1. Text

   Description automatically generated ISO/IEC JTC 1/SC 29/WG 2 N268

**ISO/IEC JTC 1/SC 29/WG 2  
MPEG Technical requirements   
Convenorship: SFS (Finland)**

**Document type:** Output Document

**Title:** Requirements for Render-based systems for MPEG Phase 1 v1.1

**Status:** Approved

**Date of document:** 2023-01-20

**Source:** ISO/IEC JTC 1/SC 29/WG 2

**Expected action:** None

**Action due date:** None

**No. of pages:** 5 (with cover page)

**Email of Convenor:** igor.curcio@nokia.com

**Committee URL:** <https://sd.iso.org/documents/ui/#!/browse/iso/iso-iec-jtc-1/iso-iec-jtc-1-sc-29/iso-iec-jtc-1-sc-29-wg-2>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 2**

**MPEG TECHNICAL REQUIREMENTS**

**ISO/IEC JTC 1/SC 29/WG 2 N** **0268**

**Online - January 2023**

|  |  |
| --- | --- |
| **Title** | **Requirements for Render-based systems for MPEG Phase 1 V1.1** |
| **Source** | **WG 2, MPEG Technical requirements** |
| **Status** | **Approved** |
| **Serial Number** | **55067** |

**Introduction**

A set of current and emerging immersive displays are characterized by “families” according to the optical technologies used by each display to create each display’s optimal visual experience [1]. Moreover, [1] further explains that the same system to deliver immersive media to such emerging immersive displays may also be relevant to the delivery of media to emerging Metaverse applications. Both types of applications (displays and related to the Metaverse) are furthermore characterized as “render-based” applications.

This document provides requirements for MPEG to develop one or more specifications that collectively describe a “system” to deliver media to a render-based client as described in [1], where such media may be represented by one or more scene-based media formats. At the core of these specifications is the development of an Independent Mapping Space consisting of a vocabulary and corresponding metadata that can facilitate the interchange of a variety of scene-based media formats to satisfy the variety of input media formats required by a diversity of render-based applications.

An initial deliverable for the “system” is identified as MPEG-I Part 28 which aims to deliver such an Independent Mapping Space (IMS). Phase 1 toward the development of the complete system shall focus on the development of the IMS, especially with respect to the interchange of assets stored in the ITMF, USD, and glTF2.0 (without MPEG extensions) formats. Part 28 shall also prioritize the delivery of these formats to render-based clients that support the real-time rendering engine formats of both Unity and/or Unreal Engine.

All other aspects of the system including: compression of assets, streaming of scene-based media, and integration and support of MPEG-derived media formats (e.g., MIV, MPEG-I Part 14 Scene Description, MPEG-I Audio, V-PCC, G-PCC ) are deferred to one or more phases beyond Phase 1 [2].

ED Note: The requirements in this document are not final at this version of the document, and additional inputs are expected and solicited.

**Glossary**

***Scene graph***

general data structure commonly used by vector-based graphics editing applications and modern computer games, which arranges the logical and often (but not necessarily) spatial representation of a graphical scene; a collection of nodes and vertices in a graph structure.

***Node***

fundamental element of the scene graph comprised of information related to the logical or spatial or temporal representation of visual or audio information;

***Attribute***

metadata associated with a node used to describe a particular characteristic or feature of a node either in a canonical or more complex form (e.g. in terms of another node).

.

***Container***

a serialized format to store and exchange information to represent all natural, all synthetic, or a mixture of synthetic and natural scenes including a scene graph and all of the media resources that are required for rendering of the scene

[***Serialization***](https://en.wikipedia.org/wiki/Serialization)

the process of translating [data structures](https://en.wikipedia.org/wiki/Data_structure) or [object](https://en.wikipedia.org/wiki/Object_(computer_science)) state into a format that can be stored (for example, in a [file](https://en.wikipedia.org/wiki/Computer_file) or memory [buffer](https://en.wikipedia.org/wiki/Data_buffer)) or transmitted (for example, across a [network](https://en.wikipedia.org/wiki/Computer_network) connection link) and reconstructed later (possibly in a different computer environment). When the resulting series of bits is reread according to the serialization format, it can be used to create a semantically identical clone of the original object.

