Universal Serial Bus Device Class Definition for Video Devices: VP8 Payload

Revision 1.5
August 9, 2012

Contributors

David Roh	Dolby Laboratories Inc.
Choon Chng	Google Inc.
Ville-Mikko Rautio	Google Inc.
Van Duros	Immedia Semiconductor Inc.
Abdul R. Ismail	Intel Corp.
Bradley Saunders	Intel Corporation
Ygal Blum	Jungo
Yoav Nissim	Jungo
Chandrashekhar Rao.	Logitech Inc.
Chris Yokum	MCCI Corporation
Stephen Cooper	Microsoft Corp.
Maribel Figuera	Microsoft Corp.
Richard Webb	Microsoft Corp.
Tim Vlaar	Point Grey Research Inc
Mark Bohm	SMSC
John Sisto	SMSC
Will Harris	Texas Instruments
Grant Ley	Texas Instruments
Paul E. Berg	USB-IF

Please send comments via electronic mail to <video-chair>@usb.org

Revision History

Version	Date	Description
1.5	July 16, 2012	Initial version release as part of UVC 1.5

Table of Contents

1 Introduction ... 1
1.1 Purpose ... 1
1.2 Scope ... 1
1.3 Related Documents ... 1
1.4 Terminology ... 1
1.4.1 Abbreviations ... 1
1.4.2 Definitions ... 1
2 Video Class Specific Information ... 2
2.1 Compression Class ... 2
2.2 Stream Header ... 2
2.3 VP8 Payload Data ... 6
2.4 Temporal Layering with VP8 Encoders (Informative) ... 6
2.5 Error Resiliency Options with VP8 Encoders ... 7
3 Payload-Specific Information ... 8
3.1 Descriptors ... 8
3.1.1 VP8 Video Format Descriptor ... 8
3.1.2 VP8 Video Frame Descriptors ... 10
3.2 Probe and Commit ... 12
3.3 Encoding Unit Controls ... 14
3.3.1 Select Layer ... 15
3.3.2 Slice Mode ... 16
3.3.3 Quantization Parameter Control ... 16
3.3.4 Synchronization and Long Term Reference Frame Control ... 18
3.3.5 Long Term Buffer Size Control ... 20
3.3.6 Long Term Reference Picture Control ... 21
3.3.7 Long Term Reference Validation Control ... 22
3.3.8 Error Resiliency Control ... 22
3.4 Video Samples ... 23

List of Tables

Table 2-1 Payload Header Format for VP8 Streams 2
Table 2-2 Format of the VP8 Payload Header 3
Table 3-1 VP8 Payload Video Format Descriptor 8
Table 3-2 VP8 Payload Video Frame Descriptor 10
Table 3-3 Video Probe and Commit Control Reinterpreted Fields 12
Table 3-4 Simulcast Stream Configuration Subfield in bmLayoutPerStream. 14
Table 3-5 VP8 Specific Use of Encoding Units 14
Table 3-6 Select Layer Control 15
Table 3-7 Bit Layout of wLayerOrViewID for SVC Wildcard Masks 15
Table 3-8 Slice Mode Control 16
Table 3-9 Quantization Parameter Control 16
Table 3-10 Mapping 6-bit Index Value to 7-bit VP8 Quantization Parameter 18
Table 3-11 Mapping 3-bit Delta Index Value to 5-bit VP8 Quantization Parameter Delta 18
Table 3-12 Synchronization and Long Term Reference Frame Control 18
Table 3-13 Long Term Buffer Size Control. 20
Table 3-14 Golden and Alternate Buffer Control with Temporal Enhancement Layers 20
Table 3-15 Long Term Reference Picture Control 21
Table 3-16 Long Term Reference Validation Control 22
Table 3-17 Error Resiliency Control. 23

List of Figures

Figure 2-1 Temporal Layering with VP8. 7
Figure 3-1 VP8 Payload in USB protocol layers. 23

1 Introduction

1.1 Purpose

This document defines the VP8 payload format for devices that are compliant with the *USB Device Class Definition for Video Devices* document.

1.2 Scope

The payload format and associated header information are fully specified in this document. This includes:

- USB Video Class stream header
- Payload-specific header

1.3 Related Documents

USB Specification Revision 3.0, November 12, 2008, www.usb.org
USB Specification Revision 2.0, April 27, 2000, www.usb.org
USB Device Class Definition for Video Devices, www.usb.org
ISO/IEC 10918-1 / ITU-T Recommendation T.81 information technology – Digital compression and coding of continuous-tone still images - Requirements and guide-lines.
RFC 6386: The VP8 Data Format and Decoding Guide (referred to hereafter simply as VP8) is specified in the following link: http://tools.ietf.org/html/rfc6386.

1.4 Terminology

1.4.1 Abbreviations

For the purposes of this specification, the following abbreviations apply:

FID	Frame Identifier
MB	Macro Block
PTS	Presentation Time Stamp
QP	Quantization Parameter
SCR	Source Clock Reference
SLI	Stream Layer ID
SOF	Start of Frame

1.4.2 Definitions

For the purposes of this specification, the following definitions apply:

Bitstream	A sequence of bits that forms a representation of a VP8 Decodable Unit.
DCT Partition	Each VP8 Decodable Unit contains one, two, four or eight DCT Partitions. Each DCT Partition contains actual quantized and logically compressed transform coefficients from one or more macroblock rows in picture.
Frame	For purposes of this specification a frame is an VP8 coded frame.

Intra Coded Inter Frame	A frame in which each macro block is intra coded but is not used to update the golden and alternate reference frames.
Intra Frame	For purposes of this specification an intra frame is defined as a coded frame that does not use any other frames as references for inter-picture prediction.
Reference Frame	A frame that may be used for inter prediction in the decoding process of subsequent frame(s) in decoding order. There are three kinds of Reference Frames in VP8: Previous Reference Frame, Alternate Reference Frame and Golden Frame.

