Advanced Micro Devices

Advanced Media Framework – HEVC Video Encoder

Programming Guide

Disclaimer

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information.

Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

AMD, the AMD Arrow logo, ATI Radeon™, CrossFireX™, LiquidVR™, TrueAudio™ and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Windows™, Visual Studio and DirectX are trademark of Microsoft Corp.

Copyright Notice

Notice Regarding Standards. AMD does not provide a license or sublicense to any Intellectual Property Rights relating to any standards, including but not limited to any audio and/or video codec technologies such as MPEG-2, MPEG-4; AVC/H.264; HEVC/H.265; AAC decode/FFMPEG; AAC encode/FFMPEG; VC-1; and MP3 (collectively, the “Media Technologies”). For clarity, you will pay any royalties due for such third party technologies, which may include the Media Technologies that are owed as a result of AMD providing the Software to you.

MIT license

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.

Contents

1 Introduction

1.1 Scope

This document provides a complete description of the AMD Advanced Media Framework (AMF) Video Encoder Component. This component exposes the AMD Video Compression Engine, which provides hardware accelerated HEVC video encoding functionality.

Figure 1 provides a system overview of the AMF Video Encoder Component.

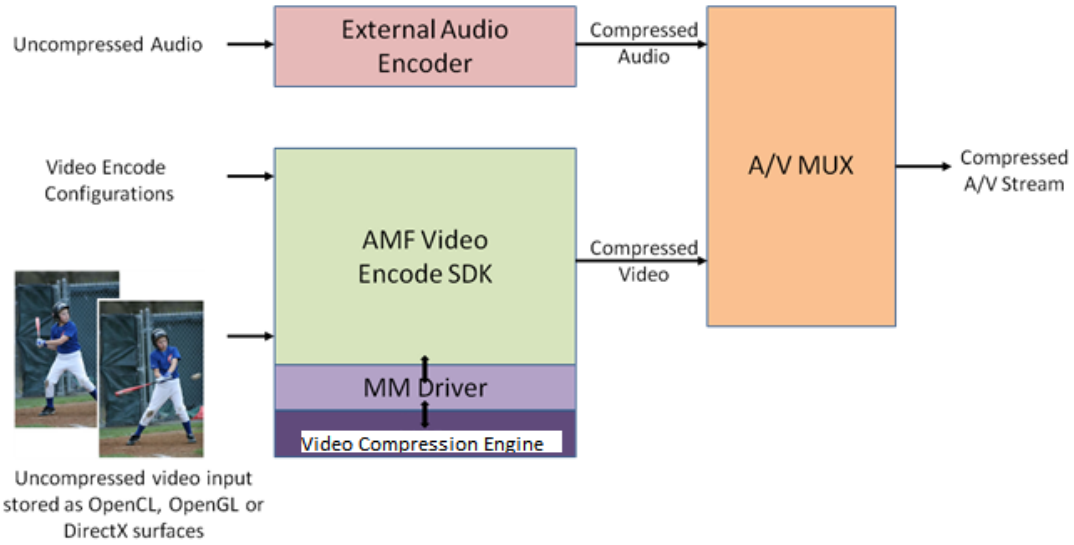

Figure 1 — System overview of the AMF Video Encode SDK

The AMF Video Encoder Component compresses RAW uncompressed video to an HEVC elementary bitstream.

The component does not provide a mechanism to handle audio compression, or stream multiplexing.

The component provides four different sets of pre-defined usages, which provide a convenient way for developers to configure the encoder to match the intended application use case. Advanced developers can also adjust encoding parameters to tailor the behavior to their specific application requirements.

HEVC encoding is currently supported on Windows only.

1.2 Pre-defined Encoder Usages

The following table provides a brief overview of the encoding usage modes that have been defined:

Usage Mode	Intended use-cases	Comments
Transcoding	Transcoding, video editing	Favor compression efficiency and throughput over latency.
Ultra-low latency	Video game streaming	Optimize for extremely low latency use cases (e.g. cap the number of bits per frame), to enable high-interactivity applications.
Low Latency	Video collaboration, remote desktop	Optimize for low latency scenarios, but allow occasional bitrate overshoots to preserve quality.
Webcam	Video conferencing	Optimize for a low-latency video conferencing scenario.
HQ	High quality mode	Optimize for best subjective video quality with possible loss of performance
HQLL	High quality low latency mode	Optimize for good quality with low latency

Note: User can override the default settings for these pre-defined usages in Table A-3. Default value of parameters.

