International Standard

ISO/IEC 23090-25

First edition
2025-01

Information technology — Coded representation of immersive media —

Part 25: Conformance and reference software for carriage of visual volumetric video-based coding data

Technologies de l'information — Représentation codée de média immersifs —

Partie 25: Conformité et logiciel de référence pour le transport de données de codage basé sur la vidéo volumétrique

Reference number
ISO/IEC 23090-25:2025(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23090 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

The conformance and reference software in this document serves two main purposes:

— Validation of the written specification of ISO/IEC 23090-10;

— Conformance testing for checking interoperability for the various applications against the reference software which aims to be compliant with ISO/IEC 23090-10.

The reference software is structured as an extension of the ISOBMFF reference software library and provides additional functions required for ISO/IEC 23090-10. In addition, it includes a small command line application that uses the library to perform some basic file operations such as multiplexing and demultiplexing a file.

Furthermore, this document is accompanied by a collection of conformance files. These files provide practical demonstrations of various features of ISO/IEC 23090-10, aiding in a more comprehensive understanding and application of ISO/IEC 23090-10.

Information technology — Coded representation of immersive media —

Part 25: Conformance and reference software for carriage of visual volumetric video-based coding data

1 Scope

This document specifies the reference software for carriage of V3C data as specified in ISO/IEC 23090-10. The information provided describes the reference software modules and the features that it supports. It also provides a description of how the reference software can be utilized. Finally, it also provides a description of conformance test vectors.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 23090-10, *Information technology — Coded representation of immersive media — Part 10: Carriage of visual volumetric video-based coding data*

3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

4 Abbreviated terms

ISOBMFF	ISO base media file format
MIV	MPEG immersive video
PCC	point cloud compression
V3C	visual volumetric video-based coding
V-PCC	video-based point cloud compression

5 Reference software for ISO/IEC 23090-10

5.1 General

The reference software and conformance files are available at:

https://standards.iso.org/iso-iec/23090/-25/ed-1/en/

5.2 Architecture

Figure 1 shows a simplified overview of the reference software implementation architecture for the carriage of visual volumetric video-based coding data. The reference software implementation implements the features defined in ISO/IEC 23090-10 and is based on the reference software for ISOBMFF,[2] the reference software for MIV[1] and other supporting libraries. Boxes with a gray background in Figure 1 are part of the scope of the reference software implementation and are described in more detail in the following sections.