2 Video Class Specific Information

2.1 Compression Class

VP8 is a video coding method developed by the WebM Project. Open source –based WebM Project (www.webmproject.org) maintains the format specifications and reference implementations of the coding method.

The main goals of the VP8 video coding method development effort have been to create high-quality and open video format for the Internet and web. There are no profiles in VP8 and all VP8 decoders should be able to decode all VP8 compliant video bitstreams.

2.2 Stream Header

Every USB Video payload transfer containing VP8 video data must start with a payload header. The format of the payload header is defined as follows.

Table 2-1 Payload Header Format for VP8 Streams

HLE	Header Length							
BFH[0]	SLI	ERR	STI	RES	SCR	PTS	EOF	FID
BFH[1]	RES	RES	RES	RES	RES	GRF	ARF	PRF
BFH[2]	EOH	RES	RES	RES	RES	RES	RES	RES
PTS	PTS[7:0]							
	PTS[15:8]							
	PTS[23:16]							
	PTS[31:24]							
SCR	SCR[7:0]							
	SCR[15:8]							
	SCR[23:16]							
	SCR[31:24]							
	SCR[39:32]							
	SCR[47:40]							
SLI	SLI[7:0]							
	SLI[15:8]							

Table 2-2 Format of the VP8 Payload Header

Offset	Field	Size	Value	Description
0	**bHeaderLength**	1	Number	Header Length field (HLE). Specifies the length of the payload header in bytes, including this field.
1	**bmHeaderInfo**	3	Bitmap	Bit Field Header (BFH) field. Provides information on the sample data following the header, as well as the availability of optional header fields in this header. D0: **Frame ID** (FID). This bit toggles at each VP8 frame start boundary and stays constant for the rest of the frame. D1: **End of Frame** (EOF). This bit indicates the end of a VP8 frame and must be set to 1 only in the last payload transfer belonging to a frame. D2: **Presentation Time Stamp** (PTS). This bit must be set to 1 for each payload header that includes **dwPresentationTime** data. D3: **Source Clock Reference** (SCR). This bit must be set to 1 for each payload transfer that includes **dwSourceClock** data. D4: **Reserved**. Set to 0. D5: **Still Image** (STI). This bit, when set, identifies the payload transfer contains data that belongs to an intra frame. D6: **Error** (ERR). This bit is set if there was an error in the VP8 bitstream or an error in the transmission for this payload. The Stream Error Code control reflects the cause of the error. D7: **Stream Layer ID** (SLI). This bit is set for each payload header that includes **wLayerOrViewID** data.

				D8: **PRF** (PRF). This bit, when set, identifies the payload transfer contains data that belongs to a frame that updates the references of the Previous Reference Frame. D9: **ARF** (ARF). This bit, when set, identifies the payload transfer contains data that belongs to a frame that updates the references of the Alternate Reference Frame. D10: **GRF** (GRF). This bit, when set, identifies the payload transfer contains data that belongs to a frame that updates the references of the Golden Reference Frame. D22-D11: Reserved. Set to 0. D23: **End of header** (EOH). This bit, when set, indicates the end of the BFH fields.
4	**dwPresentationTime**	4	Number	**Presentation Time Stamp** (PTS). The source clock time, in native device clock units, when the raw frame capture begins. This field must be present for every payload transfer. Payload transfers generated from a single capture time must have the same PTS. The PTS is in the same units as specified in the **dwClockFrequency** field of the Video Probe Control response.
8	**scrSourceClock**	6	Number	A two-part **Source Clock Reference** (SCR) value. This field must be present for each payload transfer and must be the same for all payload transfers within the same video frame. The use of SCR is redefined in this specification, putting constraints on SCR that are compatible with the UVC 1.1 specification: • STC must be captured when

				the first video data of a video frame is put on the USB bus. • SCR must remain constant for all payload transfers within a single video frame. D31..D0: Source Time Clock in native device clock units. D42..D32: 1KHz SOF token counter. D47..D43: Reserved. Set to zero. The least-significant 32 bits (D31..D0) contain clock values sampled from the System Time Clock (STC) at the source. The clock resolution shall be specified by the **dwClockFrequency** field of the Probe and Commit response of the device. This value shall comply with the associated stream payload specification. The times at which the STC is sampled must be correlated with the USB Bus Clock. To that end, the next most-significant 11 bits of the SCR (D42..D32) contain a 1-KHz SOF counter, representing the frame number at the time the STC was sampled. The STC is sampled when the first video data of a video frame is put on the USB bus. The SOF counter is the same size and frequency as the frame number associated with USB SOF tokens; it is required to match the current frame number. The most-significant 5 bits (D47..D43) are reserved, and must be set to zero.
14	**wLayerOrViewId**	2	Number	Stream Layer ID (SLI) A combination of temporal_id and stream_id indicating which stream and layer this payload transfer is related to. Bits:

				0-6: Reserved; set to 0. 7-9: temporal_id 10-12: stream_id 13-15: Reserved; set to 0. When present, this field can be used to differentiate between frames belonging to different streams and layers.

As a special case of the ERR bit in the payload header, the VS_STREAM_CODE_CONTROL may indicate the cause of the error as ‘Encoder Buffer Overflow’.

2.3 VP8 Payload Data

VP8 payload data consists of video encoded using the VP8 bitstream format and is byte-oriented. The payload transfer size is variable, and the total payload transfer length (the combined payload header and payload data) for each payload transfer must not exceed the maximum payload transfer size, as specified by the **dwMaxPayloadTransferSize** field in the video Probe and Commit Control.