2 AMF Video Encoder UVD/VCN-HEVC Component

The AMF Video Encoder HEVC component provides hardware accelerated HEVC encoding using AMD's IP.

To instantiate the AMF Video Encoder component, call the *AMFFactory::CreateComponent* method passing *AMFVideoEncoderHW_HEVC* component IDs defined in the */include/components/VideoEncoderHEVC.h* header.

2.1 Input Submission and Output Retrieval

The AMF Video Encoder component accepts *AMFSurface* objects as input and produces *AMFBuffer* objects for output.

2.2 Encode Parameters

Annex A provides the detailed description of encoding parameters (i.e., encoder properties) exposed by the Video Encoder HEVC component for the following four usages:

- Transcoding mode,
- Ultra-low latency mode,
- Low Latency mode, and
- Webcam mode.

All properties are accessed using the *AMFPropertyStorage* interface of the Encoder object.

2.2.1 Static Properties

Static properties (e.g., profile, tier, level, usage) must be defined before the `Init()` function is called, and will apply until the end of the encoding session.

2.2.2 Dynamic Properties

All dynamic properties have default values. Several properties can be changed subsequently and these changes will be flushed to encoder only before the next `Submit()` call.

2.2.3 Frame Per-Submission Properties

Per submission properties are applied on a per frame basis. They can be set optionally to force a certain behavior (e.g., force frame type to IDR) by updating the properties of the `AMFSurface` object that is passed through the `AMFComponent::Submit()` call.

2.2.4 ROI Feature

Region of importance (ROI) feature provides a way to specify the relative importance of the macroblocks in the video frame. Encoder will further adjust the bits allocation among code blocks based on the importance, on top of the base rate control decisions. More important blocks will be encoded with relatively better quality.

The ROI map can be attached to the input frame on a per frame basis. Currently, the ROI map can only use system memory. The ROI map includes the importance values of each 64x64 CTB, ranging from 0 to 10, stored in 32bit unsinged format. Refer to SimpleROI sample application for further implementation details.

2.2.5 Encoder Statistics Feedback

If an application sets the `AMF_VIDEO_ENCODER_HEVC_STATISTICS_FEEDBACK` flag on for an input picture, the encoder will feedback to the application statistics for this specific picture. After the encoding ends, the application can retrieve by name the specific statistic(s) it is interested in. The supported encoder statistics are listed in Table

A-4. This feature is supported by Radeon RX 5000 Series or newer GPUs as well as Ryzen 2000 U/H series or newer APUs.

2.2.6 Picture Transfer Mode

If an application enables AMF_VIDEO_ENCODER_HEVC_PICTURE_TRANSFER_MODE for a specific input picture, it can dump out the reconstructed picture after encoding and/or it can inject a picture to be used as the reference picture during the encoding. It is worth noting that reference picture injection is a feature that is intended for advanced algorithm testing and exploration. It needs to be used with care since the internal DPB in the current encoding session will be overridden by the injected reference picture(s). The reader can refer to SimpleFrameInjection sample application for further implementation details. This feature is supported by Radeon RX 5000 Series or newer GPUs as well as Ryzen 2000 U/H series or newer APUs.

2.2.7 SVC Properties

Scalable Video Coding (SVC) is enabled by setting AMF_VIDEO_ENCODER_HEVC_NUM_TEMPORAL_LAYERS to a value that is greater than 1. AMF_VIDEO_ENCODER_HEVC_NUM_TEMPORAL_LAYERS is a dynamic property and can be changed at any time during an encoding session. To ensure proper support, AMF_VIDEO_ENCODER_HEVC_MAX_NUM_TEMPORAL_LAYERS needs to be set before initializing the encoder to a value that is not smaller than the number of temporal layers. As an example, the maximum number of temporal layers shall be set to 4 if the number of temporal layers will be changed from 3 to 4 in an encoding session. The maximum number of temporal layers supported by the encoder can be queried from the encoder capabilities before initializing the encoder.

