 ISO/IEC JTC 1/SC 29/WG 3 N00544

**ISO/IEC JTC 1/SC 29/WG 3**

**MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** WD of Conformance and Reference Software for Scene Description

**Status:** Approved

**Date of document:** 2022-02-11

**Source:** ISO/IEC JTC 1/SC 29/WG 3

**Expected action:** ACT

**Action due date:** 2022-02-11

**No. of pages:** 16 (with cover page)

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

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

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 3**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 3 N** **00544**

**Online – April 2022**

|  |  |
| --- | --- |
| **Title** | **WD of Conformance and Reference Software for Scene Description** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **21437** |

# Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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 ISO documents 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](https://www.iso.org/directives-and-policies.html)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](https://www.iso.org/iso-standards-and-patents.html)).

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](https://www.iso.org/foreword-supplementary-information.html).

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 website.

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](https://www.iso.org/members.html).

# Introduction

The conformance and reference software of ISO/IEC 23090-14 serves following main purposes:

* Validation of the written specification of the parts of ISO/IEC 23090-14;
* Clarification of the written specification of the parts of ISO/IEC 23090-14;
* Conformance testing for checking interoperability for the various applications against the reference software which aims to be complaint with ISO/IEC 23090-1

**Contents**

[1 Scope 5](#_Toc99959953)

[2 Normative references 5](#_Toc99959954)

[3 Terms, definitions, symbols, and abbreviated terms 5](#_Toc99959955)

[4 Reference software for ISO/IEC-23090-14 MPEG-I scene description 5](#_Toc99959956)

[4.1. General 6](#_Toc99959957)

[4.2. Description 6](#_Toc99959958)

[4.3. Dependencies 7](#_Toc99959959)

[4.3.1. Spatial audio 7](#_Toc99959960)

[**4.3.2.** OpenXR 7](#_Toc99959961)

[4.4. Usage 7](#_Toc99959962)

[4.4.1. Installation 7](#_Toc99959963)

[4.5. Support for MIV content 8](#_Toc99959964)

[4.5.1 Build instructions 8](#_Toc99959965)

[4.5.2 Run instructions 9](#_Toc99959966)

[4.5.3 Location 9](#_Toc99959967)

[5 ISO/IEC-23090-14 MPEG-I scene description conformance 9](#_Toc99959968)

[5.1. Overview 9](#_Toc99959969)

[5.2. glTF-2.0 and MPEG extension schema validation 10](#_Toc99959970)

[5.2.1. General 10](#_Toc99959971)

[5.2.2. JSON schema validation 10](#_Toc99959972)

[5.2.3. Binary buffer validation 11](#_Toc99959973)

[5.2.4. Source image validation 11](#_Toc99959974)

[5.2.5. Audio validation 11](#_Toc99959975)

[5.3. Scene description metadata sample conformance 11](#_Toc99959976)

[6 Conformance software for ISO/IEC 23090-14 MPEG-I scene description 11](#_Toc99959977)

[6.1. Overview 11](#_Toc99959978)

[7.1 6.2. glTF-validator 11](#_Toc99959979)

[6.3. Carriage library 12](#_Toc99959980)

[6.4. Software 13](#_Toc99959981)

[6.4.1. MPEG scene description glTF-validator 13](#_Toc99959982)

[6.4.2. MPEG scene description carriage library 13](#_Toc99959983)

[7. Test vectors 13](#_Toc99959984)

[7.1 MIV bitstream 13](#_Toc99959985)

Information technology — Coded representation of immersive media — Part 24: Conformance and reference software for Scene Description

# Scope

This document specifies the conformance and reference software implementing the normative clauses of ISO/IEC 23090‑14.

# 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.

*glTF 2.0 Khronos Group, The GL Transmission Format (glTF) 2.0 Specification*, Available at <https://github.com/KhronosGroup/glTF/tree/master/specification/2.0/>

IEEE 754-2019, *IEEE Standard for Floating-Point Arithmetic*

*IETF RFC 6381, The ‘Codecs’ and ‘Profiles’ Parameters for “Bucket” Media Types*

*IETF RFC 6902 (April 2013), JavaScript Object Notation (JSON) Patch*

*ISO/IEC 14496-12:2020, Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

*ISO/IEC 19516, Information technology — Object management group — Interface definition language (IDL) 4.2*

*ISO/IEC 23001-15, Information technology — MPEG systems technologies — Part 15: Carriage of web resources in ISOBMFF*

*ISO/IEC 21778:2017, Information technology — The JSON data interchange syntax*

*ISO/IEC 23090-12:2021, Information technology – Coded representation of immersive media- Part 12: MPEG Immersive Video*

# Terms, definitions, symbols, and abbreviated terms

For this document, the terms, definition and symbols and abbreviated terms given in the ISO/IEC 23090-14 apply.

