlink(udev, "udev");
udev_monitor_filter_add_match_subsystem_devtype(uevent_monitor, "drm",
"drm_minor");
```

**GPU hang recovery procedure:** it is recommended to process *MFX\_ERR\_DEVICE\_FAILED*, *MFX\_ERR\_GPU\_HANG* and *MFX\_ERR\_ABORTED* uniformly using the full reset procedure described in “Hardware Device Error Handling” of SDK manual. (I.e. recreate all resources: acceleration device, frames memory, SDK sessions, SDK components).

Informative: usually it takes SDK a few seconds to detect and report GPU hang. During this time all *SyncOperation()* calls for tasks affected by GPU hang will return status *MFX\_WRN\_IN\_EXECUTION*. SDK will report the hang with status *MFX\_ERR\_GPU\_HANG* only after GPU hang is detected and HW recovery mechanism is started by driver.

**Related limitations:**

- GPU hang reporting is supported for Linux only.
- In case of multiprocess usage model, SDK may return status *MFX\_ERR\_GPU\_HANG* not to all processes. Application should ensure that upon receiving *MFX\_ERR\_GPU\_HANG* in at least one process, GPU hang recovery procedure will be performed for all processes used the same GPU. To achieve this, application can establish intra process broadcast of *MFX\_ERR\_GPU\_HANG* status returned from SDK session, monitor dmesg or follow “SDK-independent GPU hang reporting procedure” procedure.
- SDK decoders may buffer decoded frames. Decoder *SyncOperation()* call may return previously buffered frame with *MFX\_ERR\_NONE* status at the time when GPU hang was already reported by underlying HW for one of next frames. This means that even for synchronous pipeline application may get *MFX\_ERR\_GPU\_HANG* from *DecodeFrameAsync()* earlier than from *SyncOperation()*.
- False positive GPU hangs (*MFX\_ERR\_GPU\_HANG*) can be reported in a very heavy workloads (CPU and GPU utilization are close to 100%).

**Package Contents**

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

Component	Description
CentOS/intel-openc1-<id>.*.x86_64.rpm CentOS/intel-openc1-devel-<id>.*.x86_64.rpm CentOS/intel-openc1-cpu-<id>.*.x86_64.rpm	Intel® Media Server Studio - OpenCL™ Driver packages
CentOS/intel-linux-media-<id>.*.x86_64.rpm	Intel® Media Server Studio - Driver & SDK runtime package.
CentOS/intel-linux-media-devel-<id>.*.x86_64.rpm	Intel® Media Server Studio - Driver & SDK development package
CentOS/kmod-ukmd-<id>.*.x86_64.rpm CentOS/ukmd-kmod-<id>.*.src.rpm	ukmd kernel module(s)
CentOS/libdrm*-<id>.*.x86_64.rpm CentOS/drm-utils*-<id>.*.x86_64.rpm	Direct Rendering Manager runtime library runtime, development, etc
CentOS/libva*-<id>.*.x86_64.rpm CentOS/libva-utils*-<id>.*.x86_64.rpm	Video Acceleration (VA) API runtime, development, etc.
CentOS/install_scripts_centos_<id>.tar.gz CentOS/install_scripts_centos_<id>.tar.gz.sig	Intel® Media Server Studio – Driver & SDK installation scripts.
CentOS/MediaSamples_Linux_bin-<id>.tar.gz	Intel® Media Server Studio – Samples package (binaries only).

Component	Description
Generic/intel-linux-media_generic_<id>_64bit.tar.gz Generic/intel-openc1-<id>.tar.xz Generic/intel-openc1-devel-<id>.tar.xz Generic/intel-openc1-cpu-<id>.x86_64.tar.xz Generic/intel-linux-media_generic_<id>_64bit.tar.gz.sig Generic/intel-openc1-<id>.tar.xz.sig Generic/intel-openc1-devel-<id>.tar.xz.sig Generic/intel-openc1-cpu-<id>.x86_64.tar.xz.sig	Intel® Media Server Studio – Driver & SDK & OpenCL™ package for Generic OS.
intel-linux-media-patches-<id>.tar.gz	Optional tar ball may include additional patches for open source components. All details please see in README file within the tar ball.
media_server_studio_sdk_release_notes.pdf media_server_studio_sdk_getting_started_guide.pdf	Intel® Media Server Studio – SDK documentation: this file, Getting Started Guide.
intel-openc1-release-notes.pdf intel-openc1-installation.pdf	OpenCL™ Driver Release Notes and Installation Guide.

**NOTE:** uninstall\_sdk\_CentOS.sh script (which is contained in CentOS/install\_scripts\_centos\_<id>.tar.gz) was taken from prod-pc-linux-16.8-68017 driver package. Script contains fix for libdrm version downgrading.

## Installation Folders

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

Component	Description
<sdk-install-dir>/lib64	Intel® Media Server Studio – SDK Dynamic Library, hardware implementation libmfhw64-p.so.* software implementation libmfsw64-p.so.*
<sdk-install-dir>/doc	Intel® Media Server Studio – SDK documentation

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

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

Component	Description
<sdk-install-dir>/lib64	Intel® Media Server Studio – Graphics Driver
<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 – Graphics Driver, Kernel Mode Driver (KMD)
/usr/include	Direct Rendering Manager runtime library, Video Acceleration (VA) API includes

Component	Description
/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/opencv/include/CL	OpenCL™ Driver includes
/opt/intel/opencv	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 `<sdk-extract-folder>/intel-opencv-release-notes.pdf`.

## Documentation

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

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

- `<sdk-install-dir>/doc/mediasdk-man.pdf`  
"Intel Media Server Studio - SDK Reference Manual" describes the Intel Media SDK API.
- `<sdk-install-dir>/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>/doc/mediasdkjpeg-man.pdf`  
"Intel® Media Server Studio - SDK Reference Manual for JPEG\*/Motion JPEG" describes SDK API for JPEG\* processing.
- `<sdk-install-dir>/doc/mediasdkvp8-man.pdf`  
"Intel® Media Server Studio - SDK Reference Manual for VP8\*" describes SDK extension to support VP8\* video codec.
