RP 2025-2007

SMPTE RECOMMENDED PRACTICE

VC-1 Bitstream Storage in the ISO Base Media File Format

Table of Contents

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in Part XIII of its Administrative Practices.

This SMPTE Recommended Practice was prepared by Technology Committee C24.

Introduction

This section is entirely informative and does not form an integral part of this Recommended Practice.

This document was initiated by C24 in response to the need to store VC-1 coded bitstreams in file formats derived from the ISO Base Media File Format such as the MP4 file format and the 3GPP file format.

Approved April 4, 2007

1 Scope

The purpose of this specification is to define the necessary structures for the integration of VC-1 coded bitstreams in a file format that is compliant with the ISO Base Media File Format. Examples of file formats that are derived from the ISO Base Media File Format include the MP4 file format and the 3GPP file format.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this recommended practice. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this recommended practice are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE 421M-2006, Television — VC-1 Compressed Video Bitstream Format and Decoding Process

ISO/IEC 14496-12:2005, Information Technology — Coding of Audio-Visual Objects — Part 12: ISO Base Media File Format

4 VC-1 Track definition

In the terminology of the ISO Base Media File Format specification (ISO/IEC 14496-12), a VC-1 track is a video track. It therefore follows that these rules apply to the media box in the VC-1 track:

- In the Handler Reference box, the handler_type field shall be set to ‘vide’.
- The Media Information Header box shall contain a Video Media Header box.
- The Sample Description Box shall contain a box derived from VisualSampleEntry. (For VC-1 tracks, this box is called **VC1SampleEntry** and is defined in section 6.)

As is the case with all video tracks in ISO Base Media File Format, references to VC-1 samples in the Sample Table Box shall be arranged in Decoding Time order. Decoding Time order is equivalent to the concept of “coded order” in SMPTE 421M.

5 VC-1 Sample definition

A VC-1 sample is defined as follows:

- For Simple and Main profiles: Exactly one VC-1 frame.
- For Advanced profile: One or more Encapsulated Bit-stream Data Units (EBDUs), as defined in Annex E of SMPTE 421M. The VC-1 sample shall contain data belonging to exactly one VC-1 frame.

In the case of interlaced video, a VC-1 frame consists of two fields that may be coded as separate pictures. The two pictures still belong to the same VC-1 frame and hence also to the same VC-1 sample.

It shall be noted that when Advanced profile is used, a VC-1 sample may contain EBDUs that are not part of a VC-1 frame, such as a Sequence Header EBDU. The following rules apply to such EBDUs:

- If a Sequence Header is present in a VC-1 sample, it shall be the first EBDU in the sample.
- If an Entry-point Header is present in a VC-1 sample, it shall be the first EBDU in the sample, unless a Sequence Header is also present, in which case the Entry-point Header shall follow directly after the end of the Sequence Header and any Sequence-level User Data.

The EBDU Start Code field is used to identify the type of the EBDU. The list of values for the EBDU Start Code field is given in SMPTE 421M, Annex E.5.

VC-1 samples shall be byte-aligned. If necessary, up to 7 zero-valued padding bits shall be added to the end of a VC-1 sample to achieve byte-alignment. The Padding Bits Box (defined in section 8.23 of ISO/IEC 14496-12) need not be used to record padding bits that are added to a sample to align its size to the nearest byte boundary.

5.1 Random access points

For Simple and Main profiles, each VC-1 sample that contains an I-picture shall be a random access point. For Advanced profile, a VC-1 sample shall be a random access point if any of the following conditions are met:

- The sample contains both a Sequence Header and an Entry-point Header.
- The sample contains an Entry-point Header and the **no_multiple_seq** bit in **VC1AdvDecSpecStruc** is 1.

The definition of **VC1AdvDecSpecStruc** is given in section 8.4.

The Sync Sample Box shall be used to mark all random access points in the track. The Sync Sample Box is defined in section 8.20 of ISO/IEC 14496-12.