To define SVC parameters per layer, the following format must be used:

```
TL<Temporal_Layer_Number>.QL<Quality_Layer_Number>.<Parameter_name>
```

As an example with two temporal layers, to configure “Target bitrate” for the base/first temporal layer and first quality layer, the following parameter should be used:

```
“TL0.QL0.HevcTargetBitrate”
```

To configure “Target bitrate” for the second temporal layer and first quality layer, the following parameter should be used:

```
“TL1.QL0.HevcTargetBitrate”
```

When setting per layer parameters, the equivalent non-SVC layer parameters should not be set for the encoder otherwise the per layer configuration will be overwritten.

Remark: quality layers are not supported. “QL0” must be used for quality layers.

This feature is supported by Radeon RX 5000 Series or newer GPUs as well as Ryzen 2000 U/H series or newer APUs.

3 Sample Applications

The AMF Encoder Sample application show how to setup and use the AMF Video Encoder HEVC Component to encode video frames that are loaded from disk or rendered by the DirectX 3D engine.

3.1 List of Parameters

Sample applications support almost all visible encoder parameters (except PictureStructure, EndOfSequence, EndOfStream) and few additional parameters.

Additional parameters of TranscodeHW application:

Category	Name	Values	Description
Miscellaneous parameters	Frames	Number of frames to be encoded	Number of frames to render
	Codec	HEVC or H265	Specify codec type
	Input	File name, relative or absolute path	Input file with frames (AVC or HEVC)
	Output	File name, relative or absolute path	Output HEVC file for encoded data
	Engine	DX9, DX11	Specify Engine type
	AdapterID	Number	Index of GPU adapter

Additional parameters of VCEEncoderD3D application:

Category	Name	Values	Description
Miscellaneous parameters	Frames	Number of frames to be encoded	Number of frames to be encoded
	Codec	HEVC or H265	Specify codec type
	Output	File name, relative or absolute path	Output HEVC file for encoded data
	Render	DX9, DX9EX, DX11, OpenGL, OpenCL, etc.	Specify render type
	AdapterID	Number	Index of GPU adapter
	Windowmode	Flag (without any values)	Shows rendering window for D3D sample application
	FullScreen	Flag (without any values)	Enables full screen
	QueryInstanceCount	Flag (without any values)	If the flag is set, the number of independent VCE instances will be queried and printed
	UseInstance	0 to (number of instances -1)	If there are more than one VCE instances, allow to force which instance to use.

3.2 Command line example

3.2.1 Transcoding application (TranscodeHW.exe)

```
TranscodeHW.exe -input input.h264 -output out.h265 –codec HEVC -width 1280 -height 720 -Usage transcoding -RateControlMethod cbr -TargetBitrate 100000
```

This command transcodes H264 elementary stream to H.265 video. Encoder is created with “Transcoding” usage.

3.2.2 D3D application (VCEEncoderD3D.exe)

```
VCEEncoderD3D.exe -output VideoSample_1024x768.h265 –codec HEVC -width 1024 -height 768 -Usage transcoding -RateControlMethod cbr -TargetBitrate 500000 -frames 400
```

This command encodes 400 frames through D3D renderer and creates an output file with the encoded data. Encoder is created with “Transcoding” usage. Initial configuration sets bitrate to a value of 500kbits/sec.

