Text

Description automatically generated ISO/IEC JTC 1/SC 29/WG 3 N00972

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

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

**Document type:** Output Document

**Title:** Potential information of ISO/IEC DIS 23090-24 Conformance and reference software for scene description

**Status:** Approved

**Date of document:** 2023-08-11

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

**Expected action:** ACT

**Action due date:** 2023-08-11

**No. of pages:** 19 (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** **00** **972**

**Geneva, CH – July 2023**

|  |  |
| --- | --- |
| **Title** | **Potential improvements of ISO/IEC DIS 23090-24 Conformance and reference software for scene description** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **22966** |

**ISO/IEC FDIS 23090-24**

ISO/IEC TC JTC 1/SC 29/WG3

Secretariat: JISC

**Information technology — Coded representation of immersive media — Part 24: Conformance and reference software for scene description**

DIS stage

**Warning for WDs and CDs**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

*To help you, this guide on writing standards was produced by the ISO/TMB and is available at* [*https://www.iso.org/iso/how-to-write-standards.pdf*](https://www.iso.org/iso/how-to-write-standards.pdf)

*A model manuscript of a draft International Standard (known as “The Rice Model”) is available at* [*https://www.iso.org/iso/model\_document-rice\_model.pdf*](https://www.iso.org/iso/model_document-rice_model.pdf)

© ISO 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

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

# 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 23090-14

Contents

[Foreword 3](#_Toc126332336)

[Introduction 3](#_Toc126332337)

[1 Scope 6](#_Toc126332338)

[2 Normative references 6](#_Toc126332339)

[3 Terms, definitions, symbols, and abbreviated terms 6](#_Toc126332340)

[4 Reference software for ISO/IEC-23090-14 MPEG-I scene description 6](#_Toc126332341)

[4.1. General 7](#_Toc126332342)

[4.2. Description 7](#_Toc126332343)

[4.3. Dependencies 8](#_Toc126332345)

[4.3.1. Spatial audio 9](#_Toc126332346)

[4.3.2. OpenXR 9](#_Toc126332348)

[4.4. Usage 9](#_Toc126332349)

[4.4.1. Installation 9](#_Toc126332350)

[4.5. Support for MIV content 10](#_Toc126332352)

[4.5.1. Build instructions 10](#_Toc126332354)

[4.5.2. Run instructions 10](#_Toc126332356)

[4.5.3. Location 11](#_Toc126332357)

[4.6. Support for Haptics extension 11](#_Toc126332360)

[4.7. Support for AR scene playback 11](#_Toc126332361)

[4.8. Issues 12](#_Toc126332362)

[4.9. License 12](#_Toc126332363)

[5 ISO/IEC-23090-14 MPEG-I scene description conformance 12](#_Toc126332364)

[5.1. Overview 12](#_Toc126332365)

[5.2. glTF 2.0 and MPEG extension schema validation 13](#_Toc126332366)

[5.2.1. General 13](#_Toc126332367)

[5.2.2. JSON schema validation 13](#_Toc126332369)

[5.2.3. Binary buffer validation 13](#_Toc126332370)

[5.2.4. Source image validation 13](#_Toc126332372)

[5.2.5. Audio validation 14](#_Toc126332374)

[5.3. Scene description metadata sample conformance 14](#_Toc126332376)

[6 Conformance software for ISO/IEC 23090-14 MPEG-I scene description 14](#_Toc126332377)

[6.1. Overview 14](#_Toc126332378)

[6.2. glTF-validator 14](#_Toc126332380)

[6.2.1. Overview 14](#_Toc126332381)

[6.2.2. Software repository 15](#_Toc126332382)

[6.3. Carriage library 15](#_Toc126332385)

[6.3.1. Overview 15](#_Toc126332386)

[6.3.2. Software respository 15](#_Toc126332387)

[6.4. License 15](#_Toc126332390)

[7. Test vectors 16](#_Toc126332391)

[7.1. MIV bitstream 16](#_Toc126332393)

[8. References 17](#_Toc126332394)

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 from the link below: <https://standards.iso.org/iso-iec/23090/-14/ed-1/en/>.

[ Note: For the latest release and updates, please refer to the reference software which is accessible through the MPEG GitLab server at <http://mpeg.expert/software/MPEG/Systems/SceneDescription/software/reference.git>. ]

An architectural diagram for the reference software is shown in Figure 1.