- `<sdk-install-dir>/doc/mediasdk_hevc_interlace_whitepaper.pdf`  
Intel® Media Server Studio - SDK whitepaper describes Interlace content support in HEVC encoder.

## Known Limitations

This release is subject to the following known limitations:

- **Changes in behavior compared to Media Server Studio 2017 R3 - SDK (version 7.0.16053710):**
  - Intel Media Server Studio 2018 R1 - SDK and Driver are not compatible with previously released versions of Intel Media Server Studio - Essentials and Professional Editions including HEVC Decode plug-in, HEVC Encode plug-in, HEVC GPU Accelerated Encode plug-in, Premium Telecine Interlace Reverser plug-in, etc.
  - MPEG-2 encode:
    - [37189] Previously `Query()` function with `MXF_RATECONTROL_AVBR` mode in input parameters returned `MXF_RATECONTROL_VBR`. The logic is removed in this release.
  - VPP:
    - [35218] VPP resize of interlaced streams may produce output that is not bit-exact with output of previous release.
- **Reliability on Broadwell platforms**
  - This release is subject to Errata **BDW110** (for platforms of Intel® Xeon® E3-1200 v4 Family) and **BDM119** (for 5th Generation Intel® Core™ Processors).

In the result, random rare segmentation faults might be observed. For more information on Errata please refer to Specification Updates available at <http://www.intel.com/content/dam/www/public/us/en/documents/specification-updates/xeon-e3-1200v4-spec-update.pdf> for Intel® Xeon® E3-1200 v4 and at <http://www.intel.com/content/dam/www/public/us/en/documents/specification-updates/5th-gen-core-family-spec-update.pdf> for 5th Generation Intel® Core™ Processors. Please contact your hardware vendor for a BIOS update with a workaround.

- **Hardware Device Error Handling**

- Application should treat *MFX\_ERR\_ABORTED* status returned from *MFXVideoCORE\_SyncOperation()* as *MFX\_ERR\_DEVICE\_FAILED* and run recovery procedure as described in Hardware Device Error Handling section of the SDK manual.

- **API:**

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

- Only the following features among those introduced in API 1.7 are supported:
  - *RateControlMethod::MFX\_RATECONTROL\_LA*
  - *mfExtCodingOption2::LookAheadDepth*
  - *mfExtCodingOption2::MBBRC*
  - *mfExtCodingOption2::Trellis*
- Only the following features among those introduced in API 1.8 are supported:
  - *mfxVideoCodecPlugin*
  - *mfExtVPPComposite*
  - *mfExtVPPDeinterlacing*
  - *mfExtCodingOption2::LookAheadDS, RepeatPPS, BRefType*
  - *mfHandleType::MFX\_HANDLE\_VA\_DISPLAY*
  - *mfImpl::MFX\_IMPL\_VIA\_VAAPI, mfImpl::MFX\_IMPL\_AUDIO*
  - *CodecFormatFourCC::MFX\_CODEC\_HEVC, CodecLevel::HEVC level and tier definitions, CodecProfile::HEVC profile definitions*
  - *BRefControl*
  - *mfFrameData::PitchHigh, PitchLow*
- Only the following features among those introduced in API 1.9 are supported:
  - *mfExtVPPComposite, mfVPPCompInputStream::LumaKeyEnable, LumaKeyMin, LumaKeyMax, GlobalAlphaEnable, GlobalAlpha, PixelAlphaEnable*
  - *mfExtAVCRefLists*
  - *mfExtAVCEncodedFrameInfo::secondFieldOffset*
  - *mfExtCodingOption2::SkipFrame, supported for AVC and MPEG Encode*
  - *ColorFourCC::MFX\_FOURCC\_P010, MFX\_FOURCC\_A2RGB*
  - *mfExtCodingOption2::MaxSliceSize*
- Only the following features among those introduced in API 1.10 are supported:
  - *MFXVideoENC class of functions*
  - *mfENCInput*
  - *mfENCOutput*
  - *mfExtLAControl*
  - *mfExtLAFrameStatistics*
  - *RateControlMethod::MFX\_RATECONTROL\_LA\_EXT*
  - *mfExtCodingOption2::BufferingPeriodSEI and enum {MFX\_BPSEI\_DEFAULT, MFX\_BPSEI\_IFRAME}*
- Only the following features among those introduced in API 1.11 are supported:
  - *mfExtCodingOption3::WinBRCMaxAvgKbps, WinBRCMaxAvgKbps*
  - *mfFrameData:: NumExtParam, ExtParam*
  - *mfExtVPPFieldProcessing, enum VPPFieldProcessingMode, enum PicType*

- RateControlMethod::MFX\_RATECONTROL\_LA\_HRD
  - ExtendedBufferID::MFX\_EXTBUFF\_CODING\_OPTION3, MFX\_EXTBUFF\_VPP\_FIELD\_PROCESSING
- Only the following features among those introduced in API 1.12 and 1.13 are supported:
  - mfxExtCodingOption2::UseRawRef
  - mfxExtCodingOption3::DirecrBiasAdjustment, GLobalMotionBiasAdjustment, MVCostScalingFactor, MBDisableSkipMap, EnableMBQP.
  - mfxExtChromaLocInfo
  - mfxExtMBDisableSkipMap
  - mfxExtMBQP
  - MFXVideoUSER\_LoadByPath()
  - For mfxExtVPPDeinterlacing extended buffer only DeinterlacingModes MFX\_DEINTERLACING\_BOB and MFX\_DEINTERLACING\_ADVANCED are supported
- Only the following features among those introduced in API 1.14 are supported:
  - mfxExtDecodedFrameInfo
  - mfxExtTimeCode
- Only the following features among those introduced in API 1.15 are supported:
  - mfxExtThreadsParam and MFX\_EXTBUF\_THREADS\_PARAM enumerator
- Only the following features among those introduced in API 1.16 are supported:
  - GPUCopy enumerator and control in mfxInitParam structure.