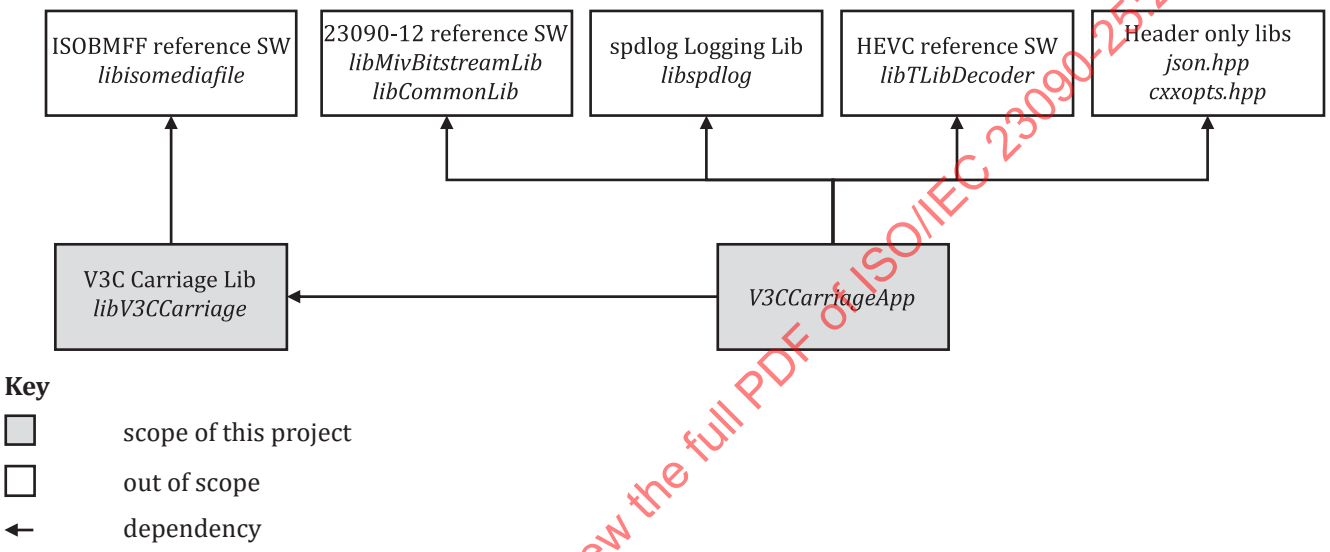

Figure 1 — Architecture overview

The reference software implementation consists of the V3C carriage library *libV3CCarriage*, and the command line application with the name *V3CCarriageApp*. While the library implements an API to parse and write data structures as defined in 23090-10, the command line application uses this API together with other helping libraries to implement actual multiplexing and demultiplexing functionality. The documentation of the API and associated classes and functions is done using Doxygen[3] and can be built by running the Doxygen command in the root folder.

5.3 Features

Table 1 summarizes a list of features adopted in the V3C carriage specification and indicates which features are currently supported by the reference software.

Table 1 — V3C Reference Software Feature Support List

	Feature	4CCs	Version	Status
Common	V3CConfigurationBox	v3cC	0	OK
	V3CUnitHeaderBox	vunt	0	OK
	V3CAtlasParamSampleGroupDescriptionEntry	vaps	0	OK
	ObjectSwitchAlternativesBox	swpc	0	OK
Single track	V3CBitstreamSampleEntry	v3e1 v3eg	0	OK
	Multiplexer	N/A		Not Implemented
	Demultiplexer	N/A		Not Implemented
	Sub-samples	N/A		Not Implemented
Multi track	V3CAtlasSampleEntry	v3c1 v3cg v3cb v3a1 v3ag	0	OK
	V3CAtlasTileConfigurationBox	v3tC	0	OK
	V3CAtlasTileSampleEntry	v3t1	0	OK
	MultiMapVideoBox	mmvi	0	OK
	PlayoutTrackGroupBox	potg	0	OK
	Single atlas multiplexer	N/A		OK
	Single atlas demultiplexer	N/A		OK
	Single atlas tiles multiplexer	N/A		Not Implemented
	Single atlas tiles demultiplexer	N/A		Not Implemented
	Multi atlas multiplexer	N/A		OK
	Multi atlas demultiplexer	N/A		OK
Non-timed	V3CConfigurationProperty	v3cp	0	OK
	V3CUnitHeaderProperty	vutp	0	OK
	V3CAtlasTileConfigurationProperty	v3tp	0	OK
	PlayoutEntityToGroupBox	eply	0	OK
	Single atlas multiplexer	N/A		OK
	Single atlas demultiplexer	N/A		Not Implemented
	Single atlas tiles multiplexer	N/A		Not Implemented
	Single atlas tiles demultiplexer	N/A		Not Implemented
	Multi atlas multiplexer	N/A		OK
	Multi atlas demultiplexer	N/A		OK