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

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

# Reference software for ISO/IEC-23090-14 MPEG-I scene description

## General

The reference software is accessible through the MPEG GitLab server at [http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/mpegtrimesh.git](https://protect-us.mimecast.com/s/JZEGCo2W9Ru6Y6f1o7tT?domain=mpegx.int-evry.fr).

## Description

The MPEG scene description reference software is called **mpegtrimesh** and is based on trimesh library. The trimesh library is open-source library and is available at <https://github.com/mikedh/trimesh>. mpegtrimesh extends the trimesh library to support timed data and is responsible for expected rendering process of the timed data after it is decoded.

The implementation in mpegtrimesh support features which are defined as glTF extensions as shown in table 1., for details see ISO/IEC 23090-14 Clause 5. “mpegtrimesh” is implemented using Python language.

Table 1. glTF 2.0 extensions in ISO/IEC 23090-14

|  |  |  |  |
| --- | --- | --- | --- |
| **Extension name** | **Brief description** | **JSON schema link** | **Implementation status** |
| MPEG\_media | Extension for referencing external media sources. | [MPEG\_Media schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_media/schema) | Yes |
| MPEG\_accessor\_timed | An accessor extension to support timed media. | [MPEG\_accessor\_timed schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_accessor_timed/schema) | Yes |
| MPEG\_buffer\_circular | A buffer extension to support circular buffers. | [MPEG\_buffer\_circular schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_buffer_circular/schema) | Yes |
| MPEG\_scene\_dynamic | An extension to support scene updates. | [MPEG\_scene\_dynamic schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_scene_dynamic/schema) | No |
| MPEG\_texture\_video | A texture extension to support video textures. | [MPEG\_texture\_video schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_texture_video/schema) | Yes |
| MPEG\_mesh\_linking | An extension to link two meshes and provide mapping information | [MPEG\_mesh\_linking schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_mesh_linking/schema) | No |
| MPEG\_audio\_spatial | Adds support for spatial audio. | [MPEG\_audio\_spatial schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_audio_spatial/schema) | Yes |
| MPEG\_viewport\_recommended | An extension to describe a recommended viewport. | [MPEG\_viewport\_recommended schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_viewport_recommended/schema) | No |
| MPEG\_animation\_timing | An extension to control animation timelines. | [MPEG\_animation\_timing schema](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/Specification/23090-14/-/tree/master/Extensions/MPEG_animation_schema/schema) | No |

A new feature proposal for ISO/IEC 23090-14 requires a test scenario as well as an implementation which demonstrates proper rendering of the test scenario scene. The implementations should be provided as a new branch with a merge request to reference software for ISO/IEC 23090-14.

## Dependencies

The following list of library packages are dependencies for mpegtrimesh.

|  |  |  |
| --- | --- | --- |
| **Library Package** | **Description** | **Version** |
| av | python binding to FFMPEG library | 8.0.2 |
| decorator | memoizing function | 4.4.2 |
| future | compatibility between Python 2 and Python3 | 0.18.2 |
| network | Creating and manipulating graphs and network | 2.5 |
| numpy | Array computation | 1.19.2 |
| pillow | Imaging processing library | 8.0.1 |
| pyglet | Windowing and multimedia library | 1.5.8 |
| scipy | Fast N-dimensional array manipulation | 1.5.3 |
| bitstream | Manage binary data as bitstreams | 2.6.0 |

### **Spatial audio**

To run test vectors for spatial demos, Soloud audio library: https://sol.gfxile.net/soloud/ is used which is a spatial audio renderer.

* + 1. **OpenXR**

TBD

## Usage

### Installation

The source code for mpegtrimesh can be clone the repository.

