Text

Description automatically generated ISO/IEC JTC 1/SC 29/WG 3 N00782

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

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

**Document type:** Output Document

**Title:** Procedures for test scenarios and reference software development for MPEG-I Scene Description

**Status:** Approved

**Date of document:** 2023-02-03

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

**Expected action:** ACT

**Action due date:** 2023-02-03

**No. of pages:** 9 (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** **00782**

**Online – January 2023**

|  |  |
| --- | --- |
| **Title** | **Procedures for test scenarios and reference software development for MPEG-I Scene Description** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **22306** |

Contents

[1 Scope 3](#_Toc126923184)

[2 Test scenarios 3](#_Toc126923185)

[2.1 Requirements 3](#_Toc126923186)

[2.2 Scenarios 4](#_Toc126923187)

[2.3 Template for test scenario 4](#_Toc126923188)

[2.4 Call for test data 4](#_Toc126923189)

[2.5 Timeline 5](#_Toc126923190)

[2.6 Available test assets 5](#_Toc126923191)

[2.7 Agreed test scenarios 5](#_Toc126923192)

[3 Contributions for Extensions 5](#_Toc126923193)

[3.1 General 5](#_Toc126923194)

[3.2 Extension Principles 7](#_Toc126923195)

[4 Software 7](#_Toc126923196)

[4.1. Reference software 7](#_Toc126923197)

[4.2. Conformance software 7](#_Toc126923198)

[5 Gitlab Management 7](#_Toc126923199)

[5.1 Git commit convention 7](#_Toc126923200)

[5.2. Branch convention 8](#_Toc126923201)

[5.2.1. Branch creation 8](#_Toc126923202)

[5.2.2. Branch update 8](#_Toc126923203)

[5.2.3. Tree model 8](#_Toc126923204)

[5.2 Scenarios 8](#_Toc126923205)

[5.3 Summary logistics 9](#_Toc126923206)

[5.4.Coordinators 9](#_Toc126923207)

# Scope

This document provides information and agreed processes to support the development of ISO/IEC 23090-14, "Scene Description for MPEG Media" as well as ISO/IEC 23090-24, "Conformance and Reference Software for MPEG-I Scene Description".

# Test scenarios

## Requirements

The work of the MPEG-I scene description is based on the requirements defined in N18965, later revised to N19511. The coverage of the requirements and the progress is documented in WG3\_N0294.

## Scenarios

An extension to MPEG-I Scene Description should be supported with a well-defined and agreed scenario(s). WG3\_N0294 also covers the mapping of requirements to scenarios.

The template for test scenarios can be found in Section 2.3.

The description of the agreed scenarios can be accessed at <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/test-assets#test-scenarios/>. The corresponding test assets as well as additional test assets can be accessed at

<http://mpegfs.int-evry.fr/mpegcontent/ws-mpegcontent/MPEG-I/Part14-SceneDescriptions>.

***Note****: access and contribution to this requires an account. To request an account, please contact the test asset coordinators (see clause 5.4)*

A new scenario must be proposed with an input contribution to an MPEG meeting with the following information as listed in clause 2.3.

## Template for test scenario

The following table should be used to propose test scenarios for scene description:

|  |  |
| --- | --- |
| **Item** | **Description** |
| **Title** | <give it a catchy title, e.g., as those listed in clause 2> |
| **Description** | * What is the basic use case? * How does it relate to MPEG-I Requirements and Use Cases? |
| **Required test assets** | * 3D scene, real-time assets for media (2D/3D) * Anything else * References to test assets |
| **Current Support** | * How can glTF Scene Description be used today * What are gaps/inefficiencies of glTF2.0 to address this scenario? |
| **Criteria** | * What are relevant criteria for the user experience/QoE? * What are relevant criteria for passing the test scenario? |

## Call for test data

Among others, we solicit the following material to be used as content for the creation and validation of MPEG-Scene Descriptions:

* 2D content that can server as overlays, video textures
* 2D and 3D content that is captured from a local camera, e.g., representing a conference room or flat surfaces for overlay
* 3D game content, e.g., provided in Unity, that can be used for the online gaming scenario
* 3D cinematographic content that includes complete scenes
* VR content and 3D mesh and point cloud content that can be used for VR scenes
* etc.

We welcome contributions of content that can be made available to the MPEG community for the development and progress of the MPEG-I Scene Description activity.

## Timeline

The data sets should be submitted as input contributions to an MPEG meeting but early submission into AHG is welcome.