A single decodable VP8 unit can span multiple payload transfer. If the payload contains the last byte of the decodable VP8 unit, the EOF flag is set in the payload header. No additional bytes may be contained in the payload transfer beyond the decodable VP8 unit containing this particular unit. The next decodable unit must start in different payload transfer. When data from a new capture time begins, the FID is toggled between 0 and 1, and the PTS/SCR must be set in the payload header.

2.4 Temporal Layering with VP8 Encoders (Informative)

Temporal layering in VP8 may be implemented using reference frames as a means to achieve the temporal scalability. In this case, a Temporal Layer in VP8 is not truly a separate layer in the bit stream but a virtual construct through limitation of reference rules between the Temporal Layers. Temporal layering may be configured during the probe and commit negotiation between the host and the device.

This specification allows configuring a maximum of three Temporal Enhancement Layers. Each Temporal Enhancement Layer in each simulcast stream can be configured using the **bmLayoutPerStream** field in Probe and Commit. For example, the Previous Reference Frame acts as the updated reference for the temporal base layer. The Alternate Reference Frame acts as the updated reference for Temporal Enhancement Layer 1 and the Golden Reference Frame acts as the updated reference for Temporal Enhancement Layer 2. Temporal Enhancement Layer 3 does not update any reference frames and depends on other layers for reconstruction. The Figure below illustrates an example of the temporal layering structure for a simulcast stream on which temporal enhancement layers are allowed to use Previous Reference Frames on the same Temporal Layer and reference frames on lower Temporal Layers.

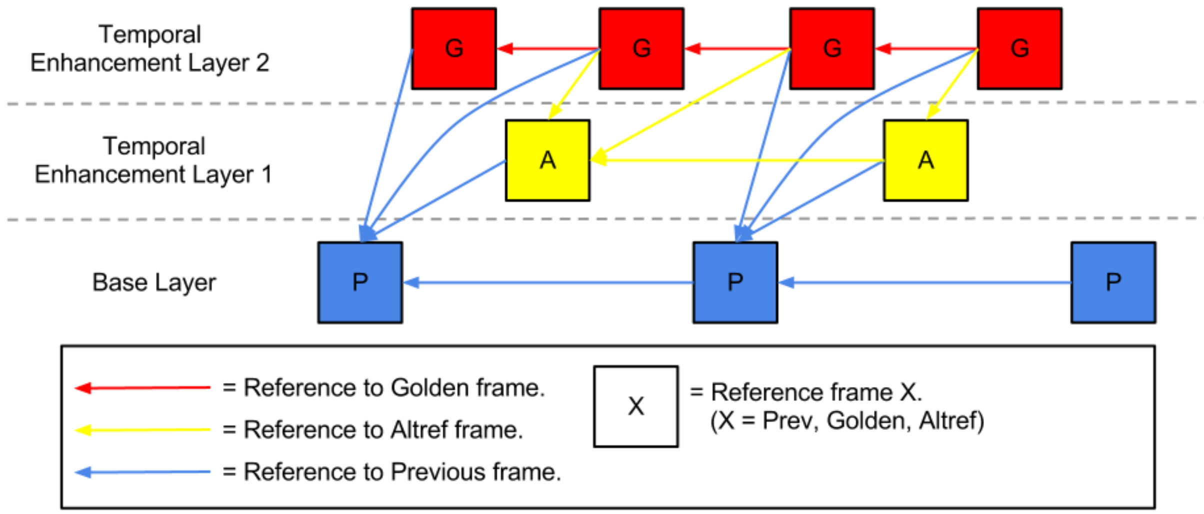

Figure 2-1 Temporal Layering with VP8 (Informative)

When such Temporal Enhancement Layers are used, control over Golden and Alternate Reference Frames may be restricted according to **bmLayoutPerStream** as set in Probe and Commit negotiation and Table 3-14. For example, the configuration presented in Figure 2-1 may be achieved by setting the subfield for the Simulcast Stream in **bmLayoutPerStream** to value 0x0ECD. This signals that:

- Simulcast stream is enabled
- there are two Temporal Enhancement Layers
- the base-layer is allowed to refer only to Previous Reference Frame
- Temporal Enhancement Layer 1 is allowed to refer to Alternate Reference and Previous Reference frames
- Temporal Enhancement Layer 2 is allowed to refer to all reference frames

2.5 Error Resiliency Options with VP8 Encoders

The encoder on the device may be configured to produce an error resilient stream on three levels: no error resiliency, frame-level error resiliency or frame and partition -level error resiliency. Frame-level error resiliency disables cumulative probability updates for entropy coding between frames. This makes entropy decoding inside a frame independent of other frames and decoder can perform correct decoding also in case of lost frames in bitstream. Partition-level error resiliency fixes probability tables also between partitions. In practice this means that even if partitions are lost in middle of frame, the rest of the partitions in the frame can be decoded. Partition level error resiliency has effect only when configuration includes multiple partitions. Trade-off for increasing error resiliency is worse compression ratio compared to encoding bitstream without error resiliency. To select between error resiliency methods, use the Error Resiliency Control available in the Encoding Unit.

3 Payload-Specific Information

3.1 Descriptors

This section provides detailed information about the following Descriptors:

- VP8 Video Format Descriptor
- VP8 Frame Descriptor

3.1.1 VP8 Video Format Descriptor

The VP8 Video Format Descriptor defines the characteristics of a specific video stream. It is used for formats that carry encoded VP8 video stream.

A Terminal corresponding to a USB IN or OUT endpoint, and the interface it belongs to, supports one or more format definitions. To select a particular format, host software sends control requests to the corresponding interface.