Use the following command to clone the scene description reference software:

git clone <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/mpegtrimesh.git>

The recommended way to installing the dependencies is to use virtual environment in python. The module used to manage and create the virtual environment in Python is *venv*: <https://docs.python.org/3/library/venv.html>. is used to manage Python packages for different projects. Using virtual environment allows to avoid installing Python packages globally which could break system tools or other projects. To install virtual environment on a specific platform, see the following command line inputs:

* macOS and Linux:

python3 -m pip install --user virtualenv

* Windows:

py -m pip install --user virtualenv

To activate a virtual environment, see the following the command line input.

python3 -m venv .venv

source .venv/bin/activate

To install the dependencies as mentioned in clause 4.3.

(.venv) python3 -m pip install -r ./requirements.txt

## Support for MIV content

NOTE: this is a Work in progress. The progress of the integration of MIV player to mpegtrimesh is captured in EE3.

A MPEG Immersive Video (MIV) player is capable of parsing MIV bitstream and perform shading operations using OpenGL ES 3.2. The MIV player is based on TMIV11.0 which is compatible with MIV version 1 standard. There is not 3D reconstruction process within MAF engine since this not needed for MIV bitstreams. Figure 1 shows different processing pipelines for MIV Main anchor A and MIV Extended Frame Packing Anchor P.

Diagram, schematic

Description automatically generated

Figure 1 Pipeline options for A300-D bitstream (top) and P300-D bitstream (bottom)

### Build instructions

The build instructions are documented in <http://mpegx.int-evry.fr/software/MPEG/MIV/other/miv-player-example/README.md>

### Run instructions

To test different bitstreams and feature, start

run.bat

which runs various test cases for different number of input frames (1, 17, 300) and posetraces (p01, p02, p03)

Running Museum bitstream (of single atlas that has AVD and GVD) 🡺 2 decoders



Running Painter bitstream (of two atlases, each has AVD and GVD) 🡺 4 decoders



Running Museum bitstream (of single atlas that has PVD) 🡺 1 decoder



Running Painter bitstream (of two atlases, each has PVD) 🡺 2 decoders



### Location

The MIV player software is accessible at: <http://mpegx.int-evry.fr/software/MPEG/MIV/other/miv-player-example>

## Support for Haptics extension

In the interests of compatibility with the glTF extension work already done by MPEG-I SD, the two haptics-related glTF extensions, will be implemented in Trimesh. Specifically:

* MPEG\_haptic: This will be implemented as a Trimesh (python) wrapper around the MPEG Haptics Phase 1 codec. It will be able to ingest haptic media in AHAP, WAV, or IVS format and output .gmpg (human-readable interchange format) and .mpg (binary distribution format) data.
* MPEG\_material\_haptic: This will also be implemented in Trimesh. As per the semantics table in Section 3.3 and Figure 1 of [2], this extension will take as input a location on the mesh and return either a value or a reference to a Haptics Phase 1 media source (haptic data to be rendered).

This implementation mechanism decouples the implementation of Hatpic extensions (in Trimesh) from their evaluation (which require more advanced platforms such as Unity or Unreal. The MPEG\_avatar and MPEG\_interactivity extensions will need to adopt a similar approach since Trimesh does not provide the features required to evaluate them.

Rendering of haptic media cannot be done in Trimesh. Therefore, evaluation of these extensions will require a haptics platform such as that provided by InterHaptics and a game engine platform such as Unity. Real-time rendering of the haptic media will be done on the InterHaptics platform and Unity will provide support for haptic devices and interactivity.

# ISO/IEC-23090-14 MPEG-I scene description conformance

## Overview

Clause 5 of ISO/IEC 23090-14 defines features which describe methods on retrieving timed data and expecting rendering process. These features are defined as MPEG extensions to Khronos gltf2.0. The extensions are under vendor-specific extensions namespaces with an MPEG prefix.

A glTF2.0 file conforming to ISO/IEC 23090-14 obeys the rules for the glTF2.0 and the MPEG defined scene description extensions ISO/IEC 23090-14 Clause 5 and the associated media file containing samples defined in ISO/IEC 23090-14 Clause 8. To conform to MPEG scene description, the following steps need to be completed:

* conformance of glTF2.0 with MPEG extensions according to clause 5.3,
* conformance of samples defined in MPEG scene description according to clause 5.4.

Graphical user interface, diagram

Description automatically generated

Figure 2 MPEG scene description conformance suite

## glTF-2.0 and MPEG extension schema validation

### General

glTF 2.0 defines an extension mechanism which allows to extend glTF 2.0 with new capabilities. ISO/IEC 23090-14 Clause 5 defines extensions which enable support for timed and MPEG media. The MPEG extensions JSON schemas are defined according to ISO/IEC 21779:2107.