Annex A: Encoding & frame parameters description

Table A-1. Encoder parameters

Category	Name (Prefix "AMF_VIDEO_ENCODER_HEVC_")	Values	Description
Encoder static parameters	USAGE	0, 1, 2, 3, 4, 5 (Transcoding, UltraLowLatency, LowLatency, Webcam, HQ(high quality), HQLL(high quality low latency)	Selects the AMF usage (see Section 1.2)
	INSTANCE_INDEX	0,1	Selects the encoder engine used for encoding
	PROFILE	Main	Selects the HEVC profile
	TIER	Main, High	Selects the HEVC tier
	PROFILE_LEVEL	1, 2, 2.1, 3, 3.1, 4, 4.1, 5, 5.1, 5.2, 6, 6.1, 6.2	Selects the HEVC Profile Level
	MAX_LTR_FRAMES	0 … 16	The number of long-term references controlled by the user. Remarks: • When == 0, the encoder may or may not use LTRs during encoding. • When > 0, the user has control over all LTR. • With user control of LTR, Intra-refresh features are not supported. • The actual maximum number of LTRs allowed depends on H.265 (HEVC) Annex A Table A-4 Level limits, which defines dependencies between the H.265 Level number, encoding resolution, and DPB size. The DPB size limit impacts the maximum number of LTR allowed.
	LTR_MODE	0, 1 (Reset unused, keep unused) Default = 0	Remove/keep unused LTRs not specified inside the LTR reference bitfield.
	MAX_NUM_REFRAMES	0 … 16	Maximum number of reference frames
	LOWLATENCY_MODE	True/False (On/Off); default is false	Enables low latency mode in the encoder
	PRE_ANALYSIS_ENABLE	bool; default = false	Enables the pre-analysis module. Some features require this one be enabled. Refer to AMF Video PreAnalysis API reference for more details.
	MAX_NUM_TEMPORAL_LAYERS	1 … Maximum number of temporal layers supported	Sets the maximum number of temporal layers. It shall not be exceeded by the number of temporal layers. The maximum number of temporal layers supported is determined by the corresponding encoder capability.
Encoder resolution parameters	FRAMESIZE	Width: 192 – 4096 Height: 128 – 2176	Frame width/Height in pixels, maximum value is hardware-specific, should be queried through *AMFCaps*
	ASPECT_RATIO	Default 1:1	Pixel aspect ratio
Encoder rate-control parameters	TARGET_BITRATE	>0	Sets the target bitrate, bit/s based on use case
	PEAK_BITRATE	>= TargetBitrate	Sets the peak bitrate

Category	Name (Prefix "AMF_VIDEO_ENCODER_HEVC_")	Values	Description
	RATE_CONTROL_METHOD	0, 1, 2, 3(CQP, VBR_LAT, VBR, CBR)	Selects the rate control method: • CQP – Constrained QP, • VBR_LAT - Latency Constrained VBR • VBR - Peak Constrained VBR, • CBR - Constant Bitrate Remarks: • When SVC encoding is enabled, some rate-control parameters can be configured differently for a particular SVC-layer. An SVC-layer is denoted by an index pair [SVC-Temporal Layer index][SVC-Quality Layer index]. E.g. The bitrate may be configured differently for SVC-layers [0][0] and [1][0]. • We restrict all SVC layers to have the same Rate Control method.
	RATE_CONTROL_SKIP_FRAME_ENABLE	True/False	Enables skip frame for rate control
	MIN_QP_I	0 – 51	Sets the minimum QP for I frame
	MAX_QP_I	0 – 51	Sets the maximum QP for I frame
	MIN_QP_P	0 – 51	Sets the minimum QP for P frame
	MAX_QP_P	0 – 51	Sets the maximum QP for P frame
	QP_I	0 – 51	Sets the constant QP for I-pictures. Remarks: Only available for CQP rate control method.
	QP_P	0 – 51	Sets the constant QP for P-pictures. Remarks: Only available for CQP rate control method.
	FRAMERATE	1*FrameRateDen … 60* FrameRateDen	Frame rate numerator/denominator
	VBV_BUFFER_SIZE	>0	Sets the VBV buffer size in bits based on use case
	INITIAL_VBV_BUFFER_FULLNESS	0 - 64	Sets the initial VBV buffer fullness
	ENFORCE_HRD	True/False	Disables/enables constraints on QP variation within a picture to meet HRD requirement(s)
	PREENCODE_ENABLE	True/False	Pre-analysis assisted rate control
	ENABLE_VBAQ	True/False	By default, disable VBAQ
	FILLER_DATA_ENABLE	True/False	Enable filler data for CBR usage
	HIGH_MOTION_QUALITY_BOOST_ENABLE	Bool; default = false	Enable high motion quality boost mode to pre-analyze the motion of the video and use this information to improve encoding.
Encoder picture-control parameters	MAX_AU_SIZE	0 – 100 000 000 bits	Maximum AU size in bits
	HEADER_INSERTION_MODE	NONE, GOP aligned, IDR aligned	Sets the headers insertion mode
	GOP_SIZE	0 ... 1000	The period to insert IDR/CRA in fixed size mode. 0 means only insert the first IDR/CRA (infinite GOP size)
	NUM_GOPS_PER_IDR	1 – 65535	Determines the frequency to insert IDR as start of a GOP. 0 means no IDR will be inserted except for the first picture in the sequence.
	DE_BLOCKING_FILTER_DISABLE	True/False	Disable/enable the de-blocking filter
	SLICES_PER_FRAME	1 - #CTBs per frame	Sets the number of slices per frame

