

**ISO/IEC JTC 1/SC 29/WG 11**

**Coding of moving pictures and audio**

**Convenorship: UNI (Italy)**

**ISO/IEC JTC 1/SC 29/WG 11 N 19069**

**Document type: Approved WG 11 document**

**Title: Procedures for standard development and software of ISO/IEC**  **23090-13**

**Status: Approved**

**Date of document: 2020-01-16**

**Source: Systems**

**Expected action:**

**No. of pages: 7**

**Email of convenor: leonardo@chiariglione.org** <mailto:qianheng@sdis.cn>

**Committee URL: mpeg.chiariglione.org**

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG 129/N18xxx**

**Brussels, BE – January 2020**

|  |  |
| --- | --- |
| **Source** | **Systems** |
| **Title** | **Procedures for standard development and reference software of ISO/IEC 23090-14 (Scene Description)** |
| **Editor** | **Thomas Stockhammer** |

# Scope

This document provides information to support the development of ISO/IEC 23090-14. It also documents aspects for which feedback by MPEG#130 or in the adhoc phase is asked for.

# Time Plan

The work on MPEG-I scene description is part of MPEG-I phase 2a (add reference).

The following time plan is considered for the first version of the MPEG-I Scene Description standard:

* WD: 2020-01
* CD: 2020-07
* DIS: 2020-10
* FDIS: 2021-04
* Publication: 2021-07

The following meetings are scheduled

* Tuesday, Feb 18, 2020 15-17 cet, AHG Call
* Thursday, Mar 12, 2020 15-17 cet, AHG Call
* Sunday, Apr 19, 2020 14 -19, Alpbach Austria – AHG Meeting
* Several breakout sessions at MPEG#130 April 20 – 24, 2020

# Extending Khronos glTF2.0

Based on the agreement during MPEG#128, MPEG-I Scene Description is developed as an extension to Khronos' glTF2.0 specification. This specification can be accessed here: [https://gitlab.com/KhronosGroup/glTF/blob/master/specification/2.0/README.md](https://github.com/KhronosGroup/glTF/blob/master/specification/2.0/README.md)

According to the specification, glTF defines an extension mechanism that allows the base format to be extended with new capabilities. Any glTF object can have an optional extensions property. For details see <https://gitlab.com/KhronosGroup/glTF/blob/master/specification/2.0/README.md#specifying-extensions>. For more information on glTF extensions, consult the [extensions registry specification](https://github.com/KhronosGroup/glTF/blob/master/extensions/README.md).

glTF supports different ways on extending the specification as documented here: [https://gitlab.com/KhronosGroup/glTF/blob/master/extensions/README.md#promoting-extensions](https://github.com/KhronosGroup/glTF/blob/master/extensions/README.md#promoting-extensions)

The following principles are agreed:

* MPEG develops extensions to Khronos glTF2.0 under the *Vendor Extensions framework*. Contributing companies should be aware of this. If contributions do not provide a statement that says otherwise, it is expected that the proponents agree to this.
* MPEG requests here an extension with the prefix MPEG [https://gitlab.com/KhronosGroup/glTF/blob/master/extensions/Prefixes.md](https://github.com/KhronosGroup/glTF/blob/master/extensions/Prefixes.md). Contact person is the convenor, the systems chair as well as the chair of the BOG.
* If MPEG contributors are generally interested that their proposal may be considered as a KHR extension without any binding commitment, input contributions may state so. However, such a statement or the absence of such a statement does not impact the processing of a contribution in the context of the MPEG-I scene description work.

# Communication with Khronos

Khronos has active work in the context of glTF2.0, see the KHR extensions under development here: [https://gitlab.com/KhronosGroup/glTF/blob/master/extensions/README.md](https://github.com/KhronosGroup/glTF/blob/master/extensions/README.md). It is also identified that there is an overlap between MPEG members and glTF participants. Khronos and graphics experts meet in Khronos meetings, but also at developer and research conferences such as GDC and Siggraph. For meetings, please refer to <https://www.khronos.org/events/>.

Khronos Member Meetings occur 3 times per year and offer the opportunity for Khronos members to come together in a face-to-face environment to discuss technical work, industry feedback, network with colleagues and have some fun.

|  |  |  |
| --- | --- | --- |
| **Meeting** | **Date** | **Location** |
| F2F Barcelona 2020 | February 3-7, 2020 | Barcelona, Spain |
| F2F Phoenix 2020 | May 4-8, 2020 | Phoenix, Arizona |
| F2F Japan 2020 | September 21-25, 2020 | Osaka, Japan |

MPEG targets the following

* To request an extension with the prefix MPEG [https://gitlab.com/KhronosGroup/glTF/blob/master/extensions/Prefixes.md](https://github.com/KhronosGroup/glTF/blob/master/extensions/Prefixes.md) from MPEG#129 Contact person is the convenor, the systems chair as well as the chair of the BOG.
* to provide information to Khronos on the MPEG work for the F2F in May 2020 by sending an LS from MPEG#130 in April 2020. Qualcomm or other MPEG members offered to present the LS at the Khronos F2F and provide any additional verbal information to Khronos on the ongoing MPEG work.
* to potentially engage with Khronos experts in a joint workshop or conference at a convenient location for MPEG and Khronos during the development phase of the MPEG-I Scene Description work, preferably in late 2020 or early 2021.

# Requirements, Scenarios and Test Assets

The work of the MPEG-I scene description is based on the requirements defined in NXXXXX.

Providing Extension to MPEG-I Scene Description is based on well-defined and agreed scenarios. Scenarios include:

* Description of the scenario
* A set of test assets that are needed for the scenario

Agreed scenarios and test assets can be accessed:

* <https://gitlab.com/mpeg-i/scene-description/Scene-Description-SampleApplications>
* <https://gitlab.com/mpeg-i/scene-description/Scene-Description-TestAssets>

# Contributions for Extensions

For every extension documented in ISO/IEC 23090-14 under the framework in clause 6.3 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
* Example content that uses the extension
* A reference implementation in one of the agreed candidate reference software libraries as documented in clause 7.

Hence, contributions addressing extensions to glTF under the framework in clause 6.3 should include the following:

* The scenarios that this extension is addressing. The scenarios are documented here:
* All information from above

As long as not all the above information is available, a documented extension is not moved into the WD, but maintained in a TuC document. The status of the completed information and the missing one is documented in the TUC.

# Reference Software

For the success of MPEG-I Scene Description and extensions, reference software is considered necessary. It is also believed that there is a strong benefit in providing the reference software as public source, i.e. the software can be used outside the MPEG standardization process.

At MPEG#129, it was decided to initially go forward with two candidate reference software libraries:

* Trimesh:
  + External Reference: https://trimsh.org/index.html
  + MPEG-I SD Gitlab Repository:
    - <https://gitlab.com/mpeg-i/scene-description>/Scene-Description-Reference-LibTrimesh.
    - Further information is provided on the Readme on gitlab
  + License https://github.com/mikedh/trimesh/blob/master/LICENSE.md
* Diligent Graphics:
  + External Reference:
    - <https://diligentgraphics.com/diligent-engine/>
    - https://github.com/DiligentGraphics/DiligentEngine
  + MPEG-I SD Gitlab Repository:
    - <https://gitlab.com/mpeg-i/scene-description>/Scene-Description-Reference-LibDiligentGraphics.
    - Further information is provided on the Readme on gitlab
  + License <https://github.com/DiligentGraphics/DiligentEngine/blob/master/License.txt>

Moving forward, Contributors to this work as well as other MPEG members are encouraged to analyze the above two candidate reference software libraries and provide feedback on the following:

* If the technical features of the reference candidate software are sufficient for the purpose of their work in the context of MPEG-I scene description.
* If the complexity (code base, language, etc.) of the reference candidate software is adequate for the purpose of their work in the context of MPEG-I scene description.
* If they are able to contribute to the reference candidate software based on the software license without modifying the license.
* If their contributions can be submitted back to the original repository of the open source library or at least be made public.
* Other feedback on the usability of the software for their purposes in the context of MPEG standardization or in other activities.