A glTF 2.0 with MPEG extensions conforming to ISO/IEC 22090-14 shall obey the glTF 2.0 defined JSON schemas as well as MPEG defined extension JSON schemas. The JSON schemas for the extensions are specified in ISO/IEC 23090-14 as indicated in the Table 1.

MPEG Scene Description *glTF-validator* tool (clause 6.2) extends Khronos glTF-validator tool <https://github.com/KhronosGroup/glTF-Validator>, to support validation of MPEG extensions.

### JSON schema validation

This clause specifies the glTF 2.0 conformance checking. The corresponding software tools and modules which are used for glTF 2.0 with MPEG extensions is provided in clause 6.2.

Application

Description automatically generated with medium confidence

Figure 3 gltf2.0 validation

The glTF 2.0 validator takes a glTF 2.0 file as input and performs JSON schema validation. The validator takes an instance of the glTF 2.0 file and compares it against the schema defined in *gltf.schema.json* as well as *mpeg.schema.json*.

### Binary buffer validation

The glTF 2.0 validator tool ensures correct accessor values, i.e., quaternions, matrixes, animation sampler input and output, etc.

### Source image validation

The glTF 2.0 validator tool ensures and validates correctness of standard texture images referred by the glTF file.

### Audio validation

TBD

## Scene description metadata sample conformance

MPEG Scene Description carriage library *libSDCarriage* (clause 6.2.) extends ISOBMFF library version 0.1.0 <https://github.com/MPEGGroup/isobmff/releases/tag/v0.1.0> and implements the carriage of the metadata samples as specified in ISO/IEC 23090-14.

# Conformance software for ISO/IEC 23090-14 MPEG-I scene description

## Overview

The glTF 2.0 validator is a tool from Khronos to validate glTF assets against the gltf2.0 specification <https://github.com/KhronosGroup/glTF/tree/master/specification/2.0>. The tool checks the JSON syntax as well the binary representation of glTF 2.0; i.e., glbV2. The tool provides validation for binary buffers, images and extensions defined by Khronos.

The software tool is written in Dart programming language. The tool can be used as a command line application or a web-based tool. The tool outputs a validation report in JSON format with potential issues with a glTF 2.0 asset. The issues identified by the glTF-validator tool are available at: <https://github.com/KhronosGroup/glTF-Validator/blob/master/ISSUES.md>.

A high-level design for the MPEG-I scene description validation and conformance software is shown in Figure 4. The conformance software extends the capabilities of the glTF-Validator to support MPEG extensions. The conformance software also includes a C library for MPEG scene description carriage which is used by the glTF-validator to validate the metadata samples.

## 6.2. glTF-validator

The glTF-validator tool reads a glTF file with MPEG extensions and performs a schema check against the extensions used. Upon a validation check, the tool generates a validation report. The validation report lists potential issues and their severity in the glTF asset as shown in Figure 4.

**Diagram

Description automatically generated with medium confidence**

Figure 4 A high-level diagram of the MPEG-I scene description conformance software

MPEG glTF validator git repository contains the source code to validate the MPEG extensions. With code patching as represented in Figure 5 , the source code for MPEG extensions is merged with Khronos glTF validator tool to create a tool for glTF-validator tool with the ability to validate MPEG extensions as well.

Graphical user interface, application

Description automatically generated

Figure 5 Compilation process of MPEG glTF-validator tool

## Carriage library

The *libSDCarriage* library is implemented in C programming language. The MPEG-I scene description glTF validator tool calls the native C APIs from *libSDCarriage* library to read a media sample and validates the media sample. A diagram for MPEG-I Scene description carriage. Figure 6 depicts the integration of MPEG-I scene description carriage library to MPEG-I scene description glTF-validator.

Diagram

Description automatically generated

Figure 6 MPEG Scene description carriage library and glTF-validator integration

## Software

### MPEG scene description glTF-validator

MPEG glTF-validator is accessible at <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/23090-24-gltf-validator> .

#### Implementation Status

The `main` branch is the legacy branch which was provided as an input in m58431. The project was moved from Gitlab.com to MPEG-Git. Currently, there two open `WIP` branches.

* + 1. `updates` branch

The ‘updates’ branch implements the changes performed on various syntax elements during the editing phase of first edition of ISO/IEC 23090-14.