Category	Name (Prefix "AMF_VIDEO_ENCODER_HEVC_")	Values	Description
	INTRA_REFRESH_NUM_CTBS_PER_SLOT	0 - #CTBs per frame	Sets the number of intra-refresh 64x64 coding-tree-blocks per slot
Encoder miscellaneous parameters	QUALITY_PRESET	Balanced, Quality, Speed	Selects the quality preset
	PICTURE_TRANSFER_MODE	PICTURE_TRANSFER_MODE_ON, PICTURE_TRANSFER_MODE_OFF	The application can turn on this flag for a specific input picture to allow dumping the reconstructed picture and/or injecting a reference picture
	QUERY_TIMEOUT	default = 0 (no wait);	Timeout for QueryOutput call in ms.
Encoder motion estimation parameters	MOTION_HALF_PIXEL	True/False	Turns on/off half-pixel motion estimation
	MOTION_QUARTERPIXEL	True/False	Turns on/off quarter-pixel motion estimation
Encoder color conversion parameters	COLOR_BIT_DEPTH	8, 10, 16	Sets the number of bits in each pixel's color component in the encoder's compressed output bitstream. Default is 8.
	INPUT_COLOR_PROFILE	UNKNOWN, 601, 709, 2020, JPEG, FULL_601, FULL_709, FULL_2020	Color profile of the input surface. SDR - Setting this parameter (COLOR_PROFILE) can fully describe a surface for SDR use case. HDR – For HDR use case the TRANSFER_CHARACTERISTIC, COLOR_PRIMARIES, and NOMINAL_RANGE parameters describe the surface. See ColorSpace.h for enumeration.
	INPUT_TRANSFER_CHARACTERISTIC	UNDEFINED, BT709, UNSPECIFIED, RESERVED, GAMMA22, GAMMA28, SMPTE170M, SMPTE240M, LINEAR, LOG, LOG_SQRT, IEC61966_2_4, BT1361_ECG, IEC61966_2_1, BT2020_10, BT2020_12, SMPTE2084, SMPTE428, ARIB_STD_B67	Characteristic transfer function of the input surface used to perform the mapping between linear light components (tristimulus values) and a nonlinear RGB signal. Used (alongside COLOR_PRIMARIES and NOMINAL_RANGE parameters) to describe surface in HDR use case. See ColorSpace.h for enumeration.
	INPUT_COLOR_PRIMARIES	UNDEFINED, BT709, UNSPECIFIED, RESERVED, BT470M, BT470BG, SMPTE170M, SMPTE240M, FILM, BT2020, SMPTE428, SMPTE431, SMPTE432, JEDEC_P22	Color space primaries for the input surface which are the maximum red, green, and blue value permitted within the color space. Used (alongside TRANSFER_CHARACTERISTIC and NOMINAL_RANGE parameters) to describe surface in HDR use case. See ColorSpace.h for enumeration.
	OUTPUT_COLOR_PROFILE	UNKNOWN, 601, 709, 2020, JPEG, FULL_601, FULL_709, FULL_2020	Color profile of the compressed output stream. SDR - Setting this parameter (COLOR_PROFILE) can fully describe a surface for SDR use case. HDR – For HDR use case the TRANSFER_CHARACTERISTIC, COLOR_PRIMARIES, and NOMINAL_RANGE parameters describe the surface. See ColorSpace.h for enumeration. Determines the optional VUI parameter "matrix_coefficients".