Table 1 *(continued)*

	Feature	4CCs	Version	Status
Partial access	Vector3	N/A		OK
	V3CBoundingBox	N/A		OK
	TileMapping	N/A		OK
	V3CObject	N/A		OK
	V3CObjectCollection	N/A		OK
	V3CSpatialRegion	N/A		OK
	V3CTileVideoComponentGroupBox	vtcg	0	OK
	V3CBoundsBox	vpbb	0	OK
	V3CSpatialRegionCollectionBox	v3sc	0	OK
	DynamicVolumetricMetadataSampleEntry	dyvm	0	OK
	Static partial access multiplexer	N/A		Not Implemented
	Static partial access demultiplexer	N/A		Not Implemented
	Dynamic partial access multiplexer	N/A		Not Implemented
	Dynamic partial access demultiplexer	N/A		Not Implemented
Viewport metadata	ExtCameraInfo	N/A		OK
	IntCameraInfo	N/A		OK
	ViewportInfo	N/A		OK
	ViewportInfoConfigurationBox	6vpC	0	OK
	ViewportInfoSampleEntry	6vpt	0	OK
	Viewport track multiplexer	N/A		Not Implemented
	Viewport track demultiplexer	N/A		Not Implemented

5.4 V3C carriage library API

5.4.1 General

This clause contains documentation for the public exported API of the reference software. You are advised to use only the functions documented here.

5.4.2 V3CCreateBox

```
MP4Err V3CCreateBox(u32 type, MP4AtomPtr *pOut)
```

Create a V3C Box of a given type. If type FOURCC is not recognized it will create an `UnknownBox` with the provided type.

Parameters:

`type` FOURCC of the box

`pOut` output Box (if box type is not supported return an unknown box)

5.4.3 V3CCreateAtlasParamSampleGroupDescriptionEntry

```
MP4Err V3CCreateAtlasParamSampleGroupDescriptionEntry
     (V3CAtlasParamSampleGroupDescriptionEntryPtr *pOut)
```

Create a V3C Atlas Parameters Sample Group Description Entry.

Parameters:

`pOut` output instance of V3CAtlasParamSampleGroupDescriptionEntry

5.4.4 V3CAddSetupUnitToAtlasParamSampleGroupEntry

```
MP4Err V3CAddSetupUnitToAtlasParamSampleGroupEntry
     (V3CAtlasParamSampleGroupDescriptionEntryPtr vapsEntry,
     MP4Handle ps)
```

Add setup unit to `V3CAtlasParamSampleGroupDescriptionEntry`. When you are done adding parameter sets to the sample group call `V3CPutAtlasParamSampleGroupEntry` to finally add the entry to `'sgpd'`.

Parameters:

`vapsEntry` `'vaps'` sample group entry to add the NAL unit to

`ps` setup NAL unit (usually atlas parameter set)

5.4.5 V3CPutAtlasParamSampleGroupEntry

```
MP4Err V3CPutAtlasParamSampleGroupEntry(MP4Media media,
      V3CAtlasParamSampleGroupDescriptionEntryPtr vapsEntry, u32 *index)
```

Put `V3CAtlasParamSampleGroupDescriptionEntry` to `'vaps'` sample group. This calls `ISOAddGroupDescription` internally and adds an entry to `'vaps'` sample group.

Parameters:

`media` input media object

`vapsEntry` `'vaps'` sample group entry

`index` output index of the added group entry

5.4.6 V3CGetAtlasParamSampleGroupDescriptionEntryNALCnt

```
MP4Err V3CGetAtlasParamSampleGroupDescriptionEntryNALCnt(MP4Handle
     vapsEntryH, u32 *nalCnt)
```

Get number of NAL units in V3C Atlas Parameters Sample Group Description Entry. Call `ISOGetGroupDescription` to get the handle.

Parameters:

`vapsEntryH` input handle with `V3CAtlasParamSampleGroupDescriptionEntry` data

`nalCnt` output number of NAL units in `'vaps'` Entry

5.4.7 V3CGetNALUnitFromAtlasParamSampleGroupEntry

```
MP4Err V3CGetNALUnitFromAtlasParamSampleGroupEntry(MP4Handle vapsEntryH,
     u32 index, MP4Handle naluH)
```

Get NAL unit from V3C Atlas Parameters Sample Group Description Entry.