* + - 1. *Change log:*

|  |  |  |  |
| --- | --- | --- | --- |
| File name | Status | Changes | |
| Added | Removed |
| mpeg\_acccessor\_timed.dart | M | * `bufferView` property | * `timedAccessor` property |
| mpeg\_buffer\_circular.dart | M | * `media` property | * `headerLength` property * `source` property |
| mpeg\_media.dart | M | * `name` property * `extraParams` property | * `unifiedTimeline` property |
| mpeg\_texture\_video.dart | M | * Format property * List of formats | * Default values for width and height |

* + 1. `5-MPEG\_animation\_timing` branch

The ‘5-MPEG\_animation\_timing’ branch implements the `MPEG\_animation\_timing`. A merge-request for this branch is open against the `main` branch.

* + - 1. *Change log:*

|  |  |  |  |
| --- | --- | --- | --- |
| File name | Status | Changes | |
| Added | Removed |
| mpeg\_animation\_timing.dart | A | * Implementation of `MPEG\_animation` timing extension |  |
| mpeg\_accessor\_timed.dart | M | * Logic for `bufferView` and `updateRate` in case of `immutable` is true |  |
| diff.patch | M | * Added the import of mpeg\_animation\_timing.dart |  |

* + Open Issues

The implementation for the following extensions is missing

1. MPEG\_viewport\_recommended
2. MPEG\_audio\_spatial
3. MPEG\_mesh\_linking
4. MPEG\_scene\_dynamic

### **MPEG scene description carriage library**

MPEG scene description carriage library is accessible at <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/23090-24-carriage>.

**Implementation status**

The `master` branch is the legacy branch which was provided as an input in m58431. The project was moved from Gitlab.com to MPEG-Git. Currently, there is one open `WIP` branch.

* + 1. `m58429` branch

|  |  |  |  |
| --- | --- | --- | --- |
| File name | Status | Changes | |
| Added | Removed |
| MetadataTrackBase.cpp  MetadataTrackBase.h | A | * Base class for metadata tracks |  |
| AnimationTrack.cpp  AnimationTrack.h | A | * sample entry |  |
| GLTFTrack.cpp  GLTFTrack.h | A | * sample entry |  |
| SDCarriageApp.cpp | M | * Minor |  |
| GLTFPatchConfigBox.h | A | * Initial implementation to on glTF Patch config Box |  |
| SDBoxes.h | A |  |  |
| SDCarriage.h  SDCarriage.cpp | M | Interface declaration and definition   * SDParseBoxUsingInputStream * SDCreateGLTFPatchConfigBox * SDCreateBox * SDNewGLTFSampleEntry * SDNewAnimationSampleEntry * SDParseBox * SDParseUnknownBox |  |
| test\_animationTrack.cpp | A | Test   * Check Static information |  |
| test\_GLTFTrack.cpp | A | Test   * Check Status information |  |
| test\_GLTFtrackConfigBox.xpp | A | Test   * Check Status information * Check Box size and serialization * Check set/get update mode value, * Check box creation from memory |  |
| test\_data.h | A | Random test data |  |
| test\_helpers.h | A | Helper functions |  |
| test\_main.cpp | A |  |  |
| .gitmodules | M | Added submodules   * spdlog * cxxopts * json |  |
| CMakeLists.txt | M | Added submodules |  |
| DoxyFile | A | Automatic documentation generation tool |  |
| README.md | M | Updates |  |

# Test vectors

The test vectors for ISO/IEC 23090–14 are available at <https://mpegfs.int-evry.fr/ws-mpegcontent/MPEG-I/Part14-SceneDescriptions>. Please ask the coordinator for an access to the MPEG-content server.

## MIV bitstream

Three MIV bitstreams in different formats are available. These bitstreams are provided to support experiments in MPEG-I scene description to facilitate support for MPEG-I codecs and validation.

* Single Track: Single bitstream (which includes all V3C units for compressed video and non-video ones for all atlases and IRAPs).
* Multi Track: Intermediate bitstream (which includes all non-video V3C units for all atlases and IRAPs) + separate compressed video sub-bitstreams.
* Decoded data: Intermediate bitstream (which includes all non-video V3C units for all atlases and IRAPs) + separate raw YUV videos.

Sample bitstreams and decoded data are available on the MPEG content server: