|  |  |
| --- | --- |
| 3GPP TS 26.265 V0.2.2 (2024-05) | |
| Technical Specification | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  Media Delivery: Video Capabilities and Operation Points (Release 19) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  https://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 5

Introduction 6

1 Scope 7

2 References 7

3 Definitions of terms, symbols and abbreviations 7

3.1 Terms 7

3.2 Symbols 8

3.3 Abbreviations 8

4 Context and Definitions 8

4.1 Motivation 9

4.2 Reference architectures and definitions 9

4.3 Specification 10

4.4 Video representation formats 10

4.5 Reference API parameters 10

5.2 Codecs, Profiles and Levels 11

5.2.1 Codec & profile 11

5.2.2 Codec & profile & Levels 11

5.3 Single-Instance Decoding Capabilities 11

5.4 Single-Instance Encoding Capabilities 12

5.5 Multi-Instance Decoding Capabilities 13

5.6 Multi-Instance Encoding Capabilities 14

6 Media Encapsulation and Playback 14

6.1 Introduction 14

7 Video Operation Points 14

7.1 Introduction 14

Annex <A> (normative): Registration Information 14

Annex <B> (informative): Mapping of Reference Architecture to Implementations 15

B.1 Introduction 15

B.2 WebCodecs API 15

Annex <X> (informative): Change history 16

For definitive guidance on drafting 3GPP TSs and TRs, see [3GPP TS 21.801](https://www.3gpp.org/DynaReport/21801.htm).

Ensure all blue guidance text is removed before submitting the TS/TR to the TSG for approval.

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# Introduction

This clause is optional. If it exists, it shall be the second unnumbered clause.

# 1 Scope

This clause shall start on a new page.

The present document …

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[h264] ITU-T Recommendation H.264 (08/2021): "Advanced video coding for generic audiovisual services".

[h265] ITU-T Recommendation H.265 (09/2023): "High efficiency video coding".

[CMAF] ISO/IEC 23000-19: "Information Technology Multimedia Application Format (MPEG-A) – Part 19: Common Media Application Format (CMAF) for segmented media".

[CENC] ISO/IEC 23001-7: "MPEG systems technologies - Part 7: Common encryption in ISO base media file format files".

[DPC] CTA-5003-A & Errata: "Web Application Video Ecosystem (WAVE): Device Playback Capabilities Specification", available at <https://cdn.cta.tech/cta/media/media/resources/standards/pdfs/cta-5003-final.pdf>.

[6381] IETF RFC 6381: The 'Codecs' and 'Profiles' Parameters for "Bucket" Media Types.

[MSE] 3GPP TR 26.857, "5G Media Service Enablers"

# 3 Definitions of terms, symbols and abbreviations

This clause and its three (sub) clauses are mandatory. The contents shall be shown as "void" if the TS/TR does not define any terms, symbols, or abbreviations.

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

**Bitstream:** A bitstream that conforms to a video encoding format and certain Operation Point.

**Operation Point:** A collection of discrete combinations of different content formats including spatial and temporal resolutions, colour mapping, transfer functions, etc. and the encoding format.

**Receiver:** A receiver that can decode and render any bitstream that is conforming to a certain Operation Point.

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

Symbol format (EW)

<symbol> <Explanation>

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

Abbreviation format (EW)

<ABBREVIATION> <Expansion>

# 4 Context and Definitions

Editor’s Note from 619, clause 5.1

The principles of existing video capabilities are built around the following principles:

**Bitstream:** A media bitstream that conforms to a video encoding format and certain Operation Point.

**Operation Point:** A collection of discrete combinations of different content formats including spatial and temporal resolutions, colour mapping, transfer functions, etc. and the encoding format.

**Receiver:** A receiver that can decode and render any bitstream that is conforming to a certain Operation Point.

Decoding capabilities are defined which are a combination of

* The capability to decode a bitstream conforming to a certain profile and level
* The bitstream being restricted in terms of flags and settings

An illustration of an operation points and decoding capabilities is provided below.

A diagram of a diagram with Crust in the background

Description automatically generated

*Receivers* are a combination of decoding capabilities and the ability to rendering the formats included in an operation point.

The timing and the properties of the format may be signaled in the bitstream, or may be signaled by external means, for example on packaging level, i.e. on ISO BMFF or RTP level.

At the receiving end conformance always refers to real-time decoding and rendering.