***[Renderer](https://en.wikipedia.org/wiki/Rendering_(computer_graphics))***

a software-based) application or process, based on a selective mixture of disciplines related to: acoustic physics, [light physics](https://en.wikipedia.org/wiki/Optics), [visual perception](https://en.wikipedia.org/wiki/Visual_system), audio perception, [mathematics](https://en.wikipedia.org/wiki/Mathematics), and [software development](https://en.wikipedia.org/wiki/Software_engineering), that, given an input scene graph and asset container, emits a visual and/or audio signal suitable for presentation on a targeted device or conforming to the desired properties as specified by attributes of a render target node in the scene graph. For visual-based media assets, a renderer may emit a visual signal suitable for a targeted display, or for storage as an intermediate asset (e.g. repackaged into another container i.e. used in a series of rendering processes in a graphics pipeline); for audio-based media assets, a renderer may emit an audio signal for presentation in a multi-channel loudspeaker and/or binauralized headphones, or for repackaging into another (output) container. Examples of renderers include: Octane, Redshift, Renderman, V-Ray, and real-time visual and audio renderers used in Unity, and Unreal.

***Scripting language***

An interpreted programming language that can be executed by a renderer at runtime to process dynamic input and variable state changes made to the scene graph nodes, which affect rendering and evaluation of spatial and temporal object topology (including physical forces, constraints, kinematics, deformation, collisions), and energy propagation and transport (light, sound).

***Shader***

a type of [computer program](https://en.wikipedia.org/wiki/Computer_program) that was originally used for [shading](https://en.wikipedia.org/wiki/Shading) (the production of appropriate levels of [light](https://en.wikipedia.org/wiki/Light), [darkness](https://en.wikipedia.org/wiki/Darkness), and [color](https://en.wikipedia.org/wiki/Color) within an image) but which now performs a variety of specialized functions in various fields of computer graphics [special effects](https://en.wikipedia.org/wiki/Special_effects) or does [video post-processing](https://en.wikipedia.org/wiki/Video_post-processing) unrelated to shading, or even functions unrelated to graphics at all.

***Ray tracing***

refers to a type of technique for 3D rendering that faithfully simulates real-world optical effects and is capable of producing photorealistic digital content.

***Rasterization***

refers to a type of technique for 3D rendering that maps 3D representations of scenes to a 2D output buffer, and is commonly used for most real-time applications (e.g., games). Most modern GPUs are highly optimized for rasterization in addition to ray tracing and deep learning.

***scene-based media***

audio, visual, haptic, and other primary types of media and media-related information organized logically and spatially by a use of a **scene graph**

Note to entry: **scene-based media** may be further organized into a linear series of scenes for time-based presentations, or into a branch-based structure for interactive presentations.

***frame-based media***

2D video with or without associated audio

***Node-graph***

Node structure (usually a node tree) that describes an asset, e.g., Alembic, glTF, FBX.

**Functions to be defined with respect to Part 28**

Part 28 shall perform the following essential functions:

* Defining an independent mapping space addressing multiple presentation engines, e.g., Unity and Unreal Engines.
* Defining a mapping between the independent mapping space and at least one presentation engine, e.g. Unreal Engine.

**MPEG requirements pertaining to Scene-based media interchange**

**General**

1. The specification shall define an independent mapping space (IMS) that addresses the need for interchange between scene graphs expressed by ITMF, USD, glTF 2.0 and multiple render engine interface formats including Unreal Engine and Unity Engine, e.g., using the most recent Long Term Support version of each engine.
2. The specification shall define a method to convey the IMS metadata into scene-graphs, e.g. USD, glTF 2.0,
3. The specification shall include an annex describing the mapping of IMS metadata into USD and glTF 2.0 formats.
4. The specification shall include an informative annex describing the mapping of IMS metadata into Unreal and Unity rendering engine interfaces.

**Independent Mapping Space**

1. The specification shall be define a vocabulary (list of labels) dense enough to support photorealistic scene-based media including but not limited to scenes capable of supporting view-dependent lighting, e.g., materials, textures, and geometry.
2. The specification shall be define a vocabulary (list of labels) dense enough to homogenize (harmonize) different scene graph representations.
3. The specification shall define a vocabulary for the structure and organization of information and media assets in a scene graph, e.g., nodes, immutable attributes, input and output parameters of each node.
4. The specification shall define a vocabulary to describe the semantics for connections that may exist between nodes and node graphs
5. The specification shall define a vocabulary to describe the semantics for nodes that may be organized into groups.
6. The specification shall define a vocabulary for the types of graphs, e.g., scene, application settings, geometry archive, scripted storage, and render job.
7. The specification shall define a vocabulary for rendering instructions and renderer interfaces, e.g., camera, environment, film and animation settings, post processing, and render layers.

**Reference Software**

1. Reference software shall be provided to implement the labelling process for one or more scene graph formats.

**References**

1. “Draft report on framework for characterizing emerging immersive displays and media for immersive applications,” ISO/IEC SC29/WG2 N00023, October 2020.
2. “Revised Draft requirements for render-based systems beyond MPEG Phase 1 v0.5,” MPEG 141, WG2 N00269, Online, January 2023.