Parameters:

`vapsEntryH` input handle with `V3CAtlasParamSampleGroupDescriptionEntry` data

`index` index of NAL unit

`naluH` output NAL Unit handle

5.4.8 V3CNewBitstreamSampleEntry

```
MP4Err V3CNewBitstreamSampleEntry(MP4Track theTrack,
     MP4VolumetricVisualSampleEntryAtomPtr *sampleEntryPtr, u32 inBandFlag,
     u32 dataRefIdx, u32 lengthSizeMinOne)
```

Create a new V3C Bitstream sample entry (single track)

Parameters:

`theTrack`	track to put the sample entry to
`sampleEntryPtr`	output sample entry box
`inBandFlag`	0 - out of band `'v3e1'`, 1 - in band `'v3eg'`
`dataRefIdx`	sample entry data reference index
`lengthSizeMinOne`	length size minus one

5.4.9 V3CNewAtlasSampleEntry

```
MP4Err V3CNewAtlasSampleEntry(MP4Track theTrack,
     MP4VolumetricVisualSampleEntryAtomPtr *sampleEntryPtr, u32 type,
     u32 dataRefIdx, u32 lengthSizeMinOne)
```

Create new Atlas Sample Entry.

Parameters:

`theTrack`	track to put the sample entry to
`sampleEntryPtr`	output sample entry box
`type`	sample entry type
`dataRefIdx`	sample entry data reference index
`lengthSizeMinOne`	length size minus one

5.4.10 V3CNewAtlasTileSampleEntry

```
MP4Err V3CNewAtlasTileSampleEntry(MP4Track theTrack,
     MP4VolumetricVisualSampleEntryAtomPtr *sampleEntryPtr, u32 dataRefIdx,
     u32 lengthSizeMinOne)
```

Create new Atlas Tile Sample Entry.

Parameters:

`theTrack`	track to put the sample entry to
`sampleEntryPtr`	output sample entry box
`dataRefIdx`	sample entry data reference index
`lengthSizeMinOne`	length size minus one

5.4.11 V3CAddV3CParameterSet

```
MP4Err V3CAddV3CParameterSet(MP4VolumetricVisualSampleEntryAtomPtr ptrSE,
     MP4Handle v3cParameterSetH)
```

Add parameter set to Volumetric visual sample entry.

Parameters:

`ptrSE`	volumetric visual sample entry to add parameter sets to
`v3cParameterSetH`	data to add

5.4.12 V3CAddSetupUnit

```
MP4Err V3CAddSetupUnit(MP4VolumetricVisualSampleEntryAtomPtr ptrSE,
     MP4Handle setupUnitH, u32 uiType, u32 array_completeness_flag)
```

Add Atlas setup units to volumetric visual sample entry.

Parameters:

`ptrSE`	volumetric visual sample entry to add atlas setup units to
`setupUnitH`	data to add
`uiType`	type of setup unit

5.4.13 V3CAddUnitHeader

```
MP4Err V3CAddUnitHeader(MP4VolumetricVisualSampleEntryAtomPtr ptrSE,
     MP4Handle v3cUnitHeaderH)
```

Add unit header.

Parameters:

`ptrSE`	volumetric visual sample entry to add unit header data to
`v3cUnitHeaderH`	data to add.

5.4.14 V3CAddTileId

```
MP4Err V3CAddTileId(MP4VolumetricVisualSampleEntryAtomPtr ptrSE, u32 tileId)
```

Add tile id to atlas tile sample entry.

Parameters:

`ptrSE`	volumetric visual sample entry to add unit header data to.
`tileId`	tile id to add.

5.4.15 V3CSetLodIdx

```
MP4Err V3CSetLodIdx(MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr,
     u32 lodIdx)
```

Set level of detail index to atlas tile sample entry.

5.4.16 V3CGetCommonInfoFromSampleEntry

```
MP4Err V3CGetCommonInfoFromSampleEntry(MP4Handle sampleEntryH,
     u32 *dataRefIdx, u32 *lenSizeMinOne, u32 *numVPS,
     u32 *numSetupUnits, u32 *numSetupArrays, u32 *setupSize,
     u32 *vuhPSID, u32 *vuhAtlasID)
```

Get common parameters (number of VPS, precision, etc.) from sample entry.

Parameters:

`sampleEntryH`	Handle with volumetric visual sample entry data
`dataRefIdx`	[out] `data_reference_index` from sample entry (ISOBMFF)
`lenSizeMinOne`	[out] `unit_size_precision_bytes_minus1`
`numVPS`	[out] Number of VPSs in this sample entry
`numSetupUnits`	[out] Number of atlas setup units in all arrays in this sample entry
`numSetupArrays`	[out] Number of atlas setup unit arrays (one array per type)
`setupSize`	[out] Number of payload bytes for all atlas setup units
`vuhPSID`	[out] v3c_unit_header.vuh_v3c_parameter_set_id from `vunt` box
`vuhAtlasID`	[out] v3c_unit_header.vuh_atlas_id from `vunt` box

5.4.17 V3CGetUnitHeaderInfoFromResvSampleEntry

```
MP4Err V3CGetUnitHeaderInfoFromResvSampleEntry(MP4Handle sampleEntryH,
     u32 *unitType, u32 *psID, u32 *atlasID, u32 *attrIdx, u32 *attrPartIdx,
     u32 *mapIdx, u32 *auxVideoFlag)
```

Get V3C unit header information from the restricted video sample entry.

Parameters:

`sampleEntryH`	Restricted video SampleEntry handle
`unitType`	[out] unit type which identifies the type of data
`psID`	[out] `vuh_v3c_parameter_set_id`
`atlasID`	[out] `vuh_atlas_id`
`attrIdx`	[out] attribute index (if applicable)
`attrPartIdx`	[out] attribute partition index (if applicable)
`mapIdx`	[out] map index (if applicable)
`auxVideoFlag`	[out] auxiliary flag (if applicable)

5.4.18 V3CGetV3CParameterSet

```
MP4Err V3CGetV3CParameterSet(MP4Handle seH, MP4Handle psH, u32 idx)
```

Get V3C parameter set from sample entry.

Parameters:

`seH`	handle with volumetric visual sample entry data.
`psH`	[out] handle which is holding the parameter set.
`idx`	the index of the parameter set.

5.4.19 V3CGetSetupUnit

```
MP4Err V3CGetSetupUnit(MP4Handle seH, MP4Handle psH, u32 type, u32 idx,
     u32 *inBandFlag)
```

Get setup unit from sample entry.

Parameters:

`seH`	handle with volumetric visual sample entry data
`psH`	[out] handle which is holding the setup unit.
`type`	type of setup unit (i.e nal unit type)
`idx`	the index of the setup unit.
`inBandFlag`	[out, optional] if output is 1, then NAL units of this type can be expected in samples.

5.4.20 V3CGetUnitHeader

```
MP4Err V3CGetUnitHeader(MP4Handle seH, MP4Handle uhH)
```

Get v3c_unit_header data (4 bytes) from sample entry.

Parameters:

`seH`	Handle with volumetric visual sample entry data.
`uhH`	[out] handle which is holding the unit header.

5.4.21 V3CGetSetupUnitCnt

```
MP4Err V3CGetSetupUnitCnt(MP4Handle seH, u32 type, u32 *cnt_all,
     u32 *cnt_outband, u32 *cnt_inband)
```

Get number of setup units of a given type form sample entry.

Parameters:

`seH`	Handle with volumetric visual sample entry data.
`type`	type of setup unit (i.e NAL unit type)
`cnt_all`	[out, optional] number of setup units (`cnt_outband + cnt_inband`)
`cnt_outband`	[out, optional] number of out-of-band setup units
`cnt_inband`	[out, optional] number of in-band setup units

5.4.22 V3CGetTileIdCnt

```
MP4Err V3CGetTileIdCnt(MP4VolumetricVisualSampleEntryAtomPtr ptrSE,
     u32 *cnt)
```

Get number of tile ids of a `v3t1` sample entry.