Alternative licenses such as already compiled by MPEG (e.g. <https://gitlab.com/mpeg-i-visual/tmiv/blob/master/LICENSE>) may be attached to software contributions, but as we start from an existing library, it is preferred to not modify the license and continue to use the existing license of the initial software in order to simplify the usage of the developed reference software.

Based on feedback for MPEG#130, we may decide to reduce the number of candidate reference software libraries to one or continue with both libraries.

# Gitlab Management

## Reference implementation software

Candidate Reference Software Libraries are documented in clause 7 . Each of these software projects are be forked at the start time of the project and the development of the MPEG extensions will be done in the 'mpeg' branch. This would ease future import from and export to the original repositories if this needs to happen.

One Git repository per library will be created. The reason is repositories are free to create and separation of different software, build platform, documentation, etc. is desirable. The name of the repository will be Scene-Description-Reference-LibXXX.

## Conformance software

### glTF files

JSON glTF file are validated using JSON schema. For glTF binary files, it is proposed to define the binary structure in the [Kaitai](https://kaitai.io/) format (YAML based). This will allow the automatic generation of parsing libraries which can in turn be used to validate these binary files.

Both the JSON schemas and the Kaitai files, if needed, are proposed to be hosted on the same Git repositories called, Scene-Description-Conformance.

## Sample applications

To illustrate the use of the standard, several informative example applications can be developed during the standardization progress. It is proposed to collect these applications under the same Git repository with a folder at the root per library then a folder for each application inside each library. The name of this repository is Scene-Description-SampleApplications.

## Test assets

Test assets are not accessed on a frequent basis but usually requires protection by password to comply with the corresponding usage licenses, at least as it is commonly done in MPEG.

A folder in the MPEG content server will be created for the MPEG-I scene description standard. All the original "raw" test assets will be stored there along with the corresponding usage licenses.

## Test vectors

The test vectors are exercising the normative aspects of the specification. They will be stored in a single Git repository. When test vectors are binary, the LFS feature of the Git hosting service will be used in order to avoid polluting the Git tree with binary files. The name of the repository would be Scene-Description-TestVectors.

## Summary logistics

|  |  |  |
| --- | --- | --- |
| **Asset** | **Hosting** | **Location name** |
| Candidate Reference software | Gitlab.com | Scene-Description-Reference-LibXXX |
| Conformance software | Gitlab.com | Scene-Description-Conformance |
| Sample applications | Gitlab.com | Scene-Description-SampleApplications |
| Test vectors | Gitlab.com with LFS for binary files | Scene-Description-TestVectors |
| Test assets | MPEG content | Part-xx-SceneDescription |

The project is hosted at : <https://gitlab.com/mpeg-i/scene-description>/.

For access to the project, please register an account on GitLab.com at <https://gitlab.com/users/sign_in> and collect the following information:

* GitLab.com username
* GitLab.com email address

Please then send an email containing this information to the gitlab managers as listed in clause 9.

# Coordinators for Efforts until MPEG#130

* BOG Chair:
  + Thomas Stockhammer (tsto@qti.qualcomm.com)
* AHG Chairs:
  + Thomas Stockhammer (tsto@qti.qualcomm.com)
  + Mary-Luc Champel (champelmaryluc@xiaomi.com)
  + Ozgur Oyman ([ozgur.oyman@intel.com](mailto:ozgur.oyman@intel.com))
* Editor of ISO/IEC 23090-14
  + Imed Bouazizi (bouazizi@qti.qualcomm.com)
  + Ozgur Oyman (ozgur.oyman@intel.com)
* Editor of permanent document
  + Lukasz Kondrad ([lukasz.kondrad@nokia.com](mailto:lukasz.kondrad@nokia.com))
* Test Asset and Scenario Coordinator
  + Emmanuel Thomas (emmanuel.thomas@tno.nl)
  + Imed Bouazizi (bouazizi@qti.qualcomm.com)
* Gitlab Management
  + Emmanuel Thomas (emmanuel.thomas@tno.nl)
  + Imed Bouazizi (bouazizi@qti.qualcomm.com)