6 VC1SampleEntry definition

The box type of the VC1SampleEntry Box shall be 'vc-1'.

The VC1SampleEntry Box is defined as follows:

VC1SampleEntry ::= **BoxHeader**
Reserved_6
Data-reference-index
Reserved_16
Width
Height
Reserved_4
Reserved_4
Reserved_4
Reserved_2
Reserved_32
Reserved_2
Reserved_2
VC1SpecificBox

Table 1 – VC1SampleEntry fields

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'vc-1'
Reserved_6	Unsigned int(8) [6]		0
Data-reference-index	Unsigned int(16)	Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference boxes.	
Reserved_16	Const unsigned int(32) [4]		0
Width	Unsigned int(16)	Maximum visual width of the stream, in pixels	
Height	Unsigned int(16)	Maximum visual height of the stream, in pixels	
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0
Reserved_2	Const unsigned int(16)		1
Reserved_32	Const unsigned int(8) [32]		0
Reserved_2	Const unsigned int(16)		24
Reserved_2	Const int(16)		-1
VC1SpecificBox		Information specific to the VC-1 decoder.	

The layout of the **VC1SampleEntry** box is identical to that of VisualSampleEntry defined in ISO/IEC 14496-12 (including the reserved fields and their values), except that **VC1SampleEntry** ends with a box containing VC-1 decoder initialization parameters called **VC1SpecificBox**. The **VC1SpecificBox** field structure for VC-1 is defined in section 7.

Width and Height: specify the maximum horizontal and vertical sizes, respectively, of the displayed frames.

7 VC1SpecificBox definition

VC1SpecificBox shall always be included in the **VC1SampleEntry** box.

VC1SpecificBox is composed of the following fields:

VC1SpecificBox ::= **BoxHeader**
DecSpecificInfo

Table 2 – The VC1SpecificBox fields

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'dvc1'
DecSpecificInfo	VC1DecSpecStruc	Structure which holds the VC-1 decoder initialization parameters	

BoxHeader Size and Type: specify the size and type of the VC1SpecificBox. The type must be 'dvc1'.

DecSpecificInfo: A structure of the VC1DecSpecStruc type which is used to store the VC-1 decoder initialization parameters. VC1DecSpecStruc is defined in section 8.

8 VC1DecSpecStruc definition

8.1 Overall definition

VC1DecSpecStruc is defined as follows:

```
struct VC1DecSpecStruc {
                    unsigned int (4)        profile;
                    unsigned int (3)        level;
                    const bit (1)           reserved = 0;
                    if (profile == 0 || profile == 4)
                    {
                            struct VC1SequenceHeader_C      struct_C;
                            struct VC1SequenceHeader_B      struct_B;
                    }
                    else if (profile == 12)
                    {
                            struct VC1AdvDecSpecStruc       struct_AP;
                    }
}
```

The definitions of **VC1DecSpecStruc** members are as follows:

profile: specifies the VC-1 encoding profile used to produce the coded bitstream. It shall be 0 for VC-1 Simple profile, 4 for VC-1 Main profile, and 12 for VC-1 Advanced profile. All other values are SMPTE reserved (as defined in SMPTE 421M).

level: the highest encoding level used in the VC-1 bit stream.
For VC-1 Simple profile, **level** shall be 0 to indicate Low level and 2 to indicate Medium level.
For VC-1 Main profile, **level** shall be 0 to indicate Low level, 2 to indicate Medium level and 4 to indicate High level.
For VC-1 Advanced profile, **level** shall take a value from 0 through 4, corresponding to VC-1 Advanced profile levels L0 through L4, respectively.
All other values are SMPTE reserved (as defined in SMPTE 421M).

struct_C: A structure of the **VC1SequenceHeader_C** type. This structure is identical to STRUCT_SEQUENCE_HEADER_C as defined in Annex J of SMPTE 421M.

struct_B: A structure of the **VC1SequenceHeader_B** type. This structure is identical to STRUCT_SEQUENCE_HEADER_B as defined in Annex J of SMPTE 421M.

struct_AP: A structure of the **VC1AdvDecSpecStruc** type. It is defined in section 8.4.