5.4.23 V3CGetTileId

```
MP4Err V3CGetTileId(MP4VolumetricVisualSampleEntryAtomPtr ptrSE,
     u32 *tileId, u32 index)
```

Get tile ID from sample entry.

5.4.24 V3CGetLodIdx

```
MP4Err V3CGetLodIdx(MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr,
     u32 *lodIdx, u32 *enabled)
```

Get level of detail index from sample entry.

5.4.25 V3CParseBox

```
MP4Err V3CParseBox(MP4Handle inputHandle, MP4AtomPtr *outBox)
```

Parse a handle into a V3C box.

Parameters:

`inputHandle`	raw data.
`outBox`	[out] parsed box.

5.4.26 V3CParseUnknownBox

```
MP4Err V3CParseUnknownBox(MP4AtomPtr inBox, MP4AtomPtr *outBox)
```

Parse an unknown Box into a V3C box.

Parameters:

`inBox`	raw data.
`outBox`	parsed box.

5.4.27 V3CNewVideoHEVCSampleEntry

```
MP4Err V3CNewVideoHEVCSampleEntry(MP4Track theTrack,
     MP4RestrictedVideoSampleEntryAtomPtr *sampleEntryPtr, u32 type,
     u32 dataRefIdx, u32 lengthSizeMinOne,
     const HEVCDecoderConfigurationRecordPtr record, MP4Handle vuhH,
     u32 mapCountMinusOne)
```

Create new HEVC restricted video sample entry.

Parameters:

`theTrack`	track to put the sample entry to.
`sampleEntryPtr`	output sample entry box.
`type`	sample entry type.
`dataRefIdx`	sample entry data reference index.
`lengthSizeMinOne`	length size minus one.

5.5 Usage of V3CCarriageApp

V3CCarriageApp is a simple console application which is using the libV3CCarriage library and the API described in subclause 5.4 to multiplex and demultiplex V3C bitstreams into ISOBMFF.

Usage:

```
  V3CCarriageApp [OPTION...]

  -h, --help                    Print usage
```

```
  -l, --log_level arg           Logging level 0-6. Default is 4.
  -c, --config arg              JSON Configuration file
  -i, --input arg               Input file name
  -o, --output_file arg         Output file name
  -m, --multiplexer_mode        Execute multiplexing if set.
  -d, --demultiplexer_mode      Execute demultiplexing if set.
  -e, --encapsulation_mode arg  Encapsulation mode when
                                multiplexer_mode is set to true.
                                0 - timed single-track
                                1 - timed multi-track,
                                2 - non-timed
```

5.6 Copyright disclaimer for software modules

Each source code module in this document contains copyright disclaimer, which shall not be removed from the source code module. A generic disclaimer is provided below:

```
The copyright in this software is being made available under the BSD License, included below.
This software may be subject to other third party and contributor rights, including patent
rights, and no such rights are granted under this license.

Copyright (c) 2010-2021, ISO/IEC
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met:
  * Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials provided
with the distribution.
  * Neither the name of the ISO/IEC nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
```

6 Conformance for ISO/IEC 23090-10

6.1 General

This clause documents the status of the conformance files.

6.2 Timed V3C for DASH

6.2.1 General

Table 2 lists and describes each of the characteristics of the sequences used in the conformance files. The source sequences were encoded with TMC2 version 18 using the default configuration for lossy random access (C2RA) and the following encoder options:

— `mapCountMinus1=0`

— `minimumImageHeight=1600`

— `minimumImageWidth=1472`

Table 2 — Characteristics of Source Sequences

ID	Name	Frames	FPS	Geometry Precision	Avg. Points per Frame
1	RedAndBlack	300	30	10 bit	700 000
2	Soldier	300	30	10 bit	1 000 000
3	LongDress	300	30	10 bit	800 000
4	Mitch	450	30	10 bit	900 000
5	Thomas	450	30	10 bit	750 000