## Available test assets

The following table lists the available assets and provides a brief description:

|  |  |
| --- | --- |
| **Asset** | **Description** |
| conferenceroom.zip | a glTF asset that represents a conference room. |
| livingroom.zip | a glTF asset that represents a living room. |
| island.zip | a glTF asset that represents an island. |
| chair.zip | a glTF asset that represents a chair. |
| bbb.mp4 | Big Buck Bunny video file in mp4 format. |
| longdress\_frame.ply | a binary PLY file from the longress point cloud sequence. |
| Scenario 11 | Test Assets:  1. Pine Forest  "author": "fangzhangmnm (https://sketchfab.com/fangzhangmnm)",  "license": "CC-BY-4.0 (http://creativecommons.org/licenses/by/4.0/)",  "source": "https://sketchfab.com/3d-models/pine-forest-ece69535f7584e099488f65f2072264e",  2. woodland-5\_trim\_SN3D.wav  Obtained and modified from EigenScape.  EigenScape is a database of acoustic scenes recorded spatially using the mh Acoustics EigenMike. https://doi.org/10.5281/zenodo.1012809  Marc Green <marc.c.green@york.ac.uk> |

***Note:*** *that the first four assets are downloaded from sketchfab and are available for download and usage under the Creative Commons license as describe in CC Attribution License:* [*https://creativecommons.org/licenses/by/4.0/*](https://creativecommons.org/licenses/by/4.0/)*.*

## Agreed test scenarios

Agreed test scenarios are provided here:

<http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/test-assets#test-scenarios/>

# Contributions for Extensions

## General

For every extension documented in ISO/IEC 23090-14 under the framework in clause 3 the following information is expected to be provided:

* The schema for the extension as part of the standard as well as a JSON document
* The semantics for the extension
* The processing model on the "Presentation Engine"
* The conformance description, i.e. conformance requirements for the Presentation Engine that supports the extension
* *A promise for example content that uses the extension that is finally available within one meeting after the technology was added. If not fulfilled, the feature is expected to be removed and this will be documented as a note in the draft standard.*
* *A promise of a reference implementation to the reference software as well as the conformance software as documented in clause 4, that is finally available within two meetings after the technology was added. If not fulfilled, the feature is expected to be removed and this will be documented as a note in the draft standard.*
* *The reference software implementation should implement the functionality to use the properties defined in the carriage format. Implementations in the reference software should support the mechanism and deliver an expected behaviour.*

**Note**: In an agile manner, this may be achieved using a JSON file which provides all the necessary data which would be included in a sample in ISOBMFF container. The reference software takes the JSON files as input and apply the properties accordingly as per the processing model. Later, the reference software implementation may incorporate and integrate a proper demuxer for the carriage formats.

* *To progress and integrate the support for carriage formats with the reference software, it is expected to bring supporting implementation in the carriage library.*

Hence, contributions addressing extensions to glTF under the framework in clause 3 should include the following:

* The scenarios that the extension is addressing. The scenarios are documented in clause 2.7.

In case, all the above-mentioned information is not available, a documented extension is not moved into the WD/CD but is maintained in the Technology under Consideration (TuC) document. The status of the completed information and the missing one is documented in the TUC.

The following text processes is recommended, but needs final verification:

*To fulfill the requirement on the reference software, it is sufficient to demonstrate that the reference software is able to properly process the test scenario. The test scenario content shall at least have a scene description file in glTF textual format that makes use of the proposed extension. The test scene description glTF document should use one of the available assets. The proposal must indicate any dependencies on other extensions.*

*The following is an example of this procedure:*

* *A test scenario is defined around support for video textures*
* *The proposal is to make use of the MPEG\_video\_texture extension*
* *A sample content is proposed based on the "conferenceroom" glTF file, which is part of the assets. The glTF file is extended to include the MPEG\_video\_texture extension. The bbb.mp4 asset is used to describe the video texture, which is attached to a rectangular mesh in the "conferenceroom" scene.*
* *The reference software is run with the modified scene description document and the expected behavior is demonstrated, showing the video texture.*

## Extension Principles

The following extension principles apply

* If the extension adds a new top-level array (by extending the root glTF object), its elements should inherit all properties of glTFChildOfRootProperty.schema.json.
* Other objects introduced by the extension should inherit all properties of glTFProperty.schema.json.
* By glTF 2.0 conventions, schemas should allow additional properties.
* Names **MUST** begin with an MPEG prefix, followed by an underscore.
* Names MUST use lowercase snake-case following the prefix, e.g., MPEG\_materials\_sand.
* Names **SHOULD** be structured as MPEG\_<scope>\_<feature>, where scope is an existing glTF concept (e.g. mesh, texture, image) and feature describes the functionality being added within that scope. This structure is recommended, but not required.
* Scope **SHOULD** be singular (e.g. mesh, texture), except where this would be inconsistent with an existing Khronos extension (e.g. materials, lights).