The **bFormatIndex** field contains the one-based index of this format Descriptor, and is used by requests from the host to set and get the current video format.

The **bDescriptorSubtype** field uniquely identifies the video data format that should be used when communicating with this interface at the corresponding format index. For a video source function, the host software will deploy the corresponding video format decoder (if necessary) based on the format specified in this field.

The **bMaxCodecConfigDelay** indicates the maximum delay, in number of frames, the device incurs to commit a change to the encoder once the request is received.

The fields **bSupportedPartitionCount**, **bmSupportedSyncFrameTypes**, and **bmSupportedRateControlModes** are used to list possible configuration settings supported by the device. For each of these fields, a specific value must be selected as part of Probe and Commit.

The **wMaxMBperSec** field provides the host with an understanding of the encoder throughput. This field is intended to provide the host with sufficient information to predict a successful multi-stream negotiation.

A VP8 Video Format Descriptor is followed by one or more VP8 Video Frame Descriptor(s); each Video Frame Descriptor conveys information specific to a frame size supported for the format.

A VP8 Video Format Descriptor identifies the following:

Table 3-1 VP8 Payload Video Format Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor in bytes. The value must be 13.
1	**bDescriptorType**	1	Constant	CS_INTERFACE descriptor type.
2	**bDescriptorSubtype**	1	Constant	VS_FORMAT_VP8 or VS_FORMAT_VP8_SIMULCAST descriptor subtype.

3	**bFormatIndex**	1	Number	Index of this format descriptor.
4	**bNumFrameDescriptors**	1	Number	Number of Frame Descriptors following that correspond to this format
5	**bDefaultFrameIndex**	1	Number	Default frame index.
6	**bMaxCodecConfigDelay**	1	Number	Maximum number of frames the encoder takes to respond to a command.
7	**bSupportedPartitionCount**	1	Number	Maximum number of supported partitions per frame. Number must be equal to some power of 2.
8	**bmSupportedSyncFrameTypes**	1	Bitmap	D0: Reset D1: Intra frame. D2: Golden frame. D3: Alternate reference frame. D4: Gradual Decoder Refresh frames. D7-D5: Reserved, set to 0
9	**bResolutionScaling**	1	Number	Specifies the support for resolution downsizing. 0: Not supported. 1: Limited to 1.5 or 2.0 scaling in both directions, while maintaining the aspect ratio. 2: Limited to 1.0, 1.5 or 2.0 scaling in either direction. 3: Limited to resolutions reported by the associated Frame Descriptors 4: Arbitrary scaling. 5 to 255: Reserved Resolution scaling is implemented using the Video Resolution Encoding Unit, and cannot set the resolution above that specified in the currently selected frame descriptor
10	**bmSupportedRateControlModes**	1	Bitmap	Supported rate-control modes. D0: Variable bit rate (VBR) D1: Constant bit rate (CBR)

				D2: Constant QP D3: Global VBR D7-D4: Reserved, set to 0.
11	**wMaxMBperSec**	2	Number	Maximum macroblock processing rate, in units of MB/s, allowed for all VP8 streams by the device.

3.1.2 VP8 Video Frame Descriptors

Each VP8 Frame Descriptor describes a unique video resolution features. Codec usages, capabilities, rate control methods, video frame rates, and so forth are then enumerated for that combination. VP8 Frame Descriptor is also used to determine the range of frame intervals that are supported for the specified frame size.
The VP8 Video Frame Descriptor is used only for video formats for which the VP8 Video Format Descriptor applies (see section 3.1.1, "VP8 Video Format Descriptor").
The **bFrameIndex** field contains the one-based index of this Frame Descriptor, and is used by requests from the host to set and get the current frame index for the format in use. This index is one-based for each corresponding Format Descriptor supported by the device.
The range of frame intervals supported is a discrete set of values where the **dwFrameInterval(x)** fields indicate the range of frame intervals (and therefore frame rates) supported at this frame size. The frame interval is the average display time of a single decoded video frame in 100ns Units.
Each Video Frame Descriptor must support at least one **dwFrameInterval**.

Table 3-2 VP8 Payload Video Frame Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor in bytes. The value must be 31 + (n * 4).
1	**bDescriptorType**	1	Constant	CS_INTERFACE descriptor type.
2	**bDescriptorSubtype**	1	Constant	VS_FRAME_VP8 descriptor subtype
3	**bFrameIndex**	1	Number	Index of this Frame Descriptor
4	**wWidth**	2	Number	The width, in pixels, of valid picture area from the decoding process. Must be a multiple of 2. Regardless of **wWidth** setting coded width is always the next multiple of 16.
6	**wHeight**	2	Number	The height, in pixels, of valid picture area from the decoding process. Must be a multiple of 2. Regardless of **wHeight** setting

				coded height is always the next multiple of 16.
8	**bmSupportedUsages**	4	Bitmap	1: Real-time. 2: Real-time with temporal layering structure as specified in **bmLayoutPerStream** field. D15-D3: Reserved; set to 0. D16: File Storage mode with I and P frames (e.g. IPPP). D17: Reserved; set to 0. D18: File storage all-I-frame mode. D31-D19: Reserved; set to 0.
12	**bmCapabilities**	2	Bitmap	D0: Reserved; set to 0. D1: Reserved; set to 0. D2: Constant frame rate. D3: Separate QP for luma/chroma. D4: Reserved; set to 0. D5: Reserved; set to 0. D6: Golden frame. D7: Alternate reference frame. D15-D8: Reserved; set to 0.
14	**bmScalabilityCapabilities**	4	Bitmap	D2-D0: Maximum number of temporal enhancement layers minus 1. D31-D3: Reserved.
18	**dwMinBitRate**	4	Number	Specifies the minimum bit rate, at maximum compression and longest frame interval, in units of bps, at which the data can be transmitted.
22	**dwMaxBitRate**	4	Number	Specifies the maximum bit rate, at minimum compression and shortest frame interval, in units of bps, at which the data can be transmitted.
26	**dwDefaultFrameInterval**	4	Number	Specifies the frame interval the device indicates for use as a default, in 100-ns units
30	**bNumFrameIntervals**	1	Number	Specifies the number of frame intervals supported (n)
31	**dwFrameInterval(**1**)**	4	Number	Shortest frame interval

				supported (at the highest frame rate), in 100-ns units.
	…			
35+(n*4) –4	**dwFrameInterval (**n**)**	4	Number	Longest frame interval supported (at lowest frame rate), in 100-ns units.