Category	Name (Prefix "AMF_VIDEO_ENCODER_HEVC_")	Values	Description
	OUTPUT_TRANSFER_CHARACTERISTIC	UNDEFINED, BT709, UNSPECIFIED, RESERVED, GAMMA22, GAMMA28, SMPTE170M, SMPTE240M, LINEAR, LOG, LOG_SQRT, IEC61966_2_4, BT1361_ECG, IEC61966_2_1, BT2020_10, BT2020_12, SMPTE2084, SMPTE428, ARIB_STD_B67	Characteristic transfer function of the compressed output stream used to perform the mapping between linear light components (tristimulus values) and a nonlinear RGB signal. Used (alongside COLOR_PRIMARIES and NOMINAL_RANGE parameters) to describe surface in HDR use case. See ColorSpace.h for enumeration.
	OUTPUT_COLOR_PRIMARIES	UNDEFINED, BT709, UNSPECIFIED, RESERVED, BT470M, BT470BG, SMPTE170M, SMPTE240M, FILM, BT2020, SMPTE428, SMPTE431, SMPTE432, JEDEC_P22	Color space primaries for the compressed output surface which are the maximum red, green, and blue value permitted within the color space. Used (alongside TRANSFER_CHARACTERISTIC and NOMINAL_RANGE parameters) to describe surface in HDR use case. See ColorSpace.h for enumeration.
Encoder SVC parameters	NUM_TEMPORAL_LAYERS	1 … Maximum number of temporal layers supported	Sets the number of temporal layers. SVC with temporal scalability is enabled when the number of layers is greater than 1. The maximum number of temporal layers supported is determined by the corresponding encoder capability. Remarks: • Actual modification of the number of temporal layers will be delayed until the start of the next temporal GOP. Intra-refresh feature is not supported with SVC.
Encoder SVC per-layer parameters	TL<TL_Num>. QL<QL_Num>. <Parameter_name>	Parameter-specific values	Configures rate-control parameter per SVC layer. • TL_Num — temporal layer number • QL_Num — quality layer number • Parameter_name — rate-control parameter name (see below) Rate-control parameters supported • HevcTargetBitrate • HevcPeakBitrate • HevcVBVBufferSize • HevcFrameRate • HevcMinQP_I • HevcMinQP_P • HevcMinQP_I • HevcMinQP_P • HevcQP_I • HevcQP_P • HevcFillerDataEnable • HevcRateControlSkipFrameEnable • HevcEnforceHRD • HevcMaxAUSize (Refer to rate-control parameters section of this table for details) Remarks: Quality layers are not supported. "QL0" must be used for quality layers.

Table A-2. Input frame and encoded data parameters

Category	Name (prefix "AMF_VIDEO_ENCODER_HEVC_")	Values	Description
Frame per-submission parameters	INSERT_HEADER	True/False	Inserts SPS, PPS and VPS
	INSERT_AUD	True/False	Inserts AUD
	FORCE_PICTURE_TYPE	NONE, IDR, I, P	Forces the picture type
	END_OF_SEQUENCE	True/False	End of sequence
	MARK_CURRENT_WITH_LTR_INDEX	-1 … (MaxOfLTRFrames -1)	If != -1, the current picture is coded as a long-term reference with the given index. Remarks: • When the user controls N LTRs (using the corresponding Create parameter), then the LTR Index the user can assign to a reference picture varies from 0 to N-1. By default, the encoder will "use up" available LTR Indices (i.e. assign them to references) even if the user does not request them to be used. • When LTR is used with SVC encoding, only base temporal layer pictures can be coded as LTR. In this case, the request to mark the current picture as LTR would be delayed to the next base temporal layer picture if the current picture is in an enhancement layer. If the user submits multiple requests to mark current as LTR between base temporal layer pictures, then only the last request is applied.
	FORCE_LTR_REFERENCE_BITFIELD	Bitfield (MaxOfLTRFrames (max possible 16 bits))	Force LTR Reference allowed bitfield. If == 0, the current picture should predict from the default reference. If != 0, the current picture should predict from one of the LTRs allowed by the bitfield (bit# = LTR Index#). Remarks: • E.g. if Bit#0 = 1, then the existing LTR with LTR Index = 0 may be used for reference. The bitfield may allow more than one LTR for reference, in which case the encoder is free to choose which one to use. This bitfield also disallows existing LTRs not enabled by it from current/future reference. • E.g. if Bit#1 = 0, and there is an existing reference with LTR Index = 1, then this LTR Index will not be used for reference until it is replaced with a newer reference with the same LTR Index.
	ROI_DATA	Video surface in AMF_SURFACE_GRAY32 format	Important value for each 64x64 block ranges from 0 to 10, stored in 32bit unsigned format.
	STATISTICS_FEEDBACK	True/False (On/Off)	Instruct encoder to collect and feedback statistics.
	PSNR_FEEDBACK	True/False (On/Off)	Signal encoder to calculate PSNR score
	SSIM_FEEDBACK	True/False (On/Off)	Signal encoder to calculate SSIM score
	BLOCK_QP_FEEDBACK	True/False (On/Off)	Instruct encoder to collect and feedback block level QP values.