# Software

## 4.1. Reference software

The reference software for MPEG-I scene description is documented in WD of ISO/IEC 23090-24

## 4.2. Conformance software

The conformance software for MPEG-I scene description is documented in the WD of ISO/IEC 23090-24 as available in WG3 N0691.

# Gitlab Management

## Git commit convention

git commit -m "<optional WIP> <type>(#<issue id> <optional scope>): <description>"

The optional WIP information is to indicate your commit is in the “Work In Progress” state. Issue id is given in the [Issue panel](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/23090-24-gltf-validator/-/issues).

Type possible:

* **feat:** *The new feature you’re adding to a particular application*
* **fix:** *A bug fix*
* **hotfix:** *A bug fix to correct a major issue*
* **style:** *Feature and updates related to styling*
* **refacto:** *Refactoring a specific section of the codebase*
* **test:** *Everything related to testing*
* **doc:** *Everything related to documentation*
* **chore:** *Regular code maintenance [something which not fit with other previous types]*

## 5.2. Branch convention

### 5.2.1. Branch creation

git branch <type>/<initials>\_<why>

* **type:** *Everything which can work for a commit message*
* **initials:** *Initials of the owner*
* **why:** *The purpose of this branch written in PascalCase*

### 5.2.2. Branch update

When two developers are working on the same project, they will have their own working branch. If one merge his/her work to the develop branch, the second person should update his/her work to fit with the latest state of the develop branch. There are two possible ways to resolve such a situation:

* Rebasing develop branch to the working branch (recommended solution)
* git checkout <my\_branch\_name>

git rebase develop

* Merging develop branch to the working branch (recommended solution)
* git checkout <my\_branch\_name>

git merge develop

On the other hand, when a working branch is finished and needs to move into develop branch. A pull request needs to be generated directly on GitLab. Once completed in the platform, the working branch will be merged with the develop branch.

### 5.2.3. Tree model

Figure 1 illustrates an example of tree model for development.

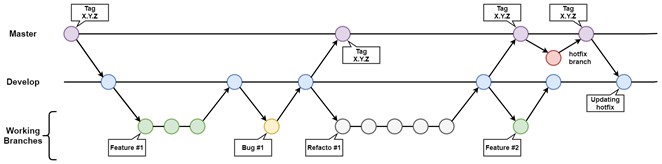


Figure 1. Tree model for branch

## Scenarios

To provide use cases that are to be supported by the standard, test scenarios are collected. Scenarios are described on what the basic setup an experience is expected to be and provides along with this test assets and test vectors (may be compressed or uncompressed) that may be used in the scenario. These test scenarios are collected at <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/test-assets#test-scenarios/>.

## Summary logistics

|  |  |  |
| --- | --- | --- |
| **Asset** | **Hosting** | **Location name** |
| Repository | MPEG Gitlab | http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription |
| Reference software | MPEG Gitlab | http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/reference |
| Conformance software | MPEG Gitlab | <http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/software/23090-24-gltf-validator> |
| Conformance software | MPEG Gitlab | https://gitlab.com/mpeg-i/scene-description/conformance |
| Scenarios | MPEG Gitlab | http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/test-assets |
| Test vectors | Gitlab.com with LFS for binary files | <https://gitlab.com/mpeg-i/scene-description/test-vectors> |
| Test assets | MPEG content | <http://mpegfs.int-evry.fr/mpegcontent/ws-mpegcontent/MPEG-I/Part14-SceneDescriptions> |

For uploading content to the Test Assets, please bring an input contribution to the MPEG meeting.

## 5.4.Coordinators

If you have created MPEG Git account but you cannot access the site, then please share the following information.

• Your name

• Your MPEG Git username

Please then send an email containing this information to the GitLab managers as listed in Table 1.

Table 1 Gitlab managers

|  |  |
| --- | --- |
| **Name** | **Email address** |
| Emmanuel Thomas | thomase@xiaomi.com |
| Imed Bouazizi | [bouazizi@qti.qualcomm.com](mailto:bouazizi@qti.qualcomm.com) |