  - mfxFrameAllocRequest.AllocId
  - mfxInfoMFX.MaxDecFrameBuffering
- Only the following features among those introduced in API 1.17 are supported:
  - mfxVideoParam.AllocId field
  - DeinterlacingMode::MFX\_DEINTERLACIF\_ADVANCED\_NOREF enumerator
  - deprecated MFXVideoCORE\_SetBufferAllocator() function, mfxBufferAllocator structure
  - deprecated fields:
    - mfxCodingOption.EndOfSequence
    - mfxCodingOption.EndOfStream
    - mfxExtCodingOption2.ExtBRC
  - deprecated MFX\_EXTBUFF\_VPP\_PICTSTRUCT\_DETECTION enumerator
- Only the following features among those introduced in API 1.19 are supported:
  - MFXVideoCORE\_QueryPlatform() function
  - MFXVideoENC\_GetVideoParam()function
  - mfxPlatform structure
  - mfxPayload.CtrlFlags field
  - mfxExtVPPScaling structure
  - mfxStatus (\*QueryPlatform) (mfxHDL pthis, mfxPlatform \*platform) for USER interfaces
  - mfxStatus MFXVideoUSER\_GetPlugin(mfxSession session, mfxUtype, mfxPlugin \*par)
- Only the following features among those introduced in API 1.20 are supported:
  - PicStruct::MFX\_PICSTRUCT\_FIELD\_SINGLE
  - PicStruct::MFX\_PICSTRUCT\_FIELD\_TOP
  - PicStruct::MFX\_PICSTRUCT\_FIELD\_BOTTOM
  - PicStruct::MFX\_PICSTRUCT\_FIELD\_PAIRED\_PREV
  - PicStruct::MFX\_PICSTRUCT\_FIELD\_PAIRED\_NEXT
- Only the following features among those introduced in API 1.21 are supported:
  - mfxExtCodingOption3.BRCPanicMode for H.264 and MPEG-2 encoders
- Only the following features among those introduced in API 1.22 and 1.23 are supported:
  - mfxExtCodingOption3.RepartitionCheckEnable for H.264 encoder

- Only the following features among those introduced in API 1.25 are supported:
  - `mfxExtCodingOption3.EnableNalUnitType` and `mfxEncodeCtrl.MfxNalUnitType` for HEVC encoder
  - `mfxExtMultiFrameParam` and `mfxExtMultiFrameControl`
  - `mfxExtBRC`, `mfxBRCFrameParam`, `mfxBRCFrameCtrl`, `mfxBRCFrameStatus`

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

- **Performance:**

- [26362] 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.
- [27023] HEVC decoder and encoder tasks can be executed only on VDBOX0 and UMD/KMD will schedule such tasks accordingly.
- To get better performance results and performance/resources tradeoff on the N:N multi-transcoding scenarios with the significant number of parallel sessions (N>4) it is recommended to consider synchronous pipeline implementations and initialize components with `mfxVideoParam::AsyncDepth=1`. In case of small number of parallel sessions asynchronous pipelines may give better results, it is recommended to consider `mfxVideoParam::AsyncDepth=2,3`.
- Per batch buffer Video Coding Engines (VDBOX) balancing has known limitations due to load balancing happening in the User Mode Driver on the moment of batch buffers construction:
  - Performance may suffer (be worse comparing to previous balancing strategy) if tasks are submitted in advance asynchronously (`AsyncDepth>1`)
  - Performance may suffer (be worse comparing to previous balancing strategy) if component's VDBOX tasks are dependent on other tasks. That's the case for Legacy (non-FEI) encoders which submit 2 tasks for each frame: ENC (Render Engine) and dependent PAK (Video Coding Engine).  
The impact of the above limitations is lower as the flow of balanced tasks is bigger. Limitations could be observed if just few (2-5) components are being balanced. Limitations should not be visible on the big number of balanced components (>~10).
- [34906] An average performance of multisession (3-4 sessions) transcoding from HEVC 4K progressive to AVC 4K progressive decreased by 7% FPS in comparison to Media Server Studio 2017 R2.
- [37705,37706] VC1 workloads may underperform in some scenarios where `AsyncDepth>1`. It is recommended to initialize components with `mfxVideoParam::AsyncDepth=1`. In case of small number of parallel sessions asynchronous pipelines may give better results.

- **Multi-Frame encode:**

- Only H.264 encode implemented.
- Only Intel Xeon E3-1500 v5 Family supported.
- Performance can be worse than usual single-frame in next known cases:
  - 4K resolution encoding.
  - N:N with HEVC decode present in pipeline.
  - cases where framerates differs significantly like 30 and 60 for different streams.

- Following functionality is not supported with multi-frame encode operation and when set, multiframe will be disabled:
  - When number of slices is controlled by parameters *NumSlice*, *NumSliceI*, *NumSliceP*, *NumSliceB*.
  - 'intra refresh' parameters are not supported.
- MFX\_MF\_MANUAL Multi-Frame submission mode rarely validated and can operate not per design.
- Flexible Encode Infrastructure is not supported for multi-frame operation, proper SDK behavior is not validated.
- Supported MaxNumFrames is 3 but only 2 will be used for next cases: *EnableMBQP*, *EnableMAD*, *EnableMBForceIntra*, *MBDisableSkipMap*.
- **HEVC decode:**
  - Function *GetPayload()* returns SEI with inserted emulation prevention bytes. If these SEI messages will be sent as Payloads to SDK HEVC encoder, emulation bytes will be duplicated.
  - When more than one frame is passed to *DecodeFrameAsync* in same input buffer, and external timestamp is set for this input, decoder will use provided timestamp to calculate PTS for 1st frame in the buffer. Decoder will not request additional timestamps for rest of frames in the buffer, and output PTS for these frames will be incorrect.
  - [27066] "Corrupted" flag in *mfxFrameData* is set by Decoder based on last slice only. I.e., if there are corruptions in previous slices, and no corruptions in last slice, "Corrupted" flag will be set by Decoder to 0.
  - If Decoder is called with flag *MFX\_BITSTREAM\_EOS* set and incomplete (or absent frame), it will return *MFX\_ERR\_MORE\_DATA* status instead of returning the buffered frames.
  - Decoder doesn't properly support API functions *GetDecodeStat()* and *SetSkipMode()*. Both functions always return status *MFX\_ERR\_NONE*. *GetDecodeStat()* doesn't modify output *mfxDecodeStat* structure. *SetSkipMode()* function doesn't affect decoder operation.
  - [36639, 37214] GPU hang can occur on streams with dependent slices if slice header syntax elements of corresponding independent slices are corrupted.
- **HEVC encode:**
  - Only following parameters from SDK API extended buffers are supported by Encoder:
    - *mfxExtHEVCParam*
      - *PicWidthInLumaSamples*
      - *PicHeightInLumaSamples*
    - *mfxExtCodingOption*
      - *VuiNalHrdParameters*
      - *NalHrdConformance*
      - *AUDelimiter*
    - *mfxExtCodingOption2*
      - *IntRefType*
      - *IntRefCycleSize*
      - *IntRefQPDelta*
      - *BRefType*
      - *NumMbPerSlice*
      - *DisableDeblockingIdc*
      - *RepeatPPS*
      - *SkipFrame*
    - *mfxExtCodingOption3*
      - *PRefType*
      - *IntRefCycleDist*
      - *EnableQPOffset*
      - *QPOffset*
      - *NumRefActiveP*

- NumRefActiveBL0
  - NumRefActiveBL1
  - EnableNalUnitType
- mfxExtEncoderCapability
  - MBPerSec

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

- Encoder cannot accept Prefix SEI payloads with total size over 256 bytes. As a result of bigger payload error `MFX_ERR_DEVICE_FAILED` will be returned from `SyncOperation()`.
- Default GOP structure is different for CQP (7 B-frame B-pyramid) and CBR/VBR (frame B-pyramid).
- Encoder doesn't support Intra Refresh.
- Support of Bitrate Control is limited. There could be quality problems for encoding with CBR/VBR (e.g. low coding quality for CBR and B-pyramid). ICQ, VCM, QVBR BRC methods aren't supported. LookAhead BRC modes aren't supported.
- Encoder doesn't support "Region Of Interest" feature.
- [31459] Encoder may produce non-bit exact streams from run-to-run. Run to run difference doesn't affect visual quality
- [29260] MBBRC option is not supported. For CBR/VBR, MBBRC is always enabled for TU 1-2, and always disabled for TU 3 - 7.
- Parameter `NumMBPerSlice` is treated as number of coding units of maximum size supported by underlying HW. For Skylake it's 32x32. May vary for future platforms.
- Encoder doesn't support insertion of HRD information to coded stream. At the same time Encoder tries to keep HRD conformance for given BRC and HRD parameters. E.g. padding will be inserted for CBR BRC mode.
- Padding for CBR is done by series of trailing\_zero\_8bits syntax element.
- For TargetUsage 7 encoder produces on-par or worse RD quality than AVC encoder for non-linear types of motion (e.g. rotation, shaking). Work around is to set multiple references using `NumRefActiveP`, `NumRefActiveBL0`, `NumRefActiveBL1`.
- If application needs to disable deblocking, it needs to always set respective value of `DisableDeblockingIdc` if extended buffer `MfxExtCodingOption2` is sent to encoder in runtime. If value isn't set (equals to 0), and buffer is sent to `EncodeFrameAsync`, zero value will be applied to current frame, and deblocking will be turned on.
- Attempt to create HEVC encoder on Broadwell will cause segmentation fault in `Init()`, `Query()` and `QueryIOSurf()` functions.
- [27145, 27477] Encoder does not support resolution 16x16. `MFX_ERR_DEVICE_FAILED` will be returned on attempt to create encoder with such small resolutions from the SDK `Init()` function. Meanwhile `Query()` function will return `MFX_ERR_NONE`.
- Values of `PicWidthInLumaSamples`, `PicHeightInLumaSamples` must be multiple of 16.
- For very complex content `BufferSizeInKB` reported by `GetVideoParam()` may not be enough to hold coded bitstream. If it happen, HEVC encoder will return `MFX_ERR_NOT_ENOUGH_BUFFER` in runtime.
- If in `EncodedOrder` `mfxEncodeCtrl.FrameType` is set to `MFX_FRAMETYPE_P` | `MFX_FRAMETYPE_REF` and `mfxExtHEVCRefLists` contains non-empty L1 list, output frame will contain visual artifacts (as well as all frames which use this frame for reference).
- If in `EncodedOrder` application sends to Encoder equal `FrameOrder` values for subsequent frames (e.g. all zeroes, or some other value), Encoder may hang inside `EncodeFrameAsync` call.

- Encoder cannot change frame rate via Reset() call (new value will be ignored). But if frame rate is changed along with bitrate by same Reset() call, encoder will apply new value.
- [38510] Encoder cannot change bit rate via Reset() call for user-defined BRC if HRD conformance is enabled. *MFX\_ERR\_INCOMPATIBLE\_VIDEO\_PARAM* is returned.
- Encoder may return *MFX\_ERR\_DEVICE\_FAILED* in case of very high bitrates or low QP when compression rate is about 1. Is it required to call reset and do re-encode with lower bitrate or higher QP.
- For CQP encoding of very complex content with low QP *BufferSizeInKB* reported by *GetVideoParam()* may not be enough to hold coded bitstream. If it happens, HEVC encoder returns *MFX\_ERR\_NOT\_ENOUGH\_BUFFER* from SyncOperation. Application may workaround it by allocation of bigger buffer than SDK reported in *BufferSizeInKB*. Following value is enough: \* 3.
- [18046] Encoder may generate too high bitrate in CBR mode.
- Per-LCU QP mode has the following limitations:
  - the feature is applicable only for CQP BRC.
  - QP values can be set in range 0-51.
  - only *MFX\_MBQP\_MODE\_QP\_VALUE* is supported.
  - the feature supports only LCU 32x32. However, application should allocate memory for QP map for blocks 16x16. Please see an example for 720x480 case:

0	1	2	3	4	5	6	7	...	44	
45	46	47	48	49	50	51	52	...	88	
89	90	91	92	93						
...	...		720x480						...	
