ISO/IEC JTC 1/SC 29/WG 03 N0695

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Text of ISO/IEC CD 23090-25 Conformance and reference software for carriage of visual volumetric video-based coding data

**Status:** Approved

**Date of document:** 2022-11-17

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 12 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N0695**

**November 2022, Mainz**

|  |  |
| --- | --- |
| **Title** | **Text of ISO/IEC CD 23090-25 Conformance and reference software for carriage of visual volumetric video-based coding data** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **21974** |

# 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.

[Ed. (DP): The proponents of the technologies are invited to participate in the development process of the software. The development status summary can be found in [this google sheet](https://docs.google.com/spreadsheets/d/1LvHMEu3GxANkG2p4AcYcl58wubYoy8Ef1FT97RSWVgs/edit#gid=0) and in the [GitLab repository](http://mpegx.int-evry.fr/software/MPEG/Systems/PCC-SYS/23090-10-conformance) itself.]

# 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 23090-10:2021, Information technology — Coded representation of immersive media — Part 10: Carriage of visual volumetric video-based coding data*

*ISO/IEC 23090-5:2021, Information technology — Coded representation of immersive media — Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)*

# Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC FDIS 23090-10 and the following apply.

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

* ISO Online browsing platform: available at <https://www.iso.org/obp>
* IEC Electropedia: available at <http://www.electropedia.org>

# Abbreviated terms

For the purposes of this International Standard, the following abbreviations apply:

|  |  |
| --- | --- |
| ISOBMFF | ISO base media file format |
| PCC | point cloud compression |
| V3C | visual volumetric video-based coding |
| V-PCC | video-based Point Cloud Coding |

# Reference software for ISO/IEC 23090-10

## General

The source code for ISO/IEC 23090-10 reference software is available on MPEG’s GitLab server and is accessible to MPEG members via the following URL:

<http://mpegx.int-evry.fr/software/MPEG/Systems/PCC-SYS/23090-10-conformance>

All merge requests should be submitted to the repository on MPEG’s GitLab server after following the contribution guidelines from [CONTRIBUTING.md](http://mpegx.int-evry.fr/software/MPEG/Systems/PCC-SYS/23090-10-conformance/-/blob/master/CONTRIBUTING.md) file.

[Ed. (DP): It could be considered making public mirror for releases on MPEG’s GitHub account once the important merge requests are merged into the master branch.]

## Overview

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 23090-10 [1] and is based on the the reference software for [ISOBMFF](https://github.com/MPEGGroup/isobmff) [3], the reference software for [MIV](https://gitlab.com/mpeg-i-visual/tmiv) [2] 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](https://doxygen.nl/) and can be built by running the doxygen command in the root folder.

## Feature support list

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 |  | In pull request |
| Demultiplexer | N/A |  | In pull request |
| Sub-samples | N/A |  | TBD |
| 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 |  | In pull request |
| Single atlas demultiplexer | N/A |  | In pull request |
| Single atlas tiles multiplexer | N/A |  | TBD |
| Single atlas tiles demultiplexer | N/A |  | TBD |
| Multi atlas multiplexer | N/A |  | TBD |
| Multi atlas demultiplexer | N/A |  | TBD |
| Non-timed | V3CConfigurationProperty | v3cp | 0 | OK |
| V3CUnitHeaderProperty | vutp | 0 | OK |
| V3CAtlasTileConfigurationProperty | v3tp | 0 | OK |
| PlayoutEntityToGroupBox | eply | 0 | OK |
| Single atlas multiplexer | N/A |  | In pull request |
| Single atlas demultiplexer | N/A |  | In pull request |
| Single atlas tiles multiplexer | N/A |  | TBD |
| Single atlas tiles demultiplexer | N/A |  | TBD |
| Multi atlas multiplexer | N/A |  | TBD |
| Multi atlas demultiplexer | N/A |  | TBD |
| 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 |  | TBD |
| Static partial access demultiplexer | N/A |  | TBD |
| Dynamic partial access multiplexer | N/A |  | TBD |
| Dynamic partial access demultiplexer | N/A |  | TBD |
| Viewport metadata | ExtCameraInfo | N/A |  | OK |
| IntCameraInfo | N/A |  | OK |
| ViewportInfo | N/A |  | OK |
| ViewportInfoConfigurationBox | 6vpC | 0 | OK (PR) |
| ViewportInfoSampleEntry | 6vpt | 0 | OK (PR) |
| Viewport track multiplexer | N/A |  | TBD |
| Viewport track demultiplexer | N/A |  | TBD |

## V3C carriage library API

This section contains documentation for the public exported API of the reference software. You are advised to use only the functions documented here. All function prototypes can be found in V3CCarriage.h.

[Ed. (DP): The API should be cleaned up after merging ongoing pull requests. We will need to group API by the level of operations. E.g.: track operations, moovie operations, sample group operations, etc.]

### 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) |

### V3CCreateAtlasParamSampleGroupDescriptionEntry

MP4Err V3CCreateAtlasParamSampleGroupDescriptionEntry (V3CAtlasParamSampleGroupDescriptionEntryPtr \*pOut)

Create a V3C Atlas Parameters Sample Group Description Entry.

**Parameters**

|  |  |
| --- | --- |
| **pOut** | output instance of **V3CAtlasParamSampleGroupDescriptionEntry** |

### 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) |

### 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 |

### 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 |

### 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 in |
| **naluH** | output NAL Unit handle |

### 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 |

### 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 |

### 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 |

### V3CAddV3CParameterSet

MP4Err V3CAddV3CParameterSet (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, MP4Handle v3cParameterSetH)

Add parameter set to Volumetric Visual SampleEntry.