Category	Name (prefix "AMF_VIDEO_ENCODER_HEVC_")	Values	Description
	REFERENCE_PICTURE	AMFSurface	Injected reference picture. Valid with PICTURE_TRANSFER_MODE turned on
Encoded data parameters	OUTPUT_DATA_TYPE	I, P	Type of encoded data
	OUTPUT_MARKED_LTR_INDEX	-1 … (MaxOfLTRFrames -1)	Marked as LTR Index. If != -1, then this picture was coded as a long-term reference with this LTR Index.
	OUTPUT_REFERENCED_LTR_INDEX_BITFIELD	Bitfield (MaxOfLTRFrames (max possible 16 bits))	Referenced LTR Index bitfield. If != 0, this picture was coded to reference long-term references. The enabled bits identify the LTR Indices of the referenced pictures (e.g. if Bit #0 = 1, then LTR Index 0 was used as a reference when coding this picture).
	OUTPUT_TEMPORAL_LAYER	0 … (Maximum number of temporal layers supported - 1)	Temporal layer of the encoded picture
	RECONSTRUCTED_PICTURE	AMFSurface	Reconstructed picture. Valid with PICTURE_TRANSFER_MODE turned on

Table A-3. Default value of parameters

Type	Name (prefix "AMF_VIDEO_ENCODER_HEVC_")	Transcoding	Ultra low latency	Low latency	Webcam	HQ	HQLL
Static Parameters (Set at creation time)	PROFILE	Main	Main	Main	Main	Main	Main
	PROFILE_LEVEL	6.2	6.2	6.2	6.2	6.2	6.2
	TIER	Main	Main	Main	Main	Main	Main
	MAX_LTR_FRAMES	0	0	0	0	0	0
	MAX_NUM_TEMPORAL_LAYERS	1	1	1	1	1	1
Rate control	RATE_CONTROL_METHOD	PEAK_CONSTRAINED_VBR	LATENCY_CONSTRAINED_VBR	PEAK_CONSTRAINED_VBR	PEAK_CONSTRAINED_VBR	PEAK_CONSTRAINED_VBR	PEAK_CONSTRAINED_VBR
	FRAMERATE	30 fps	30 fps	30 fps	30 fps	30 fps	30 fps
	VBV_BUFFER_SIZE	20 mbits	735 kbits	4 mbits	2 mbits	40 mbits	10 mbits
	INITIAL_VBV_BUFFER_FULLNESS	64	64	64	64	64	64
	PREENCODE_ENABLE	false	false	false	false	false	false
	ENABLE_VBAQ	false	false	false	false	false	false
	TARGET_BITRATE	20 mbps	20 mbps	20 mbps	20 mbps	20 mbps	20 mbps
	PEAK_BITRATE	30 mbps	20 mbps	20 mbps	20 mbps	80 mbps	30 mbps
	MIN_QP_I	0	0	0	0	0	0
	MAX_QP_I	51	51	51	51	51	51
	MIN_QP_P	0	0	0	0	0	0
	MAX_QP_P	51	51	51	51	51	51
	QP_I	26	26	26	26	26	26
	QP_P	26	26	26	26	26	26
	ENFORCE_HRD	false	true	false	false	false	false
	MAX_AU_SIZE	0	0	0	0	0	0
	FILLER_DATA_ENABLE	false	false	false	false	false	false
	RATE_CONTROL_SKIP_FRAME_ENABLE	false	true	true	true	false	false
Picture Control	HEADER_INSERTION_MODE	0	0	0	0	0	0
	GOP_SIZE	30	300	300	30	30	30
	NUM_GOPS_PER_IDR	1	1	1	1	1	1
	DE_BLOCKING_FILTER_DISABLE	false	false	false	false	false	false
	SLICES_PER_FRAME	1	1	1	1	1	1
	QUALITY_PRESET	Balanced	Speed	Speed	Speed	Quality	Quality
	INTRA_REFRESH_NUM_CTBS_PER_SLOT	0	0	0	0	0	0
Motion	MOTION_HALF_PIXEL	1	1	1	1	1	1