3.2 Probe and Commit

VP8 uses same video Probe and Commit controls (VS_PROBE_CONTROL and VS_COMMIT_CONTROL) as specified in the core USB Video Class specification. Only the following fields have different interpretation within the context of VP8.

Table 3-3 Video Probe and Commit Control Reinterpreted Fields

bUsage	1: Real-time. 2: Real-time with temporal layering structure as specified in **bmLayoutPerStream** field. 3-16: Reserved. Set to 0. 17: File Storage mode with I and P frames (e.g., IPPP) 18: Reserved. Set to 0. 19: File storage with all-I-frame mode. 20-24: File storage modes. 0, 25-31: Reserved. Set to 0. 32-255: Undefined.
bmSettings	D0-D1: Reserved; set to 0. D2: Constant frame rate. D3: Separate QP for luma/chroma. D4-D7: Reserved; set to 0.
bMaxNumberOfRefFramesPlus1	0: Only previous frame. 1: Previous and golden frame. 2: Previous and alternate reference frame. 3: Previous, golden and alternate reference frame.
bmRateControlModes	This field contains 4 subfields, each of which is a 4 bit number. Indicates the rate-control mode for each stream. When **bUsage** is in the range [1, 2], the number of streams is inferred from the **bmLayoutPerStream** field. For all other **bUsage** values, the number of streams is 1. D3-D0: Rate-control mode for the first stream (with stream_id=0.) D7-D4: Rate-control mode for the second stream (with stream_id=1). D11-D8: Rate control mode for the third stream (with stream_id=2). D15-D12: Rate control mode for the fourth stream (with stream_id=3.)

	When **bmRateControlModes** is non-zero, each 4-bit subfield can take one of the following values: **0:** Not applicable, because this stream is non-existent. **1**: VBR . **2**: Reserved. **3**: Constant QP. **4:** Global VBR. **5-15**: Reserved. Set to 0.
bmLayoutPerStream	This field contains 4 subfields, each of which is a 2 byte number. For **bUsage** with value 1, this field indicates the number of simulcast streams using only the "Stream enabled" bit for each stream. [1] For **bUsage** with value 2, this field enables or disables up to four VP8 simulcast streams and may determine each simulcast stream's temporal dependency structure. For other **bUsage** values, this field is ignored. It is recommended to associate streams with lower resolution/lower bit rate with smaller stream_id. D15-D0: Temporal layering structure for stream with stream_id 0. D31-D16: Temporal layering structure for stream with stream_id 1. (set to zero when *not* VS_FORMAT_VP8_SIMULCAST) D47-D32: Temporal layering structure for stream with stream_id 2. (set to zero when *not* VS_FORMAT_VP8_SIMULCAST) D63-D48: Temporal layering structure for stream with stream_id 3. (set to zero when *not* VS_FORMAT_VP8_SIMULCAST) For this specification, each simulcast stream's temporal layering structure bits may be interpreted according to Table 3-4.

[1] E.g. to enable two VP8 simulcast streams when **bUsage**=1 host would set this field to 0x00000101. To enable four simulcast streams this field would be set to 0x01010101.

Table 3-4 Simulcast Stream Configuration Subfield in bmLayoutPerStream. (Informational)

		Temporal Layer Dependency Structure												Simulcast Stream Properties		
		Enhancement layer #3			Enhancement layer #2			Enhancement layer #1			Base Layer					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	Golden allowed	Alt allowed	Prev allowed	Golden allowed	Alt allowed	Prev allowed	Golden allowed	Alt allowed	Prev allowed	Golden allowed	Alt allowed	Prev allowed	Number of temporal enhancement layers		Enabled

3.3 Encoding Unit Controls

Following table summarizes the application of UVC Encoding Unit controls specified in the base UVC Class specification in the context of VP8 Payload. The interpretation of each control with "VP8 specific interpretation" is explained in VP8 context in the following chapters. Controls with "Not applicable" are not supported when streaming data using the VP8 Payload. All unsupported controls must protocol STALL with error code control "Invalid Control" when streaming VP8.

Table 3-5 VP8 Specific Use of Encoding Units

Control	Interpretation
Select Layer	VP8 specific interpretation
Video Resolution	As in UVC Class specification
Profile and Toolset	Not applicable
Minimum Frame Interval	As in UVC Class specification
Slice Mode	VP8 payload specific interpretation
Rate Control Mode	As in UVC Class specification
Average Bitrate Control	As in UVC Class specification
CPB Size Control	As in UVC Class specification
Peak Bit Rate	As in UVC Class specification
Quantization Parameter	VP8 payload specific interpretation
Quantization Parameter Range	As in UVC Class specification
Synchronization and Long Term Reference Frame	VP8 payload specific interpretation
Long Term Reference Buffer Size	VP8 payload specific interpretation
Long Term Picture	VP8 payload specific interpretation
Long Term Reference Validation	VP8 payload specific interpretation
SEI Message Control	Not applicable

Priority ID	Not applicable
Start or Stop Layer	As in UVC Class specification
LevelIDC	Not applicable
Error Resiliency	VP8 payload specific interpretation

Rate control model in VP8 is similar to the model specified in the video class specification with the exception that there is no concept of Hypothetical Reference Decoder (HRD) in VP8 and there is no separate low-delay mode.