**Parameters**

|  |  |
| --- | --- |
| **sampleEntryPtr** | Volumetric Visual SampleEntry to add parameter sets to |
| **v3cParameterSetH** | data to add |

### V3CAddSetupUnit

MP4Err V3CAddSetupUnit (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, MP4Handle setupUnitH, u32 uiType, u32 array\_completeness\_flag)

Add Atlas setup units to Volumetric Visual SampleEntry.

**Parameters**

|  |  |
| --- | --- |
| **sampleEntryPtr** | Volumetric Visual SampleEntry to add Atlas setup units to |
| **setupUnitH** | data to add |
| **uiType** | type of setup unit |

### V3CAddUnitHeader

MP4Err V3CAddUnitHeader (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, MP4Handle v3cUnitHeaderH)

Add unit header.

**Parameters**

|  |  |
| --- | --- |
| **sampleEntryPtr** | Volumetric Visual SampleEntry to add Unit header data to |
| **v3cUnitHeaderH** | data to add |

### V3CAddTileId

MP4Err V3CAddTileId (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, u32 tileId)

Add tile id to Atlas Tile SampleEntry.

### V3CSetLodIdx

MP4Err V3CSetLodIdx (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, u32 lodIdx)

Set LoD index to Atlas Tile Sample Entry.

### V3CGetCommonInfoFromSampleEntry

MP4Err V3CGetCommonInfoFromSampleEntry (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, u32 \*dataRefIdx, u32 \*lenSizeMinOne, u32 \*numVPS, u32 \*numSetupUnits)

Get common parameters (number of VPS, precision, etc.) from Sample Entry.

**Parameters**

|  |  |
| --- | --- |
| **sampleEntryPtr** | Volumetric Visual SampleEntry to get common parameters from |
| **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 |

### V3CGetV3CParameterSet

MP4Err V3CGetV3CParameterSet (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, MP4Handle ps, u32 index)

Get V3C parameter set from Sample Entry.

**Parameters**

|  |  |
| --- | --- |
| **sampleEntryPtr** | Volumetric Visual SampleEntry to get VPS from |
| **ps** | [out] handle which is holding the parameter set. |
| **index** | the index of the parameter set |

### V3CGetSetupUnit

MP4Err V3CGetSetupUnit (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, MP4Handle ps, u32 type, u32 index, u32 \*inBandFlag)

Get setup unit from Sample Entry.

**Parameters**

|  |  |
| --- | --- |
| **sampleEntryPtr** | Volumetric Visual SampleEntry to get setup unit from |
| **ps** | [out] handle which is holding the setup unit. |
| **type** | type of setup unit (i.e nal unit type) |
| **index** | the index of the setup unit |
| **inBandFlag** | [out, optional] if output is 1, then NAL units of this type can be expected in samples |

### V3CGetUnitHeader

MP4Err V3CGetUnitHeader (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, MP4Handle uh)

Get v3c\_unit\_header data (4 bytes) from Sample Entry.

### V3CGetSetupUnitCnt

MP4Err V3CGetSetupUnitCnt (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, u32 type, u32 \*cnt\_all, u32 \*cnt\_outband, u32 \*cnt\_inband)

Get number of setup units of a given type form Sample Entry.

### V3CGetTileIdCnt

MP4Err V3CGetTileIdCnt (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, u32 \*cnt)

Get number of tile ids of a v3t1 Sample Entry.

### V3CGetTileId

MP4Err V3CGetTileId (MP4VolumetricVisualSampleEntryAtomPtr sampleEntryPtr, u32 \*tileId, u32 index)

Get tile Id from Sample Entry.

### V3CGetLodIdx

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

Get LoD index from Sample Entry.

### V3CParseBox

MP4Err V3CParseBox (MP4Handle inputHandle, MP4AtomPtr \*outBox)

Parse a handle into a V3C Box.

**Parameters**

|  |  |
| --- | --- |
| **inputHandle** | raw data |
| **outBox** | parsed box |

### V3CParseUnknownBox

MP4Err V3CParseUnknownBox (MP4AtomPtr inBox, MP4AtomPtr \*outBox)

Parse an unknown Box into a V3C Box.

**Parameters**

|  |  |
| --- | --- |
| **inBox** | raw data |
| **outBox** | parsed box |

### V3CNewVideoHEVCSampleEntry

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

Create new 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 |

## Usage of V3CCarriageApp

V3CCarriageApp is a simple console application which is using the libV3CCarriage library and the API described in clause 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.  
 -e, --encapsulation\_mode arg Encapsulation mode when multiplexer\_mode is  
 set to true. 0 - timed single-track, 1 -  
 timed multi-track, 2 - non-timed

[Ed. (DP): This section needs to be finalized once we merge all pull requests which are related to multiplexer and demultiplexer.]

## 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 OFTHE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. |

# Conformance for ISO/IEC 23090-10

This clause documents the status of the conformance files.

[Ed. Note: (DP) We need to figure out where to store conformance files and document it in this section.]

# Bibliography

1. ISO/IEC 23090-10 “Information technology — Coded representation of immersive media — Part 10: Carriage of visual volumetric video-based coding data”
2. ISO/IEC 23090-12 "Information technology — Coded Representation of Immersive Media — Part 12: Immersive Video"
3. ISO/IEC 14496-32 "Information technology — Coding of audio-visual objects — Part 32: File format reference software and conformance"