INTERNATIONAL STANDARD

ISO/IEC 14496-5

Second edition
2001-12-15

AMENDMENT 40
2019-06

Information technology — Coding of audio-visual objects —

Part 5: Reference software

AMENDMENT 40: Printing material and 3D graphics coding for browsers reference software

Technologies de l'information — Codage des objets audiovisuels —
Partie 5: Logiciel de référence
AMENDEMENT 40: Matériel d'impression et codage graphique 3D pour le logiciel de référence des navigateurs

Reference number
ISO/IEC 14496-5:2001/Amd.40:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

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 document 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC 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) or the IEC list of patent declarations received (see http://patents.iec.ch).

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.

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 14496 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.

Information technology — Coding of audio-visual objects —

Part 5: Reference software

AMENDMENT 40: Printing material and 3D graphics coding for browsers reference software

Clause 7

Add two new subclauses (7.7 and 7.8):

7.7 Reference software for the IndexedPrintingRegionSet (IPRS)

7.7.1 General

This is the description of the reference software for IPRS which can assign an 3D printing material information per region to print the input 3D model whit proper printing materials. The reference software is available at https://standards.iso.org/iso-iec/14496/-5/ed-2/en/amd/40.

7.7.2 Description of classes

This subclause describes the new classes added for IPRS.

Class	Files	Folder structure	Description
IndexedPrintingRegionSet	indexedPrintingRegionSet.h	indexedPrintingRegionSet.h	Class containing IPRS data parsing function. With this class, the parsed IPRS data is stored in the PrintMatRegion structure.
	indexedPrintingRegionSet.cpp	indexedPrintingRegionSet.cpp	
CSlicer	Slicer.h	Slicer.h	Class containing Slicing functions. Slicing is done according to the plane and the intersection points and lines are stored and rendered by graphics rendering engine. In this reference code, the QT is used.
	Slicer.cpp	Slicer.cpp	
FileIO	FileIO.h	FileIO.h	Class containing file in-out function. This text reading function is used in the IndexedPrintingRegionSet class
	FileIO.cpp	FileIO.cpp	
main	—	—	This is not a class. This is the start of the code
	main.cpp	main.cpp	

7.8 Reference software for Web3DCoding

7.8.1 General

This is the description of the reference software for Web3DCoding which can be executed natively by web browsers. The reference software is available at https://standards.iso.org/iso-iec/14496/-5/ed-2/en/amd/40.

In ISO/IEC 14496-16:2011/Amd.3 the JSON schema that implements the scene description and how it connects to the SC3DMC-TFAN (Scalable Complexity 3D Mesh Coding – Triangle FAN) and BBA (Bone

Based Animation) is defined. The reference software is natively supported by web browsers as it is using the JSON schema based on the glTF format and the decoders is implemented in JavaScript.

7.8.2 Structure and description of files

This subclause describes the files structure added for Web3DCoding.

File name
gc/IndexedFaceSet.js
three.js
OrbitControls.js
gc/SC3DMCHeader.js
gc/ConnectivityBasedPredictor.js
gc/BinaryAlign.js
gc/SC3DMCDecoder.js
gc/ABone.js
gc/Animator.js
gc/ANMAtom.js
gc/matrix4.js
gc/CBBABone.js
gc/CMorph.js
gc/CMuscle.js
gc/bbaDecoder.js
gc/SC3DMCDecoder_main.js
gc/BaseConnectivityDecoder.js
gc/MultiVectOpt.js
gc/DecodeIntArray.js
gc/DecodeFloatArray.js
gc/TFANConnectivityDecoder.js
gc/TFANDecoder.js
gc/StorageOps.js
gc/InverseQuatization.js
gc/Adaptive_Data_Model.js
gc/Adaptive_Bit_Model.js
gc/Static_Bit_Model.js
gc/Static_Data_Model.js
gc/Arithmetic_codec.js
gc/InverseBinarizeIntArray.js
gc/InversePrediction.js

7.8.3 Instantiation description

This subclause describes the usage of Web3DCoding functions.

Renderer instantiation: THREE.WebGLRenderer {canvas:container, alpha:true, antialias: true});

Scene creation: THREE.Scene();

Camera instantiation: PerspectiveCamera (angle, width / height, 1, 100000);

Scene representation: JSON.parse(getJsonFile('json file'));

MPEG-SC3DMC decoder instantiation: gc_SC3DMCDecodeObject (filename);

MPEG-BBA bone hierarchy load: obj.object.shapes[i].anm[i].skeleton[i].numBones;

MPEG-BBA decoder instantiation:

getBinaryFile (objanimation_fileName);

THREE.SkinnedMesh (geometry, material);

THREE.Animation (skinnedMesh, animation);

ICS 35.040.40
Price based on 3 pages