Diagram

Description automatically generated

Figure 1 pympegsd package

## Description

The MPEG scene description reference software is a standalone python module called ***pympegsd.*** The module contains the reference implementation for MPEG-I scene description extensions as specified in ISO/IEC 23090-14. The implementation in *pympegsd* supports features which are defined as glTF extensions under MPEG namespace as shown in table 1., for details see ISO/IEC 23090-14 Clause 5. *pympegsd* is implemented using Python language.

*Note: The support for carriage formats as specified in clause 7 of ISO/IEC 23090-14 is in progress.*

The *pympegsd* has external dependencies such as *trimesh* and others. *pympegsd* supports timed data. The *pympegsd* interacts with *trimesh* and *pympegsd* which are responsible for expected rendering process of the timed data after it is decoded via operation supported through pipelines.

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

|  |  |  |
| --- | --- | --- |
| **Extension name** | **Brief description** | **Implementation status** |
| MPEG\_media | Extension for referencing external media sources. | Yes |
| MPEG\_accessor\_timed | An accessor extension to support timed media. | Yes |
| MPEG\_buffer\_circular | A buffer extension to support circular buffers. | Yes |
| MPEG\_scene\_dynamic | An extension to support scene updates. | No |
| MPEG\_texture\_video | A texture extension to support video textures. | Yes |
| MPEG\_mesh\_linking | An extension to link two meshes and provide mapping information | No |
| MPEG\_audio\_spatial | Adds support for spatial audio. | Yes |
| MPEG\_viewport\_recommended | An extension to describe a recommended viewport. | Yes |
| MPEG\_animation\_timing | An extension to control animation timelines. | 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 *pympegsd*.

|  |  |  |
| --- | --- | --- |
| **Library Package** | **Description** | **Version** |
| av | Python binding to FFmpeg library | 9.2.0 |
| networkx | Creating and manipulating graphs and network | 2.5 |
| 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 |
| trimesh | Loading and using triangular meshes | 3.5.6 |
| pyopenxr | Python binding for OpenXR SDK | 1.0.2301 |
| pyrender | Rendering engine | 0.1.45 |
| aiortc | Library for real-time communication | 1.3.2 |
| aiohttp | Asynchronous HTTP client/server | 3.8.1 |

### **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**

*pyopenxr* is a python module which provides python binding to Khronos OpenXR SDK. OpenXR backend allows for rendering a scene to any OpenXR compatible device. The dependencies for running the renderer need to be installed first.

On Windows platforms, any compatible device that is registered as the default OpenXR device may be used to render the scene. The recommended device is Oculus Quest 2.

On Linux platforms, the Monado service may be used to simulate or render on compatible XR devices.

The following command line may be used to render on an OpenXR-compatible device:

python3 ./rendeerer.py ./path/to/scene.gltf [--xr]

## Usage

### Installation

The source code for *pympegsd* is available in a compressed zip package as indicated in Section 4.1.

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* [[1]](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 -e .

## Support for MIV content

An MPEG Immersive Video (MIV) player that is capable of parsing MIV bitstreams and perform shading operations using OpenGL ES 3.2 is provided with the reference software. The MIV player is based on TMIV 11.0 which is compatible with MIV version 1 standard. There is no 3D reconstruction process for MIV within the MAF engine since this is not needed in the case of MIV bitstreams. Figure 2 shows different processing pipelines for the MIV Main anchor A and the MIV Extended Frame Packing Anchor P.

Diagram, schematic

Description automatically generated

Figure 2 Pipeline options for A300-D bitstream (top) and P300-D bitstream (bottom)

### Build instructions

The build instructions are documented in [[2].](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 pose traces (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 the [2].

## 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 *pympegsd*. Specifically:

* *MPEG\_haptic*:

This will be implemented as a *pympegsd* (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 *pympegsd*. 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 Haptic extensions (in *pympegsd*) from their evaluation, which requires more advanced platforms such as Unity or Unreal. The MPEG\_avatar and interactivity extensions will need to adopt a similar approach since *pympegsd* does not provide the features required to evaluate them.

Rendering of haptic media cannot be done in *pympegsd*. Therefore, evaluation of these extensions will require a haptics platform. Real-time rendering of haptic media will be done on the haptics platform which provides support for haptic devices and interactivity.

* 1. **Support for AR scene playback**

Augmented reality device application programming interface such as ARCore API [11] provides the ability to record all the information pertaining to an AR session in terms of sensor data and user events.

From such a file, it should then be possible to:

1. Determine the position of the smartphone camera over time (even absolute if GPS activated) using the rotation and displacement data.