Bitstreams can either conform to any of the above “circles”

* Codec & Profile
* Level
* Decoding capabilites
* Operation Point

Concurrent decoding capabilities are defined as the ability to decode several bitstreams in parallel.

Encoding capabilities are defined by the ability to encode a *video signal* with certain boundary parameters to a bitstream that is decodable (and possibly can be rendered). Typically, specifications would require real-time encoding.

Decoding capabilities can be shared across many different applications.

Operation Points are more specific towards applications and may not or only partially be defined in a new spec.

## 4.1 Motivation

Video codecs, encoders and decoders are core components of 3GPP services. At the same time, video encoders and decoders residing on 3GPP UEs and defined in 3GPP specifications also provide interoperability points for third-party services. Video capabilities are predominantly independent of the service in use. This specification addresses the definition of video capabilities and operating points such that 3GPP service specifications as well as third-party service providers can refer to the interoperability points defined in this specification.

The present specification makes use some of the concepts recommended in TR 26.857 [2], i.e. the concept of Media Service Enablers.

## 4.2 Reference architectures and definitions

In order to define the normative aspects of this specification, reference architectures are defined. The core architecture is provided in Figure 4.2-1. The workflow addresses the generation of a *video bitstream* from a video signal using a *video encoder* as well as the decoding of a video bitstream by a *video decoder* and providing the resulting decoded video as well as associated metadata to a rendering and display process. The video encoder as well as the video decoder may be configured to certain operations indicated by APIs in Figure 4.2-1. These APIs are not normatively specified but serve as an example reference to configure encoders and decoders as documented in Annex [A].



Figure 4.2-1 Reference architecture for video operating points and capabilities

A more system-centric architecture is provided in Figure 4.2-2. The workflow addresses the generation of a *transport stream* from a video signal using a *video encoder* and a *packager*. The package may include for example timing and metadata information. The de-packaging and decoding of the *transport stream* by a de-packager and a *video decoder*, respectively, allows for providing the resulting video signal as well as associated metadata to a rendering and display process. Again, the packager/encoder as well as the de-packager/decoder may be configured to certain operations indicated by APIs in Figure 4.2-2.



Figure 4.2-2 Reference architecture for system operating points and capabilities

Editor’s Note: A reference architecture for multiple decoders still needs to be defined.

Based on this introduction, the following terms are defined

**Operating Point:** A collection of different possible video formats including spatial and temporal resolutions, colour mapping, transfer functions, etc. and a video encoding format.

**Bitstream**: A compressed media representation presented as a sequence of bits that conforms to a particular video coding specification/format and one or more Operating Points.

**Receiver**: A device that can ingest and decode any bitstream that is conforming to a particular video coding specification and Operating Point, and optionally render it

In addition, on system level the following terms are defined:

**System Operating Point:** A collection of different possible video formats including spatial and temporal resolutions, colour mapping, transfer functions, etc., a video encoding and a packaging format.

**Transport Stream:** A packaged media bitstream that conforms to a particular video coding and packaging specification/format and one or more Operating Points.

**System Receiver:** A receiver that can de-package and decode any system bitstream that is conforming to a particular System Operating Point, and optionally render it.

## 4.3 Specification

This specification defines the following capabilities:

- Video Decoding capability: The capability to decode any video bitstream that conforms to an operating point and provides a conforming output video signal and possibly associated metadata.

- System Receiver capability: The capability to un-package and decode any transport stream that conforms to a system operating point and provides a conforming output video signal and possibly associated metadata

- Video Encoding capability: The capability to encode any video signal included in the operating point to a bitstream that is decodable by decoder that conforms to the same operating point.

- System Transmitter capability: The capability to encode and package any video signal included in the operating point to a system bitstream that can be unpacked and decoded by a system receiver that conforms to the same operating point.

While not explicitly stated in the capabilities, it is a requirement for decoders and receivers to process the data in real-time. For encoder, real-time encoding is typically also a requirement.

## 4.4 Video representation formats

Editor’s Note: Need to define video signal parameters, SDR, HDR, etc.

## 4.5 Reference API parameters

Editor’s Note: The following parameters just summarize parameters available.

Decoding:

* Codec string
* Metadata processing

Encoding:

* Codec string
* Bitrate and bitrate modes
* displayWidth/Height
* framerate
* latency modes
* Codec specific parameters
* Metadata

Packaging

## 5.2 Codecs, Profiles and Levels

Editor’s Note: This is copy and paste from S4-240619, clause 5.2.1 and 5.2.2. More edits are needed.

### 5.2.1 Codec & profile

* AVC/H.264 Progressive High Profile
* HEVC/H.265 Main Profile Main Tier
* HEVC/H.265 Main-10 Profile Main Tier

### 5.2.2 Codec & profile & Levels

* AVC/H.264 Progressive High Profile Level 3.1
* HEVC/H.265 Main Profile Main Tier Level 3.1
* AVC/H.264 Progressive High Profile Level 4.0
* AVC/H.264 Progressive High Profile Level 4.2
* HEVC/H.265 Main-10 Profile Main Tier Level 4.1
* AVC/H.264 Progressive High Profile Level 5.1
* HEVC/H.265 Main-10 Profile Main Tier Level 5.1
* AVC/H.264 Progressive High Profile Level 5.1
* HEVC/H.265 Main-10 Profile Main Tier Level 6.1

## 5.3 Single-Instance Decoding Capabilities

Editor’s Note: This is copy and paste from S4-240619, clause 5.2.3. More edits are needed.

The following decoding capabilities are defined:

**- AVC-FullHD-Dec**: the capability to decode H.264 (AVC) Progressive High Profile Level 4.0 [7] bitstreams.

**- AVC-UHD-Dec:** the capability to decode H.264 (AVC) Progressive High Profile Level 5.1 [7] bitstreams with the following additional requirements:

- the maximum VCL Bit Rate is constrained to be 120 Mbps with cpbBrVclFactor and cpbBrNalFactor being fixed to be 1250 and 1500, respectively; and,

- the bitstream does not contain more than 10 slices per picture.

**- AVC-8K-Dec:** the capability to decode H.264 (AVC) Progressive High Profile Level 6.1 [7] bitstreams with the following requirements:

- the maximum VCL Bit Rate is constrained to be 120 Mbps with cpbBrVclFactor and cpbBrNalFactor being fixed to be 1250 and 1500, respectively; and,

- the bitstream does not contain more than 16 slices per picture.

- the bitstream shall not include horizontal motion vector component values that exceed the range from −2048 to 2047, inclusive, or that have vertical motion vector component values that exceed the range from −512 to 511, inclusive, in units of ¼ luma sample displacement. This constraint should be indicated by using values of log2\_max\_mv\_length\_horizontal less than or equal to 11 and values of log2\_max\_mv\_length\_vertical less than or equal to 9.

- **HEVC-HD-Dec**: the capability to decode H.265 (HEVC) Main Profile, Main Tier, Level 3.1[3] bitstreams that have general\_progressive\_source\_flag equal to 1, general interlaced\_source\_flag equal to 0, general\_non\_packed\_constraint\_flag equal to 1, and general\_frame\_only\_constraint\_flag equal to 1.

- **HEVC-FullHD-Dec**: the capability to decode H.265 (HEVC) Main10 Profile, Main Tier, Level 4.1[3] bitstreams that have general\_progressive\_source\_flag equal to 1, general interlaced\_source\_flag equal to 0, general\_non\_packed\_constraint\_flag equal to 1, and general\_frame\_only\_constraint\_flag equal to 1.

- **HEVC-UHD-Dec**: the capability to decode H.265 (HEVC) Main10 Profile, Main Tier, Level 5.1[3] bitstreams that have general\_progressive\_source\_flag equal to 1, general interlaced\_source\_flag equal to 0, general\_non\_packed\_constraint\_flag equal to 1, and general\_frame\_only\_constraint\_flag equal to 1.

- **HEVC-8K-Dec**: the capability to decode H.265 (HEVC) Main10 Profile, Main Tier, Level 6.1[3] bitstreams that have general\_progressive\_source\_flag equal to 1, general interlaced\_source\_flag equal to 0, general\_non\_packed\_constraint\_flag equal to 1, and general\_frame\_only\_constraint\_flag equal to 1 with the following further limitations:

- the bitstream does not exceed the maximum luma picture size in samples of 33,554,432,

- the maximum VCL Bit Rate is constrained to be 80 Mbps with CpbVclFactor and CpbNalFactor being fixed to be 1000 and 1100, respectively.

## 5.4 Single-Instance Encoding Capabilities

Editor’s Note: This is copy and paste from S4-240619, clause 5.2.4. More edits are needed.

The following encoding capabilities are defined:

**- AVC-FullHD-Enc:** the capability to encode a video signal to a bitstream that is decodable by a decoder that is *AVC-FullHD-Dec* capable as defined in clause 7.1.1.1 with the following additional constraints:

- up to 245,760 macroblocks per second;

- up to a frame size of 8,192 macroblocks;

- up to 240 frames per second;

- the chroma format being 4:2:0; and

- the bit depth being 8 bit;

- **HEVC-HD-Enc**: the capability to encode a video signal with

- up to 33,177,600 luma samples per second;

- up to a luma picture size of 983,040 samples;

- up to 120 frames per second;

- the Chroma format being 4:2:0; and

- the bit depth being 8 bit;

to a bitstream that is decodable by a decoder that is **HEVC-HD-Dec** capable as defined in clause 4.2.2.1.

**- HEVC-FullHD-Enc:** the capability to encode a video signal to a bitstream that is decodable by a decoder that is *HEVC-FullHD-Dec* capable as defined in clause 7.1.1 with the following additional constraints:

- up to 133,693,440 luma samples per second;

- up to a luma picture size of 2,228,224 samples;

- up to 240 frames per second;

- the Chroma format being 4:2:0; and

- the bit depth being either 8 or 10 bit;

**- HEVC-UHD-Enc:** the capability to encode a video signal to a bitstream that is decodable by a decoder that is *HEVC-UHD-Dec* capable as defined in clause 7.1.1 with the following additional constraints:

- up to 534,773,760 luma samples per second;

- up to a luma picture size of 8,912,896 samples;

- up to 480 frames per second;

- the Chroma format being 4:2:0; and

- the bit depth being either 8 or 10 bit;

## 5.5 Multi-Instance Decoding Capabilities

Editor’s Note: This is copy and paste from S4-240619, clause 5.2.5. More edits are needed.

The following multi-instance decoding capabilities are defined:

**- AVC-FullHD-Dec-2**: The capability of supporting up to two (*N*=2) concurrent decoder instances with the aggregate capabilities of *AVC-FullHD-Dec*.

**- AVC-UHD-Dec-4**: The capability of supporting up to four (*N*=4) concurrent decoder instances with the aggregate capabilities of *AVC-UHD-Dec*.

**- HEVC-UHD-Dec-4:** The capability of supporting up to four (*N*=4) concurrent decoder instances with the aggregate capabilities of *HEVC-UHD-Dec*.

**- UHD-Dec-4**: The capability supporting up to four (*N*=4) concurrent decoder instances with either:

- the aggregate capabilities of *AVC-UHD-Dec-4*;

- the aggregate capabilities of *HEVC-UHD-Dec-4*; or,

- the capability of decoding up to 4 bitstreams for which each bitstream does not exceed the capability of being decodable either with *AVC-FullHD-Dec* or *HEVC-FullHD-Dec*.

**- AVC-8K-Dec-8:** The capability of supporting up to eight (*N*=8)concurrent decoder instances with the aggregate capabilities of *AVC-8K-Dec*.

**- HEVC-8K-Dec-8:** The capability of supporting up to eight (*N*=8)concurrent decoder instances with the aggregate capabilities of *HEVC-8K-Dec*.

**- 8K-Dec-8**: The capability supporting up to eight (*N*=8)concurrent decoder instances with either:

- the aggregate capabilities of *AVC-8K-Dec-8*;

- the aggregate capabilities of *HEVC-8K-Dec-8*; or,

- the capability of decoding up to:

- eight bitstreams for which each bitstream does not exceed the capability of being decodable either with *AVC-FullHD-Dec* or *HEVC-FullHD-Dec*; or,

- four bitstreams for which each bitstream does not exceed the capability of being decodable either with *AVC-UHD-Dec* or *HEVC-UHD-Dec*.

## 5.6 Multi-Instance Encoding Capabilities

Editor’s Note: Details not yet collected in S4-240619.

# 6 Media Encapsulation and Playback

## 6.1 Introduction

# 7 Video Operation Points

## 7.1 Introduction

Annex <A> (normative):  
Registration Information

Editor’s Note: Will collect and registration information such as URNs.

Annex <B> (informative):  
Mapping of Reference Architecture to Implementations

# B.1 Introduction

This annex provides some background on how to map the reference architectures defined in clause 4 into concrete implementations. The mapping of the capabilities, the configuration of the encoders and decoders through APIs as well as some workflow aspects are provided.

The Annex is not considered to prescribe any implementation but is expected to support implementors to integrate the capabilities and operating points defined in this specification into their workflows.

The Annex also serves as an analyis on what functionalities are available in existing implementations and where there are potential gaps that may be addressed by the owners of the implementation to fully support all features.

# B.2 WebCodecs API

Editor’s Note: Analyze the configuration information with the APIs defined in WebCodecs. More work on this is needed.

The configuration of the codec is here

dictionary ***VideoDecoderConfig*** {

required [DOMString](https://webidl.spec.whatwg.org/#idl-DOMString) [codec](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-codec);

[AllowSharedBufferSource](https://webidl.spec.whatwg.org/#AllowSharedBufferSource) [description](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-description);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] [unsigned long](https://webidl.spec.whatwg.org/#idl-unsigned-long) [codedWidth](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-codedwidth);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] [unsigned long](https://webidl.spec.whatwg.org/#idl-unsigned-long) [codedHeight](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-codedheight);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] [unsigned long](https://webidl.spec.whatwg.org/#idl-unsigned-long) [displayAspectWidth](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-displayaspectwidth);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] [unsigned long](https://webidl.spec.whatwg.org/#idl-unsigned-long) [displayAspectHeight](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-displayaspectheight);

[VideoColorSpaceInit](https://www.w3.org/TR/webcodecs/#dictdef-videocolorspaceinit) [colorSpace](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-colorspace);

[HardwareAcceleration](https://www.w3.org/TR/webcodecs/#enumdef-hardwareacceleration) [hardwareAcceleration](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-hardwareacceleration) = "no-preference";

[boolean](https://webidl.spec.whatwg.org/#idl-boolean) [optimizeForLatency](https://www.w3.org/TR/webcodecs/#dom-videodecoderconfig-optimizeforlatency);

};

dictionary ***VideoEncoderConfig*** {

required [DOMString](https://webidl.spec.whatwg.org/#idl-DOMString) [codec](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-codec);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] required [unsigned long](https://webidl.spec.whatwg.org/#idl-unsigned-long) [width](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-width);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] required [unsigned long](https://webidl.spec.whatwg.org/#idl-unsigned-long) [height](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-height);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] [unsigned long](https://webidl.spec.whatwg.org/#idl-unsigned-long) [displayWidth](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-displaywidth);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] [unsigned long](https://webidl.spec.whatwg.org/#idl-unsigned-long) [displayHeight](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-displayheight);

[[EnforceRange](https://webidl.spec.whatwg.org/" \l "EnforceRange)] [unsigned long long](https://webidl.spec.whatwg.org/#idl-unsigned-long-long) [bitrate](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-bitrate);

[double](https://webidl.spec.whatwg.org/#idl-double) [framerate](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-framerate);

[HardwareAcceleration](https://www.w3.org/TR/webcodecs/#enumdef-hardwareacceleration) [hardwareAcceleration](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-hardwareacceleration) = "no-preference";

[AlphaOption](https://www.w3.org/TR/webcodecs/#enumdef-alphaoption) [alpha](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-alpha) = "discard";

[DOMString](https://webidl.spec.whatwg.org/#idl-DOMString) [scalabilityMode](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-scalabilitymode);

[VideoEncoderBitrateMode](https://www.w3.org/TR/webcodecs/#enumdef-videoencoderbitratemode) [bitrateMode](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-bitratemode) = "variable";

[LatencyMode](https://www.w3.org/TR/webcodecs/#enumdef-latencymode) [latencyMode](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-latencymode) = "quality";

[DOMString](https://webidl.spec.whatwg.org/#idl-DOMString) [contentHint](https://www.w3.org/TR/webcodecs/#dom-videoencoderconfig-contenthint);

};

For video codec registry, see here: <https://www.w3.org/TR/webcodecs-codec-registry/#video-codec-registry>

For HEVC codec registrations, please go here: <https://www.w3.org/TR/webcodecs-hevc-codec-registration/>

Annex <X> (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Change history | | | | | | | |
| Date | Meeting | TDoc | CR | Rev | Cat | Subject/Comment | New version |
| 2024-04 | SA4#127bis-e | S4-240616 |  |  |  | Initial version | 0.0.0 |
| 2024-04 | SA4#127bis-e | S4-240758 |  |  |  | Version agreed at SA4#127bis-e | 0.1.0 |
| 2024-05 | SA4#128 | S4-241369 |  |  |  | Version agreed at SA4#128 including S4-240911, S4-241296, S4-241298 | 0.2.2 |