All files relevant to a test sequence are stored in a single folder. Each sequence folder is structured as follows:

```
<seq_name>
|
 |- mp4
 |- fmp4
 +- dash
    |- stream.mpd
    |- init.mp4
    |- atlas
    |  +- <seq_name>_atlas-<comp_id>_seg-<seg_number>.m4s
    |
    +- video
       |- occp
       |  +- <seq_name>_occp-<comp_id>_seg-<seg_number>.m4s
       |
       |- geom
       |  +- <seq_name>_geom-<comp_id>_seg-<seg_number>.m4s
       |
       +- attr_<attr_type>
          +- <seq_name>_attr_<attr_type>-<comp_id>_seg-<seg_number>.m4s
```

Where:

— `<seq_name>` is the name of the sequence.

— `<comp_id>` is the id of the V3C component.

— `<seg_number>` is the DASH segment number.

— `<attr_type>` is the type of the V3C attribute component.

Initialization segments use the ".mp4" extension while media segments use the ".m4s" extension. MPD files use the ".mpd" file extension

The conformance files provided for each sequence include: a non-fragmented MP4 file (multi-track), a fragmented MP4 file (multi-track), and MPEG-DASH MPD and associated fMP4 segments (multi-track, live profile).

6.2.2 RedAndBlack

The following features of ISO/IEC 23090-10 are included in the test vector:

— `v3mt` brand

— `volv` media handler

— `vvhd` volumetric visual media header

— `v3c1` sample entry

— `v3cC` V3C configuration

— `vunt` V3C unit header

— `v3vo`, `v3vg`, `v3va` references

— V3C preselection

— V3C descriptor

— V3C video component descriptor

6.2.3 Soldier

The following features of ISO/IEC 23090-10 are included in the test vector:

— `v3mt` brand

— `volv` media handler

— `vvhd` volumetric visual media header

— `v3c1` sample entry

— `v3cC` V3C configuration

— `vunt` V3C unit header

— `v3vo`, `v3vg`, `v3va` references

— V3C preselection

— V3C descriptor

— V3C video component descriptor

6.2.4 LongDress

The following features of ISO/IEC 23090-10 are included in the test vector:

— `v3mt` brand

— `volv` media handler

— `vvhd` volumetric visual media header

— `v3c1` sample entry

— `v3cC` V3C configuration

— `vunt` V3C unit header

— `v3vo`, `v3vg`, `v3va` references

— V3C preselection

— V3C descriptor

— V3C video component descriptor

6.2.5 Mitch

The following features of ISO/IEC 23090-10 are included in the test vector:

— `v3mt` brand

— `volv` media handler

— `vvhd` volumetric visual media header

— `v3c1` sample entry

— v3cC V3C configuration

— vunt V3C unit header

— v3vo, v3vg, v3va references

— V3C preselection

— V3C descriptor

— V3C video component descriptor

6.2.6 Thomas

The following features of ISO/IEC 23090-10 are included in the test vector:

— v3mt brand

— volv media handler

— vvhd volumetric visual media header

— v3c1 sample entry

— v3cC V3C configuration

— vunt V3C unit header

— v3vo, v3vg, v3va references

— V3C preselection

— V3C descriptor

— V3C video component descriptor

6.3 Timed V3C

6.3.1 General

The following files are included for conformance of timed content with multiple video tracks and a single atlas track:

— ./timed_vpcc_01.mp4

— ./timed_vpcc_02.mp4

— ./timed_miv_01.mp4

— ./timed_miv_02.mp4

— ./LongDress/mp4/S26C2RAR04_longdress.mp4

— ./Mitch/mp4/S42C2RAR04_mitch.mp4

— ./RedAndBlack/mp4/S24C2RAR04_redandblack.mp4

— ./Soldier/mp4/S25C2RAR04_soldier.mp4

— ./Thomas/mp4/S43C2RAR04_thomas.mp4

LongDress, Mitch, RedAndBlack, Soldier, and Thomas conformance files also come with a fragmented version of each file in fmp4 subfolder.