2. Create a point cloud frame or mesh frame from each recorded video frame based on the associated depth map.

*NOTE: If the depth sensor is not used for the recording, the depth map should either be generated via an algorithm (out of scope) or retrieved from the ARCore API and stored in an MP4 file using a custom-made application.*

3. Position this point cloud frame or mesh frame in the scene over time.

Once volumetric data corresponding to the AR Session is generated, it could constitute an AR test asset for MPEG-I Scene Description work which could be then played back in *pympegsd*.

* 1. **Issues**

Issues related to the reference software are tracked in [3].

## License

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) <YEAR>, ISO/IEC SC29 WG03

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 <ORGANIZATION> 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.

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

## Overview

Clause 5 of ISO/IEC 23090-14 defines features which describe methods for retrieving timed data and expected rendering process. These features are defined as MPEG extensions to Khronos glTF 2.0. The extensions are under vendor-specific extensions namespaces with an MPEG prefix.

A glTF 2.0 file conforming to ISO/IEC 23090-14 obeys the rules for the glTF 2.0 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 glTF 2.0 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 3 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 [4], 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 4 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

In order to validate audio streams for the glTF 2.0 files that have the MPEG\_audio\_spatial extension as defined in ISO/IEC 23090-14 listed in their extensionsRequired or extensionsUsed , the following applies:

1. All audio sources listed in the MPEG\_audio\_spatial extension shall conform to their respective codec listed in the MPEG\_media.media.alternative.track.codecs. Conformance testing for audio codec is subject to the codec in use.
2. All audio sources listed in the MPEG\_audio\_spatial extension should be playable by libav library.

## Scene description metadata sample conformance

MPEG Scene Description carriage library *libSDCarriage* (clause 6.2.) extends the ISOBMFF library version 0.1.0 [5] 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 [[6]](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 in [7].

A high-level design for the MPEG-I scene description validation and conformance software is shown in Figure 5. 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.

## glTF-validator

* + 1. **Overview**

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 5

**Diagram

Description automatically generated with medium confidence**

Figure 5 A high-level diagram of the MPEG-I scene description conformance software

The MPEG glTF validator Git repository contains the source code to validate the MPEG extensions. With code patching as represented in Figure 6 , the source code for MPEG extensions is merged with the 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 6 Compilation process of MPEG glTF-validator tool

### Software repository

The MPEG glTF-validator is accessible is accessible from the link below: <https://standards.iso.org/iso-iec/23090/-14/ed-1/en/>.

[ Note: For the current release/update, please refer to the MPEG glTF-validator repository which is accessible at <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/23090-24-gltf-validator>. ]

## Carriage library

### Overview

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 7 depicts the integration of MPEG-I scene description carriage library to MPEG-I scene description glTF-validator.

Diagram

Description automatically generated

Figure 7 MPEG Scene description carriage library and glTF-validator integration

### **Software respository**

The MPEG SD carriage library is accessible is accessible from the link below: <https://standards.iso.org/iso-iec/23090/-14/ed-1/en/>.

[ Note: For the current release/update, please refer to the MPEG carriage library repository which is accessible at <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/23090-24-carriage>] .

## License

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) <YEAR>, ISO/IEC SC29 WG03

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 <ORGANIZATION> 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.

# Test vectors

The test vectors for ISO/IEC 23090–14 are available at [[8]](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:

The bitstreams are accessible at /MPEG-I/Part12-ImmersiveVideo/test\_material/m58438\_ExtendingMivPlayerExample folder

* Single-file bitstreams  /bitstreams/SingleTrack
* Intermediate bitstreams + compressed video sub-bitstreams  /bitstreams/MultiTrack
* Intermediate bitstreams + decoded videos  /data
* Sample TMIV encoder configs  /TMIVConfigs

# References

1. Python venv, Online, <https://docs.python.org/3/library/venv.html>
2. MIV player example, Online, <http://mpegx.int-evry.fr/software/MPEG/MIV/other/miv-player-example/README.md>
3. *pympegsd* GitLab issue tracker, Online, <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/reference/-/issues>
4. Khronos glTF validator, Online, <https://github.com/KhronosGroup/glTF-Validator>
5. ISOBMFF reference software: tag v0.1.0, Online, <https://github.com/MPEGGroup/isobmff/releases/tag/v0.1.0>
6. Khronos glTF2.0 specification, Online, <https://github.com/KhronosGroup/glTF/tree/master/specification/2.0>
7. Khronos glTF validator issues list, Online, <https://github.com/KhronosGroup/glTF-Validator/blob/master/ISSUES.md>
8. Scene description software/assets, Online, <https://mpeg.expert/software/MPEG/Systems/SceneDescription/software/assets>
9. ARCore API, Online, <https://developers.google.com/ar/reference>