									1349	

where 0...1349 are indices of a 16x16 blocks in a linear memory array of the QP map.

QP values in blocks 0, 2, 4, 6, ..., 89, 91, 93, ... are used by encoder. The rest blocks are ignored.

- SkipFrame feature has the following limitations:
  - works only with CQP BRC mode.
  - *MFX\_SKIPFRAME\_NO\_SKIP*, *MFX\_SKIPFRAME\_INSERT\_DUMMY*, *MFX\_SKIPFRAME\_INSERT\_NOTHING* are supported.
  - 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.
- [37552] In CQP mode encoder doesn't accept zero QP values in *mfxEncodeCtrl.QP*  
**Workaround:** Use QP=1.
- In CQP mode encoder ignores zero initialization QPI/QPP/QPB values. Internal default QP is used instead.  
**Workaround:** Use QP=1.
- [37754/37756] Encoder ignores *MFX\_GOP\_CLOSED*/*MFX\_GOP\_STRICT* flags set in *GopOptFlag*.
- [38139] Encoder doesn't check input frames *FrameOrder* values in the encoded order mode. If frame numbers in display order (before an application reorders them) don't monotonically increase, encoder doesn't return errors and produces

- corrupted output.
- Maximal number of supported active references is 3 for L0 list and 1 for L1 list.
- **H.264 decode:**
  - The H.264 decoder may leave Corrupted flag as 0 in case of minor corruption in macroblock bitstream data.
  - Decoder returns *MFX\_ERR\_UNSUPPORTED* for streams which cannot be processed by hardware, software fallback was removed.
  - For decoders it's not guaranteed that GPU hang will be timely reported from *SyncOperation* (there could be several frames delay). *MFX\_ERR\_GPU\_HANG* will be timely reported from *DecodeFrameAsync* call.
  - Function *GetPayload()* returns SEI with inserted emulation prevention bytes. If these SEI messages will be sent as Payloads to SDK HEVC encoder, emulation bytes will be duplicated.
  - When more than one frame is passed to *DecodeFrameAsync* in same input buffer, and external timestamp is set for this input, decoder will use provided timestamp to calculate PTS for 1st frame in the buffer. Decoder will not request additional timestamps for rest of frames in the buffer, and output PTS for these frames will be incorrect.
- **H.264 encode:**
  - Reporting of per-frame QP via *mfxExtAVCEncodedFrameInfo* isn't supported by Encoder (zero QPs are always returned).
  - Encoder doesn't support use of *MaxFrameSize* and *MaxSliceSize* together. If *MaxSliceSize* is set, *MaxFrameSize* is ignored.
  - Reset function isn't supported for LookAhead BRC modes (except *MaxSliceSize* mode). Reset returns *MFX\_ERR\_INVALID\_VIDEO\_PARAM* for LA BRC.
  - LookAhead BRC modes (including *MaxSliceSize* mode) don't support CAVLC coding. CAVLC will be switched to CABAC for LA BRC.
  - Downscale factor 2 of LookAhead BRC modes may produce better objective coding quality than scale factor 1
  - To change encoding parameters on the fly with *Reset()* function without IDR insertion application should drain all the buffered surfaces from encoder. Otherwise encoder may demonstrate undefined behavior after *Reset*.
  - Call of *Reset* which starts new sequence (inserts IDR) will drop HRD conformance over the inserted IDR (CPB removal counter will be set to 0 in the IDR Picture Timing SEI).
  - Encoder may produce non-bit exact streams from run to run on Broadwell and Skylake. Run to run difference doesn't affect visual quality.
  - Chroma artefacts are possible at very low bitrates (when QP is close to 51).
  - 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*.
  - [32675] Encoder may not insert PCM macroblocks when required.
  - Usual Look Ahead BRC may generate non HRD-compliant streams.
  - Look Ahead BRC (*MFX\_RATECONTROL\_LA\_HRD*) may not give quality gain for B-pyramid case (or give very little gain in comparison with regular B-frame configuration)
  - Careful memory/resource planning is needed when using Look Ahead BRC due to storage of pre-analyzed frames. 1:N and N:N transcoding use cases are especially demanding for memory.
  - When external Look Ahead BRC is used in transcoding pipeline which includes Frame Rate conversion (FRC), FRC in the pipeline should take place before

- external Look Ahead. Otherwise encoder may return error *MFX\_ERR\_UNDEFINED\_BEHAVIOR* from *EncodeFrameAsync* calls.
- MBBRC option is enabled by default on lower target usages but can be switched off. Exact implementation details are hidden and may change with time and between platforms, so using *Query* function to retrieve actual support is strongly recommended.
  - SkipFrame feature has the following limitations:
    - If GOP has only P frames, arbitrary P can be skipped. When skipped, it is made non-reference.
    - If GOP has B frames, only non-reference B can be skipped.
    - [40205] Encoder may produce stream with HRD violation if application skips frames using SkipFrame interface.
    - [28500] Encoder incorrectly calculates *cpb\_removal\_delays* for cases when several frames are dropped using *MFX\_SKIPFRAME\_BRC\_ONLY*. Delay is always incremented, but it needs to be increased by value of *mfxEncodeCtrl.SkipFrame*. As a result stream falls to HRD underflow.
  - MBQP mode is applicable only for CQP BRC and can be set in value range 1-51.
  - [35082] Encoder doesn't release Locked counters of input surfaces in *Close()* function.
  - Encoder doesn't support WeightedPrediction and FadeDetection features.
  - For very complex content *BufferSizeInKB* reported by *GetVideoParam()* may not be enough to hold coded bitstream. If it happens AVC encoder will cut bytes that don't fit to output buffer.
  - If application needs to disable deblocking, it needs to always set respective value of *DisableDeblockingIdc* if extended buffer *MfxExtCodingOption2* is sent to encoder in runtime. If value isn't set (equals to 0), and buffer is sent to *EncodeFrameAsync*, zero value will be applied to current frame, and deblocking will be turned on.