3.3.1 Select Layer

VP8 supports only temporal scalability but otherwise the select layer control differs only by the interpretation of bits in **wLayerOrViewID**. The bit specification for VP8 is defined in the following table.

Table 3-6 Select Layer Control

Control Selector		EU_SELECT_LAYER_CONTROL		
Mandatory Requests		SET_CUR, GET_CUR, GET_INFO, GET_LEN		
wLength		2		
Offset	Field	Size	Value	Description
0	**wLayerOrViewID**	2	Number	A combination of temporal_id and stream_id. Bits: 0-6: Reserved; set to 0 7-9: temporal_id 10-12: stream_id 13-15: Reserved; set to 0

3.3.1.1.1 Wildcard Masks

To reduce the number of control calls for encoding units, applications may use wildcard masks. A wildcard mask is a **wLayerOrViewID** where one or more of the subfields have all bits set to 1. Table 3-7 below shows the bit values required to enable a wildcard mask for each of the two subfields in **wLayerOrViewID**.

Table 3-7 Bit Layout of wLayerOrViewID for SVC Wildcard Masks

wLayerOrViewID bits	**reserved**			**stream_id**			**temporal_id**			**reserved**						
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
All Temporal Layers							1	1	1							
All Streams				1	1	1										

Using Wildcard masks, **wLayerOrViewID** can be set to 0x0380, 0x01C0 and 0x1F80 to indicate the scope of configuration applies to all temporal layers, all simulcast streams and all layers across all simulcast streams, respectively.

3.3.2 Slice Mode

In VP8 there is no concept of slice. However, there is concept of DCT Partitions which is method of splitting each picture into smaller pieces of data for transmission and decoding purposes.
For VP8 the Slice Mode control is used to specify the partition mode of the current *stream* that applies to *all layers* in the stream. This control is only supported if the codec supports using more than one DCT Partition per frame.
All GET requests apply to the current partitioning mode.

Table 3-8 Slice Mode Control

Control Selector		EU_SLICE_MODE_CONTROL		
Mandatory Requests		SET_CUR, GET_CUR, GET_DEF, GET_MIN, GET_MAX, GET_INFO, GET_LEN		
wLength		4		
Offset	Field	Size	Value	Description
0	**wSliceMode**	2	Number	0: No partitioning mode. 1: Number of DCT Partitions per frame mode. 2-255: Reserved
2	**wSliceConfig Setting**	2	Number	The meaning of this field depends on wSliceMode mode: Mode 0: No effect. Mode 1: Number of DCT Partitions per frame. Applies to all layers in the stream. Valid values are 1, 2, 4 and 8.

3.3.3 Quantization Parameter Control

For VP8 each frame has four different QP values: 7-bit unsigned absolute value for AC component of the luma transformation result and three signed 5-bit delta values for luma transformation result's DC component, chrome transformation result's AC and DC components. These values can be set in constant QP mode for intra and predicted frames as specified in UVC Class specification. As bi-predictive frames are not known in VP8 context, the quantization parameter values given in **wQpPrime_B** are applied for golden frames instead.

Table 3-9 Quantization Parameter Control

Control Selector		EU_QUANTIZATION_PARAMS_CONTROL		
Mandatory Requests		SET_CUR, GET_CUR, GET_MAX, GET_MIN, GET_DEF, GET_INFO, GET_LEN		
wLength		6		
Offset	Field	Size	Value	Description
0	**wQpPrime_I**	2	Number	Only applicable in constant QP rate-control mode. Use this parameter to set/get QP for I frames.

				D6-D0: $QP'_{Y,AC}$; These bits signal the magnitude of the quantization parameter for luma AC component. 6-bits are interpreted as unsigned index value and mapped to the 7-bit quantization value according to Table 3-10. D9-D7: $\Delta QP'_{Y,DC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11. D12-D10: $\Delta QP'_{Chroma,AC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11. D15-D13: $\Delta QP'_{Chroma,DC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11.
2	**wQpPrime_P**	2	Number	Only applicable in constant QP rate-control mode. Use this parameter to set/get QP for P frames. D6-D0: $QP'_{Y,AC}$; Bits signal the magnitude of the quantization parameter for luma AC component. 6-bits are interpreted as unsigned index value and mapped to the 7-bit quantization value according to Table 3-10. D9-D7: $\Delta QP'_{Y,DC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11. D12-D10: $\Delta QP'_{Chroma,AC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11. D15-D13: $\Delta QP'_{Chroma,DC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11.
4	**wQpPrime_B**	2	Number	Only applicable in constant QP rate control mode. Use this parameter to set/get QP for golden frames. D6-D0: $QP'_{Y,AC}$; Bits signal the magnitude of the quantization parameter for luma AC component. 6-bits are interpreted as unsigned index value and mapped to the 7-bit quantization value according to Table 3-10. D9-D7: $\Delta QP'_{Y,DC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11. D12-D10: $\Delta QP'_{Chroma,AC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11. D15-D13: $\Delta QP'_{Chroma,DC}$; Bits signal the magnitude of delta to $QP'_{Y,AC}$ for this frame type. Bits are interpreted according to Table 3-11.