8.2 VC1SequenceHeader_B definition

VC1SequenceHeader_B is defined as follows:

struct **VC1SequenceHeader_B** {

unsigned int (3)	**level**;
unsigned bit (1)	**cbr**;
const bit (4)	**res1 = 0**;
unsigned int (24)	**hrd_buffer**;
unsigned int (32)	**hrd_rate**;
unsigned int (32)	**framerate**;

}

The definitions of **VC1SequenceHeader_B** members are as follows:

level: shall be set to the same value as is used for the level field in VC1DecSpecStruc (see section 8.1).

cbr: shall be 1 if the content was generated using a constant bit rate model. If it was generated using any other bit rate model, **cbr** shall be 0.

res1: is reserved and shall be set to zero.

hrd_buffer: specifies the buffer size *B* of the hypothetical reference decoder (HRD) in millisecond units. (The HRD is defined in Annex C of SMPTE 421M.)

hrd_rate: specifies the peak transmission rate R of the hypothetical referenced decoder (HRD) in bits per second.

framerate: shall be set to the rounded frame rate (fps) of the track. The **framerate** field shall be set to 0xffffffff if the frame rate is not known, unspecified, or non-constant.

8.3 VC1SequenceHeader_C definition

VC1SequenceHeader_C is defined as follows:

struct **VC1SequenceHeader_C** {

unsigned int (4)	**profile**;
unsigned int (3)	**frmrtq_postproc**;
unsigned int (5)	**bitrtq_postproc**;
unsigned bit (1)	**loopfilter**;
const bit (1)	**reserved1 = 0**;
unsigned bit (1)	**multires**;
const bit (1)	**reserved2 = 1**;
unsigned bit (1)	**fastuvmc**;
unsigned bit (1)	**extended_mv**;
unsigned int (2)	**dquant**;
unsigned bit (1)	**vstransform**;
const bit (1)	**reserved3 = 0**;

```
                    unsigned bit (1)        overlap;
                    unsigned bit (1)        syncmarker;
                    unsigned bit (1)        rangered;
                    unsigned int (3)        maxbframes;
                    unsigned int (2)        quantizer;
                    unsigned bit (1)        finterpflag;
                    const bit (1)           reserved4 = 1;
}
```

The definitions of **VC1SequenceHeader_C** members are as follows:

profile: shall be set to the same value as is used for the profile field in VC1DecSpecStruc (see section 8.1).

frmrtq_postproc: signals quantized frame rate information as defined in Annex J.1.7 of SMPTE 421M.

bitrtq_postproc: signals quantized frame rate information as defined in Annex J.1.8 of SMPTE 421M.

loopfilter: shall be set to 1 if loop filtering is enabled and Main profile is used. If loop filtering is not enabled, the value shall be 0. The value shall be 0 when Simple profile is used.

reserved1: shall be set to 0 and other values shall be forbidden.

multires: shall be set to 1 if the coded frames may be coded at a lower resolution than specified by the **width** and **height** fields in **VC1SampleEntry**, otherwise it shall be set to 0. See Annex J.1.10 of SMPTE 421M for additional constraints on the use of this field.

reserved2: shall be set to 1 and other values shall be forbidden.

fastuvmc: controls the subpixel interpolation and rounding of color-difference motion vectors as defined in Annex J.1.11 of SMPTE 421M. The **fastuvmc** field shall be 1 when Simple profile is used.

extended_mv: shall be set to 1 if extended motion vectors may be present in P- and B-pictures. The **extended_mv** field shall be 0 when Simple profile is used.

dquant: indicates if the quantization step size may vary within a frame, as defined in Annex J.1.13 of SMPTE 421M.

vstransform: shall be set to 1 if variable-sized transform coding is enabled. It shall be set to 0 if variable-sized transform coding is not enabled.

reserved3: shall be set to 0 and other values shall be forbidden.