  - [27936] Turning ON Trellis for B-frames may slightly worsen objective quality of coded streams in terms of BDPSNR.
  - Encoder cannot change frame rate via *Reset()* call (new value will be ignored). But if frame rate is changed along with bitrate with same *Reset()* call, encoder will apply new value.
  - Encoder cannot change bit rate via *Reset()* call for software BRC if HRD conformance is enabled. *MFX\_ERR\_INCOMPATIBLE\_VIDEO\_PARAM* is returned.
  - Encoder cannot accept SEI payloads with total size over ~1130 bytes (the exact number may slightly vary depending on Encoder configuration). As a result of bigger payload error *MFX\_ERR\_DEVICE\_FAILED* will be returned from *SyncOperation()*.
  - Parameters *MaxQP* and *MaxFrameSize* may contradict each other. If for particular frame *MaxQP* doesn't give BRC to reach *MaxFrameSize*, *MaxQP* is obeyed, and *MaxFrameSize* is violated.
  - Encoder may produce significantly higher bitrates than requested when coding with frequent alternation of progressive/interlaced picture structure (e.g. about 40% bigger bitrate when picture structure is alternated for every other frame). Behavior in such stress conditions may significantly differ between current release and previous release (MSS2016 R1) (e.g. resulting stream may have tenth of percent of BDRATE difference with either better or worse quality)
  - [27539] When I/P field pair is encoded at the beginning of new GOP, P field in the pair may use frames from previous GOP as reference. So I/P field pairs can't be used as clean random access points, unlike IDR/P field pairs.
  - Parameters that are part of PPS header in AVC standard (e.g. weighted prediction) can't be correctly applied on per-frame level if RepeatPPS option is set to OFF during encoder initialization.
  - [28660] Encoder implements "force-to-skip" functionality: Inter frame is coded w/o coefficients and motion vectors if there is a risk of HRD underflow. But this feature doesn't guarantee HRD compliant encoding. In rare cases for very low

- bitrates and very complex content HRD underflow may happen (e.g. 500 kbps 1080p and white noise content). It's actual for both Skylake and Broadwell.
- [28667, 31597] Rolling Intra refresh with MB-rows doesn't work correctly - all MBs below current Intra region in the frame are forced to Intra as well. Intra refresh with MB-columns could be used as work around.
  - Encoder produces different results for synchronous and asynchronous pipelines for TFF interlace encoding with TU7 on Broadwell. Visual quality isn't affected by this difference.
  - [27161] For encoding with temporal layers together with multislice, only first slice in a picture has SVC prefix NALu prior to slice header (there are number of prefix NALus equal to number of slices).
  - [27162] On Broadwell encoder may produce visual quality drops when motion/scene change comes after long simple static scene. A example is screen capture when window is popping up after several seconds of still picture.
  - Encoder may produce different output if cases when size of input surface is bigger than frame resolution, and when size of surface equal to frame resolution. Difference doesn't affect objective quality, and isn't visible by eye.
  - When number of slices is controlled by parameters *NumSlice*, *NumSliceI*, *NumSliceP*, *NumSliceB*, resulting slices are always aligned to MB-row. In addition all slices have equal sizes (except last one). Such alignment may cause change of slice number set by application. Application should use *NumMBPerSlice* parameter to get slices of arbitrary MB size.
  - Following restrictions are applied to reference lists set via *mfxExtAVCRefLists* for interlaced encoding. List should start from reference field with same polarity as current one. Fields in reference lists should alternate while it's possible to pick field of alternative parity. If listed restrictions are violated, coded field may contain visual artifacts, or be encoded with Intra macroblocks only.
  - Enabling or disabling of Rolling Intra Refresh by *Reset()* call will lead to insertion of IDR. E.g. if Encoder is initialized with *IntRefType* = 0 (disabled refresh), and *IntRefType* = 1 is sent to *Reset* call, IDR will be inserted right after *Reset*, and only then first Intra refresh cycle will be started.
  - For CQP encoding of very complex content with low QP *BufferSizeInKB* reported by *GetVideoParam()* may not be enough to hold coded bitstream. If it happens, AVC encoder truncates coded bitstream to fit to the buffer provided by application and returns *MF\_X\_ERR\_NONE* from respective *SyncOperation* call. Application may workaround it by allocation of bigger buffer than SDK reported in *BufferSizeInKB*. Following value is enough: \* 3.
  - Encoder fills *mfxExtAVCEncodedFrameInfo::SecondFieldOffset* correctly only if application attaches only 1 such buffer for the case of interlaced content.
  - [31266] Visual artifacts may occur when encoder tries to satisfy too small *MaxFrameSize* video parameter.
  - [27076] Encoder on Broadwell may produce worse objective quality for interlace encoding using B-pyramid. Average quality difference on big stream set is about 0.5% PSNR BDRATE.
  - Encoder doesn't support "Region Of Interest" feature.
  - [34037] On 4K content Look Ahead BRC may produce worse quality with *MF\_X\_TARGETUSAGE\_BEST\_QUALITY* than with *MF\_X\_TARGETUSAGE\_BALANCED* / *MF\_X\_TARGETUSAGE\_SPEED*.  
**Workaround:** Use 2x2 downscaling to work around the issue.
  - [35309] In *EncodedOrder* mode AVC encoder always returns status *MF\_X\_ERR\_MORE\_DATA* for *EncodeFrameAsync* calls with zero pointer to *mfxFrameSurface1* structure. It means that when *EncodedOrder* mode is used together with asynchronous encoding (*AsyncDepth* > 1), buffered *AsyncDepth* frames cannot be retrieved from encoder at the end of encoding by mechanism described in Media SDK specification.
  - [35288] On SKL frequent (several time per second) BRC resets using VBR interlace encode results in low bitrate.

- [38227] PSNR drop (without coding artifacts) is observed with dynamic *MaxFrameSize* parameter on screen content encode. It affects one frame when a complex scene is changed. On the other hand, an improvement in encoding quality is observed on the rest frames.
- [37925] Encoder in CQP mode doesn't insert CABAC trailing bits.
- [37721] Usage of *Query()* function with *mfxExtEncoderCapability* extended buffers may cause segmentation faults.