Table 3-10 Mapping 6-bit Index Value to 7-bit VP8 Quantization Parameter

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
QP	0	1	2	3	4	5	7	8	9	10	12	13	15	17	18	19
Index	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
QP	20	21	23	24	25	26	27	28	29	30	31	33	35	37	39	41
Index	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
QP	43	45	47	49	51	53	55	57	59	61	64	67	70	73	76	79
Index	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**
QP	82	85	88	91	94	97	100	103	106	109	112	115	118	121	124	127

Table 3-11 Mapping 3-bit Delta Index Value to 5-bit VP8 Quantization Parameter Delta

Bits	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
QP	0	-15	-8	-4	-2	2	4	8

3.3.4 Synchronization and Long Term Reference Frame Control

This control is used to manage insertion of intra frames as well as golden and alternate reference frames into the current *stream*. When the host requests the generation of a sync frame, the encoder shall insert the specified **bSyncFrameType** into all the dependency representations associated with the current stream as identified by the current **wLayerOrViewID**.

Table 3-12 Synchronization and Long Term Reference Frame Control

Control Selector		EU_SYNC_REF_FRAME_CONTROL		
Mandatory Requests		SET_CUR, GET_CUR, GET_MIN, GET_MAX, GET_INFO, GET_LEN		
wLength		4		
Offset	Field	Size	Value	Description
0	**bSyncFrameType**	1	Number	0: Reset. Allow the encoder to determine the timing of synchronization frames. 1: Generate an intra frame for the selected layers. Also golden and alternate reference frames will be updated to point to the intra frame. 2: Gradual Decoder Refresh (GDR). Use this synchronization frame type with **bGradualDecoderRefresh** set to 0 to force Intra-Coded Interframe.

				3: GDR with update of the golden frame at the frame where GDR finishes. 4: GDR with update of the alternate reference frame at the frame where GDR finishes. 5: GDR with update of the golden and alternate reference frames at the frame where GDR finishes. 6: Update golden frame. 7: Update alternate reference frame. 8: Update golden and alternate reference frames. 9-255: Reserved.
1	**wSyncFrameInterval**	2	Number	In milliseconds. This field indicates the periodic recurrences of the selected bSyncFrameType. A value of **wSyncFrameInterval**=0 indicates a single **bSyncFrameType** with no requirement for periodic recurrence.
3	**bGradualDecoderRefresh**	1	Number	Indicates a count of frames over which the gradual decoder refresh occurs. Only valid when 2 ≤ **bSyncFrameType ≤** 5 (GDR). When **bSyncFrameType** is not within this range, this field must be 0. From a recovery point of view, (bGradualDecoderRefresh+1) represents the number of frames in decoding order required to completely refresh the picture. Bits: 0-6: Gradual Refresh Period 7: Reserved Use **wSyncFrameInterval** to establish the interval between Gradual Decoder Refresh (GDR) periods.

Note that there may be only one Gradual Decoder Refresh cycle active at any given time. When Gradual Decoder Refresh feature is reconfigured the new setting will override the old one.

GET_MIN and GET_MAX can be used to determine the minimum and maximum Gradual Refresh Period over which the encoder can implement GDR, the minimum and maximum recurrence of all **bSyncFrameType** and whether the device supports changes to **wSyncFrameInterval**. If GET_MIN and GET_MAX return the same **wSyncFrameInterval** value as GET_CUR, then the device does not support changes to this value.

GET_CUR can be used to check whether the last SET_CUR request issued by the host was set correctly on the device. Prior to a successful SET_CUR, GET_CUR is undefined.

3.3.5 Long Term Buffer Size Control

The EU_LTR_BUFFER_SIZE_CONTROL allows for discovery and control of golden and alternate frames on the device. The control provides for a subset of the reference frames to be allocated for host control. If the device does not have enough memory to enable both golden and alternate reference frames at the current resolution or device does not allow the host to manage any of the reference frames, then the GET_MAX shall return **bNumHostControlLTRBuffers** equal to 0. Once the host controllable reference frames are known, the host may then gain the control on them by issuing SET_CUR request with **bNumHostControlLTRBuffers** value indicating the reference frames that are to be controlled by host.

Table 3-13 Long Term Buffer Size Control.

Control Selector		EU_LTR_BUFFER_SIZE_CONTROL		
Mandatory Requests		SET_CUR, GET_CUR, GET_DEF, GET_INFO, GET_LEN, GET_MAX, GET_MIN		
wLength		2		
Offset	**Field**	**Size**	**Value**	**Description**
0	**bNumHostControlLTRBuffers**	1	Number	Reference frames host can control: 0 – none 1 – golden frame 2 – alternate reference frame 3 – golden and alternate reference frames. 4 and above – Reserved.
1	**bTrustMode**	1	Number	Trust mode for the LTR feature. 0 – For each inserted LTR, device sets associated bit in **bmValidLTRs** to 0 (Don't Trust Until) 1 – For each inserted LTR, device sets associated bit in **bmValidLTRs** to 1 (Trust Until)

For this specification, host controllable VP8 reference frame buffers will be limited according to following table.

Table 3-14 Golden and Alternate Buffer Control with Temporal Enhancement Layers

Number of temporal enhancement layers	Golden frame	Alternate reference frame

0	Host or device controllable	Host or device controllable
1	Host or device controllable	Only device controllable
2 or more	Only device controllable	Only device controllable

3.3.6 Long Term Reference Picture Control

The EU_LTR_PICTURE_CONTROL tells the encoder to generate a golden or alternate reference frame. Golden or alternate reference frame can be generated by request only if host has the control over them as specified by EU_LTR_BUFFER_SIZE_CONTROL.