Type	Name (prefix "AMF_VIDEO_ENCODER_HEVC_")	Transcoding	Ultra low latency	Low latency	Webcam	HQ	HQLL
estimation	MOTION_QUARTERPIXEL	1	1	1	1	1	1
SVC	NUM_TEMPORAL_LAYERS	1	1	1	1	1	1
New parameters	ENABLE_VBAQ	off	off	off	off	on	on
	LOWLATENCY_MODE	off	on	off	off	off	on
	HIGH_MOTION_QUALITY_BOOST_ENABLE	off	off	off	off	on	on
	PREENCODE_ENABLE	off	off	off	off	on	off
	PRE_ANALYSIS_ENABLE	off	off	off	off	off	off
	MAX_NUM_REFRAMES	1	1	1	1	1	1
	QUERY_TIMEOUT	0	0	0	0	50	50

Table A-4. Encoder statistics feedback

Statistic Name	Description
STATISTIC_FRAME_QP	QP of the first encoded CTB in a picture
STATISTIC_AVERAGE_QP	Average QP of all encoded CTBs in a picture
STATISTIC_MAX_QP	Max QP among all encoded CTBs in a picture
STATISTIC_MIN_QP	Min QP among all encoded CTBs in a picture
STATISTIC_PIX_NUM_INTRA	Number of intra-coded pixels
STATISTIC_PIX_NUM_INTER	Number of inter-coded pixels
STATISTIC_PIX_NUM_SKIP	Number of skip-coded pixels
STATISTIC_BITCOUNT_RESIDUAL	Frame level bit count of residual data
STATISTIC_BITCOUNT_MOTION	Frame level bit count of motion vectors
STATISTIC_BITCOUNT_INTER	Frame level bit count of inter CTBs
STATISTIC_BITCOUNT_INTRA	Frame level bit count of intra CTBs
STATISTIC_BITCOUNT_ALL_MINUS_HEADER	Frame level bit count of the bitstream excluding header
STATISTIC_MV_X	Accumulated absolute values of MVX for full encoding
STATISTIC_MV_Y	Accumulated absolute values of MVY for full encoding
STATISTIC_RD_COST_FINAL	Frame level final RD cost
STATISTIC_RD_COST_INTRA	Frame level RD cost for intra mode
STATISTIC_RD_COST_INTER	Frame level RD cost for inter mode
STATISTIC_SATD_FINAL	Frame level final SATD
STATISTIC_SATD_INTRA	Frame level SATD for intra mode
STATISTIC_SATD_INTER	Frame level SATD for inter mode

Table A-5. Encoder PSNR/SSIM feedback

Statistic Name	Description
STATISTIC_PSNR_Y	PSNR Y
STATISTIC_PSNR_U	PSNY U
STATISTIC_PSNR_V	PSNR V
STATISTIC_PSNR_ALL	PSNR YUV
STATISTIC_SSIM_Y	SSIM Y
STATISTIC_SSIM_U	SSIM U
STATISTIC_SSIM_V	SSIM V
STATISTIC_SSIM_ALL	SSIM YUV