- [34742] Enbdecoder doesn't support *AdaptiveMaxFrameSize*.
- **MPEG-2 decode:**
  - Decoder does not support bitstreams with resolution bigger than 2048x2048. *MFXVideoDECODE\_Init* returns *MFX\_ERR\_UNSUPPORTED* on such bitstreams.
  - Decoder does not support MPEG-1 bitstreams. It is interpreted as corrupted MPEG-2 bitstream. *MFXVideoDECODE\_Init* returns *MFX\_ERR\_NONE* and *MFXVideoDECODE\_DeclareFrameAsync* returns *MFX\_ERR\_MORE\_DATA* until valid MPEG-2 bitstream is found.
  - Sequence headers are skipped if resolution exceeds maximum supported values (2048x2048) or level/chroma are invalid. It affects *MFXVideoDECODE\_DeclareFrameAsync*.
  - For decoders it's not guaranteed that GPU hang will be timely reported from *SyncOperation* (there could be several frames delay). *MFX\_ERR\_GPU\_HANG* will be timely reported from *DecodeFrameAsync* call.
  - If incoming bitstream is corrupted decoder may skip corrupted slices or entire frames. Usually corruptions in bitstream does not influence on decoding of next frames in bitstream that are not corrupted. It is recommended to reset decoder once corrupted frame occur to mitigate the issue.
  - Decoder returns *MFX\_ERR\_UNSUPPORTED* for streams which cannot be processed by hardware, software fallback was removed.
  - [33394] Decoder may return *MFX\_ERR\_NONE* with a zero sync point on corrupted streams. Application should explicitly check for non-zero sync point to get decoded valid and skip frames with zero sync points.
  - [36208] There is a performance regression up to ~15% on streams with one or several B frames. The issue affects performance only on one channel cases, multisession pipelines are not affected. The issue doesn't affect density.
  - [28713] Decoder may leave Corrupted flag as 0 in case of minor corruption in macroblock bitstream data.
- **MPEG-2 encode:**
  - [36085, 35800] Encoder may produce non-bit exact streams. Run to run difference doesn't affect visual quality.
  - [28337] The MPEG-2 encoder may produce output that under-runs the MPEG-2 video buffer verifier model (Vbv) on some streams. We suggest the following guideline of the parameter values to be followed to keep Vbv compliance.
    - MPEG2 buffer usage is mainly restricted by the number of bits used for I frame. The minimum size of each 16x16 blocks of intra frame at highest QP is about 50 bits. The minimum initial buffer fullness (*InitVbvBufferFullnessInBit*) should be at least twice the size of the initial I frame, and the minimum buffer size (*vbv\_buffer\_size*) should be twice of the initial buffer fullness (4 times of the initial I frame).
  - MBQP mode is applicable only for CQP BRC and can be set in value range 1-122.
  - SkipFrame feature has the following limitations:
    - works only with CQP BRC mode
    - only *MFX\_SKIPFRAME\_INSERT\_DUMMY* is supported
    - If GOP has only P frames, arbitrary P can be skipped. When skipped, it is made non-reference.
  - Software implementation of MPEG-2 Encoder may erroneously insert a duplicated field in interlace field encode mode on specific content.
  - [26996] 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. Sometimes Encoder Bitrate Control acts too conservative and produces artifacts caused by so-called panic mode.

**Workaround:** Disable Encoder Bitrate Control panic mode. This may increase visual quality on low bitrates at the cost of potential VBV compliance violation.

- Resetting MPEG-2 Encoder with new aspect ratio may return *MXF\_ERR\_INCOMPATIBLE\_VIDEO\_PARAM*, workaround - explicitly close and re-initialize encoder.
- Contrary to SDK Reference Manual (mediasdkman.pdf) MPEG-2 Encoder may allocate surfaces on Reset call.
- Encoder doesn't return errors on attempts to initialize it with some unsupported parameters (e.g. WeightedPrediction, FadeDetection and so on). Encoder ignores such parameters and returns *MXF\_ERR\_NONE* from Query/Init/Reset functions.
- Once Encoder is initialized with some resolution, Encoder doesn't return error status on attempt to submit frame of bigger resolution to *EncodeFrameAsync*. Frame is accepted, consequence may be corrupted output, segmentation fault or some other undefined behavior.
- [31656] On some content encoder may generate stream with lesser bitrate than requested.
- [31893] On complex content like interlace stream with fast motion encoded to progressive frame PSNR quality may vary frame to frame by more than 10 dB from average.
- [38895] In software implementation of MPEG2 encoder encoding quality depends on a number of executions threads. Bigger number of threads improves performance, but decreases quality. The number of executions threads depends on a number of CPU cores.
- **JPEG/MJPEG decode and encode support only the below feature set:**
  - Baseline mode only
    - DCT based
    - 8 - bit samples
    - sequential
    - loadable 2 AC and 2 DC Huffman tables
    - 2 loadable quantization matrixes
    - interleaved and non-interleaved scans
    - single and multiple scans
  - No extended, lossless and hierarchical modes
    - no 12-bit samples
    - no progressive
    - no arithmetic coding
    - no 4 AC and 4 DC Huffman tables
  - JPEG/MJPEG hardware accelerated decoder supports resolutions <= 8192. In case of bigger resolution fallback to SW will be notified via *MXF\_WRN\_PARTIAL\_ACCELERATION* from *Init/Query/QueryIOSurf* functions.
  - Decoder supports YUV2 output format and multi-scan pictures only by SW fallback.
  - Hardware accelerated encoder (available on Skylake only) supports only NV12, RGB32 input color formats with SW fallback in other cases.
  - MJPEG Encoder doesn't encode correctly nv12 monochrome format input.
  - [33278] HW decoder doesn't support 4:1:1, 4:4:4 chroma subsamplings (chroma subsamplings 4:2:0, 4:2:2 are supported).
  - Decoder with rotation doesn't support rgb32 and yuy2 formats, it supports nv12 pixel format only.