Table 3-15 Long Term Reference Picture Control

Control Selector		EU_LTR_PICTURE_CONTROL		
Mandatory Requests		SET_CUR, GET_CUR, GET_DEF, GET_INFO, GET_LEN, GET_MAX, GET_MIN		
wLength		2		
Offset	**Field**	**Size**	**Value**	**Description**
0	**bPutAtPositionInLTRBuffer**	1	Number	Next frame should be update the certain reference frames. This frame must not be one of the scheduled sync frames schedule by the EU_SYNC_REF_FRAME_CONTROL control. 0 – Encoder is free to choose where to save the frame inside the device controlled references. It cannot be saved in any of the host controlled reference frames. 1 – Update host controlled golden frame. 2 – Update host controlled alternate reference frame. 3 – Update both host controlled golden and alternate reference frames. 4 and above – Reserved.
1	**bLTRMode**	1	Number	Determines which frames can be used as references when creating the new golden or alternate reference frame. 0 – Encoder is restricted to valid host managed reference frames.

				1 – Encoder may use any valid host or device controlled reference frames. 2 – Encoder may use any valid frame. 3 and above: Reserved.

When **bLTRMode** is 0, the new frame generated by this control must only reference host controlled golden or alternate reference frames that have been validated. When **bLTRMode** is 1, the new frame generated by this control must only reference valid host or encoder controlled golden or alternate reference frames. When **bLTRMode** is 2, the encoder is free to use any golden, alternate reference or previous frame references it wishes when creating the new frame. When a request for a new golden or alternate reference frame is still pending, the device shall protocol stall any new requests to this control for new golden or alternate reference frames.

3.3.7 Long Term Reference Validation Control

EU_LTR_VALIDATION_CONTROL instructs the device to update the list of valid reference frames which may be used to encode subsequent frames. A value of 1 means the reference frame is valid while a value of 0 means the frame is invalid. There are two different lists of reference frames, those managed by the host and those managed by the device. This control validates or invalidates both. The **bmValidLTRs** bitmask in this control can be used to indicate which host controlled reference frames can be used by EU_LTR_PICTURE_CONTROL. The **bmValidLTRs** bitmask may also be used to limit which reference frames should be used as references.

Table 3-16 Long Term Reference Validation Control

Control Selector		EU_LTR_VALIDATION_CONTROL		
Mandatory Requests		SET_CUR, GET_CUR, GET_DEF, GET_INFO, GET_LEN		
wLength		2		
Offset	**Field**	**Size**	**Value**	**Description**
0	**bmValidLTRs**	2	Bitmap	A list of reference frames that may be used to generate subsequent frames. D0 – Previous reference frame. D1 – Golden frame. D2 – Alternate reference frame. D3–D15 – Reserved.

3.3.8 Error Resiliency Control

The Error Resiliency control is used to set error resiliency features on the video encoder.

Table 3-17 Error Resiliency Control.

Control Selector		EU_ERROR_RESILIENCY_CONTROL		
Mandatory Requests		SET_CUR, GET_CUR, GET_DEF, GET_RES		
wLength		2		
Offset	Field	Size	Value	Description
0	**bmErrorResiliencyFeatures**	2	Bitmap	Bitmap of error resiliency features supported. D0 – Random Macroblock Intra Refresh. D1 – Frame-level Error Resiliency. D2 – Partition-level Error Resiliency. D15-D3 – Reserved. Set to 0.

In response to a GET_RES response, device shall set the bits for supported error resiliency features to 1. All other bits should be set to 0. In response to a GET_DEF request device shall set the bits to 1 for the tools that are enabled in the device default configuration. All other bits shall be set to 0.

3.4 Video Samples

Each VP8 frame, as defined above, is considered a single video sample. A video sample is made up of one or more *payload transfers* (as defined in the USB Device Class Specification for Video Devices).

Following figure illustrates the layering and its relationship to layering of USB protocol.

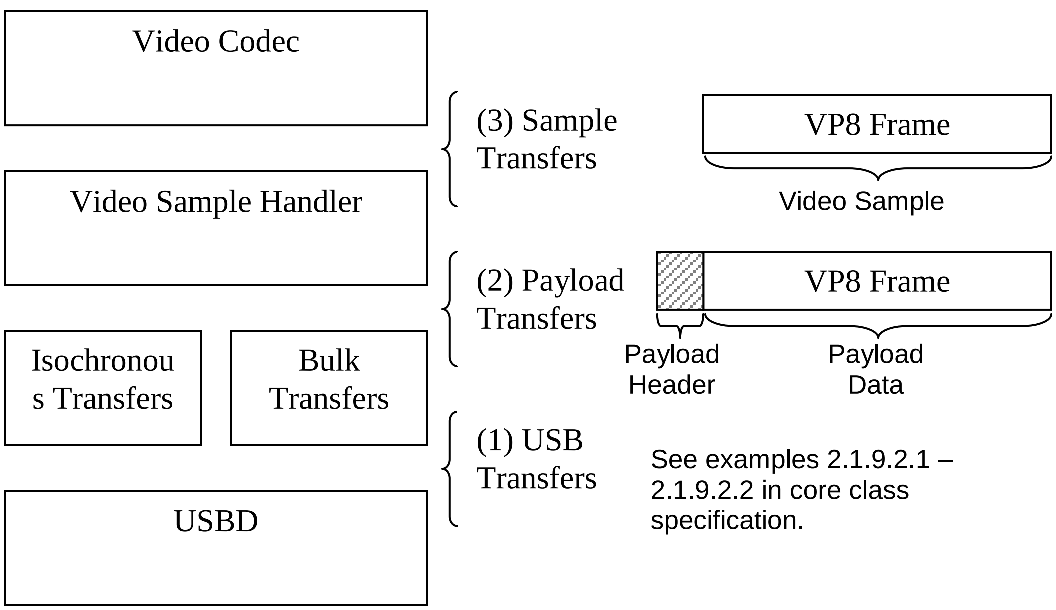

Figure 3-1 VP8 Payload in USB protocol layers.