overlap: shall be set to 1 if overlapped transforms may be used. It shall be set to 0 if overlapped transforms are not used.

syncmarker: shall be set to 1 if synchronization markers may be present in the bit stream and Main profile is used. The value shall be 0 if synchronization markers are not present in the bit stream. The **syncmarker** field shall be 0 when Simple profile is used.

rangered: indicates if range reduction is used in each frame, as defined in Annex J.1.17 of SMPTE 421M. The **rangered** field shall be 0 when Simple profile is used.

maxbframes: indicates the maximum number of B-frames between I- and P-frames, as defined in Annex J.1.18 of SMPTE 421M. If **maxbframes** is 0, B-frames shall not be present in the track. When Simple profile is used, **maxbframes** shall always be 0.

quantizer: specifies the quantizer used in the track as defined in Annex J.1.19 of SMPTE 421M.

finterpflag: shall be set to 1 if the syntax element INTERPFRM is present in the VC-1 Picture header. The **finterpflag** field shall be set to 0 if the syntax element INTERPFRM is not present in the VC-1 Picture header.

reserved4: shall be set to 1 and other values shall be forbidden.

8.4 VC1AdvDecSpecStruc definition

VC1AdvDecSpecStruc is defined as follows:

```
struct VC1AdvDecSpecStruc {
                        unsigned int (3)        level;
                        unsigned bit (1)        cbr;
                        const bit (6)           reserved1 = 0;
                        unsigned bit (1)        no_interlace;
                        unsigned bit (1)        no_multiple_seq;
                        unsigned bit (1)        no_multiple_entry;
                        unsigned bit (1)        no_slice_code;
                        unsigned bit (1)        no_bframe;
                        const bit (1)           reserved2 = 0;
                        unsigned int (32)       framerate;
                        unsigned int (8)        seqhdr_ephdr[];
}
```

The definitions of **VC1AdvDecSpecStruc** members are as follows:

level: shall be set to the same value as is used for the level field in VC1DecSpecStruc (see section 8.1).

cbr: shall be 1 if the content was generated using a constant bit rate model. If it was generated using any other bit rate model, **cbr** shall be 0.

reserved1: is reserved and shall be set to zero.

no_interlace: shall be set to 1 if interlaced coding of frames is not used. The **no_interlace** field shall be 0 if frames may use interlaced coding.

no_multiple_seq: shall be set to 1 if the track contains no Sequence headers or if all Sequence headers in the track are identical to the Sequence header that is specified in the **seqhdr_ephdr** field. The **no_multiple_seq** field shall be 0 if the track may contain Sequence headers that are different from the Sequence header specified in the **seqhdr_ephdr** field.

no_multiple_entry: shall be set to 1 if all Entry-point headers in the track are identical to the Entry-point header that is specified in the **seqhdr_ephdr** field. The **no_multiple_entry** field shall be 0 if the track may contain Entry-point headers that are different from the Entry-point header specified in the **seqhdr_ephdr** field.

no_slice_code: shall be set to 1 if frames are not coded as multiple slices; i.e., the VC-1 slice layer is not used. The **no_slice_code** field shall be 0 if frames may be coded as multiple slices.

no_bframe: shall be set to 1 if neither B-frames nor BI-frames are present in the track. The **no_bframe** field shall be 0 if B-frames or BI-frames may be present in the track.

reserved2: is reserved and shall be set to zero.

framerate: shall be set to the rounded frame rate (fps) of the track. The **framerate** field should be set to 0xffffffff if the frame rate is not known, unspecified, or non-constant.

seqhdr_ephdr: shall contain a Sequence header EBDU followed by an Entry-point header EBDU. One or more Sequence-level User Data EBDUs may be present between the Sequence header EBDU and the Entry-point EBDU. One or more Entry-point level User Data EBDUs may be present directly following the end of the Entry-point header EBDU. If the size of the last EBDU in the **seqhdr_ephdr** field is not a multiple of 8 bits, up to 7 zero-valued padding bits shall be added to the end to achieve byte-alignment.