  - Decoder doesn't properly support output surfaces pool bigger than 128. *MXF\_ERR\_UNSUPPORTED* may be returned on attempt to use such big surface pool for decoder output.

- Decoder doesn't support GPU hang reporting.
- Decoder may produce non bitexact output from run to run without visual artifacts.
- [17872] Decoder rotate feature may leave green areas on resulting image.
- [32293] MJPEG rotate feature may trigger GPU hang on Broadwell.
- [35437] Decoder and encoder don't support opaque memory. Usage of opaque memory may cause segmentation fault.
- **VC1 decoder:**
  - Decoder may cause GPU hangs and return status `MFX_ERR_UNDEFINED_BEHAVIOR` during decoding of corrupted content.
  - Decoder doesn't support GPU hang reporting.
- **VP8 decoder:**
  - Decoder doesn't support GPU hang reporting.
  - [34602] GPU hang can occur on streams with corrupted partition sizes.
- **VPP:**
  - Multiple VPP filters being combined in one session may produce output that is not bit-exact with the output from the same VPP filters that are split by separate sessions, but the difference does not affect visual quality.
  - Field copy processing cannot be used with any other VPP filters including resize and color conversion. In case field processing is requested, all other VPP filters are skipped without error/warning messages.
  - [29614] Frames with interlaced content must have CropH multiple of 4. Otherwise, VPP may produce color artefacts on the bottom lines.
  - VPP interpolated FRC was deprecated and replaced by simple arbitrary FRC. Setting FRC algorithm to `MFX_FRCALGM_FRAME_INTERPOLATION` in `mfxExtVPPFrameRateConversion` extended buffer will cause fallback to `MFX_FRCALGM_PRESERVE_TIMESTAMP` algorithm. No errors/warning are returning by SDK in this case.
  - `MFXVideoVPP_Reset` could return error if additional memory allocation is required based on provided video parameters (changed type of color or frame rate conversion for example). The application should close VPP component and then re-initialize it in this case. See `MFXVideoVPP_Reset` in SDK API Reference Manual for more details about possible return statuses.
  - Multiple RGB4 surfaces with `PixelAlphaEnable` enabled could give poor visual quality after composition on the same area with overlapping.
  - When composition is used for 8 or more channels AND at least one channel has `GlobalAlphaEnable=true` AND first channel has `GlobalAlphaEnable=false` => first channel is composed with artifacts looking like certain global alpha value was applied to it despite `GlobalAlphaEnable` is false.  
**Workaround:** For the first stream, replace `GlobalAlphaEnable=false` with `GlobalAlphaEnable=true` plus `GlobalAlpha=255`. This combination is visually equal to `GlobalAlphaEnable=false` and works correctly.
  - De-interlacing is supported for NV12, YUY2 formats only.
  - `MFX_DEINTERLACING_ADI` can produce color artifacts and out of order frames after scene change. `MFX_DEINTERLACING_ADI_SCD` should be used.
  - `MFX_DEINTERLACING_ADI_SCD` is handling frame with scene change and two or four frames after with BOB algorithm producing frame from the second field (from bottom in case of TFF and from top in case of BFF). Number of frames which is processed by BOB depends on deinterlacing mode: two for 30i->30p and four for 30i->60p.
  - [35025] `MFX_DEINTERLACING_ADI_SCD` uses previous and current input frame to generate current output. The output timing may be offset by a field, which may be more visible after a scene change.
  - [27242] `MFX_DEINTERLACING_ADI` and `MFX_DEINTERLACING_ADI_SCD` may produce color artefacts in case of harmonic motion (repeated pattern and motion magnitude is the same as the periodic of repeated pattern).
  - VPP doesn't support standalone scene change detection filter and ignores

*MFX\_EXTBUFF\_VPP\_SCENE\_CHANGE, MFX\_EXTBUFF\_VPP\_SCENE\_ANALYSIS* extended buffers.

- Fields weaving and fields splitting cannot be used with any other VPP filters including resize and color conversion. In case fields weaving or fields splitting is requested together with any other VPP filters, VPP initialization shall fail.
  - [29431] Per-pixel alpha blending doesn't work with system-memory mode. Use video memory to work around the issue.
  - De-interlacing of mixed picture structure requires that the application or the decoder provides the correct picture structure for each frame as input to Video Processing. This feature has been developed for processing H264 decoded streams with Picture Adaptive Frame/Field coding (PAFF) enabled. Issue may be found with other decoders and some ADI\_SCD corner cases.
- **Misc:**
- Only the following functionality from software implementation of the SDK is fully validated and has production quality: VPP scaling and color conversion. All the rest (decoders, encoder, other VPP filters) is provided for demo purpose only and is not guaranteed to have production quality.
  - Due to specifics of GPU Copy implementation it is required to close/destroy SDK associated resources (including VADisplay and frame surfaces) only after MFXClose call.
  - Using system memory as input/output with *mfxInitParam::GPUCopy* set to *MFX\_GPUCOPY\_ON* has restriction for system memory allocation to be aligned at 64 bytes.
  - Encode quality may be different (non-bit exact) between CPU generations.
  - In case of executable compiled with -rdynamic or SDK loaded by application plugin, Dispatcher linker symbols would conflict with SDK symbols. To mitigate symbol conflict application should:
    - link against dispatch\_shared.a instead of libmfx.a
    - define MFX\_DISPATCHER\_EXPOSED\_PREFIX before any SDK includes
  - General user plugin should not expect more than four input and four output surfaces in MFXVideoUSER\_ProcessFrameAsync call. User should not provide more than four parameters to mfxHDL \*in, mfxU32 in\_num, mfxHDL \*out, mfxU32 out\_num.
  - [17856] Both software and hardware SDK implementations require at least two logical cores to run correctly.
  - [38431] This release is not binary compatible with the media driver available at [GitHub](#) due to different versions of VA-API.

## Legal Information

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

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting [www.intel.com/design/literature.htm](http://www.intel.com/design/literature.htm).

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, Iris, Intel HD Graphics, Intel Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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

© Intel Corporation.

### 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

Copyright (C) 2012, 2013 Cisco Systems

All rights reserved.

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

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

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