BS ISO/IEC 14496-11:2015

BSI Standards Publication

Information technology — Coding of audio-visual objects

Part 11: Scene description and application engine

bsi.

...making excellence a habit.™

Document Sharing Hub

Full document details: BS ISO/IEC 14496-11:2015

National foreword

This British Standard is the UK implementation of ISO/IEC 14496-11:2015.

The UK participation in its preparation was entrusted to Technical Committee IST/37, Coding of picture, audio, multimedia and hypermedia information.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2015.
Published by BSI Standards Limited 2015

ISBN 978 0 580 82200 1

ICS 35.040

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2015.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 14496-11:2015

INTERNATIONAL STANDARD

ISO/IEC 14496-11

Second Edition
2015-11-01

Information technology — Coding of audio-visual objects —

Part 11: Scene description and application engine

Technologies de l'information — Codage des objets audiovisuels —

Partie 11: Description de scène et moteur d'application

Reference number
ISO/IEC 14496-11:2015(E)

Documents
Document Sharing Hub

© ISO/IEC 2015

Full document details: BS ISO/IEC 14496-11:2015

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2015

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 14496-11:2015

Contents

Page

Foreword v

0 Introduction vii
0.1 Scene Description vii
0.2 Extensible MPEG-4 Textual Format ix
0.3 MPEG-J ix

1 Scope 1

2 Normative references 1

3 Additional reference 2

4 Terms and definitions 2

5 Abbreviations and Symbols 7

6 Conventions 7

7 MPEG-4 Systems Node Semantics 8
7.1 Scene Description 8
7.2 Node Semantics 24
7.3 Informative: Differences Between MPEG-4 Scripts and ECMA Scripts 181
7.4 Informative: FlexTime behavior 182
7.5 Informative: Implementation of MaterialKey node 183
7.6 Informative: Example implementation of spatial audio processing (perceptual approach) 184
7.7 Informative: MPEG-4 Audio TTS application with Facial Animation 188
7.8 Informative: 3D Mesh Coding in BIFS scenes 188
7.9 Profiles 189
7.10 Metric information for resident fonts 220
7.11 Font metrics for SANS SERIF font (Albany) 221
7.12 Font metrics for SERIF font (Thorndale) 227
7.13 Font metrics for TYPEWRITER font (Cumberland) 234

8 BIFS 242
8.1 Introduction 242
8.2 Decoding tables, data structures and associated functions 242
8.3 Quantization 247
8.4 Compensation process 257
8.5 BIFS Configuration 258
8.6 BIFS Command Syntax 262
8.7 BIFS Scene 274
8.8 BIFS-Anim 305
8.9 Interpolator compression 310
8.10 Definition of bodySceneGraph nodes 349
8.11 Adaptive Arithmetic Decoder for BIFS-Anim 357
8.12 Informative : Adaptive Arithmetic Encoder for BIFS-Anim 359
8.13 View Dependent Object Scalability 360

9 The Extensible MPEG-4 Textual Format 381
9.1 Introduction 381
9.2 XMT-A Format 381
9.3 XMT-Ω Format 433
9.4 XMT-C Modules 478
9.5 XMT Schemas 486
9.6 Informative: XMT/X3D Compatibility 486
9.7 Informative: The usage of XMT-A BitWrapper element in authoring side 487

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: BS ISO/IEC 14496-11:2015

10 MPEG-J **500**
10.1 Architecture **500**
10.2 MPEG-J Session **502**
10.3 Delivery of MPEG-J Data **503**
10.4 MPEG-J API List **506**
10.5 Informative: Starting the Java Virtual Machine **512**
10.6 Informative: Examples of MPEG-J API usage **513**

Annex A (normative) **Curve-based animators** **522**

Annex B (normative) **Procedural textures algorithms** **525**

Annex C (informative) **Text Processing in BIFS** **530**

Annex D (informative) **Patent statements** **532**

Bibliography **533**

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 14496-11:2015

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 14496-11 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 29, *Coding of Audio, Picture, Multimedia and Hypermedia Information*.

This second edition cancels and replaces the first edition, which has been technically revised.

ISO/IEC 14496 consists of the following parts, under the general title *Information technology — Coding of audio-visual objects*:

— *Part 1: Systems*

— *Part 2: Visual*

— *Part 3: Audio*

— *Part 4: Conformance testing*

— *Part 5: Reference software*

— *Part 6: Delivery Multimedia Integration Framework (DMIF)*

— *Part 7: Optimized reference software for coding of audio-visual objects* [Technical Report]

— *Part 8: Carriage of ISO/IEC 14496 contents over IP networks*

— *Part 9: Reference hardware description* [Technical Report]

— *Part 10: Advanced Video Coding*

— *Part 11: Scene description and application engine*

— *Part 12: ISO base media file format*

— *Part 13: Intellectual Property Management and Protection (IPMP) extensions*

— *Part 14: MP4 file format*

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

— *Part 15: Advanced Video Coding (AVC) file format*

— *Part 16: Animation Framework eXtension (AFX)*

— *Part 17: Streaming text format*

— *Part 18: Font compression and streaming*

— *Part 19: Synthesized texture stream*

— *Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)*

— *Part 21: MPEG-J GFX*

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Introduction

1.1 Scene Description

1.1.1 Overview

ISO/IEC 14496 addresses the coding of audio-visual objects of various types: natural video and audio objects as well as textures, text, 2- and 3-dimensional graphics, and also synthetic music and sound effects. To reconstruct a multimedia scene at the terminal, it is hence not sufficient to transmit the raw audio-visual data to a receiving terminal. Additional information is needed in order to combine this audio-visual data at the terminal and construct and present to the end-user a meaningful multimedia scene. This information, called scene description, determines the placement of audio-visual objects in space and time and is transmitted together with the coded objects as illustrated in Figure 1. Note that the scene description only describes the structure of the scene. The action of assembling these objects in the same representation space is called composition. The action of transforming these audio-visual objects from a common representation space to a specific presentation device (i.e. speakers and a viewing window) is called rendering.

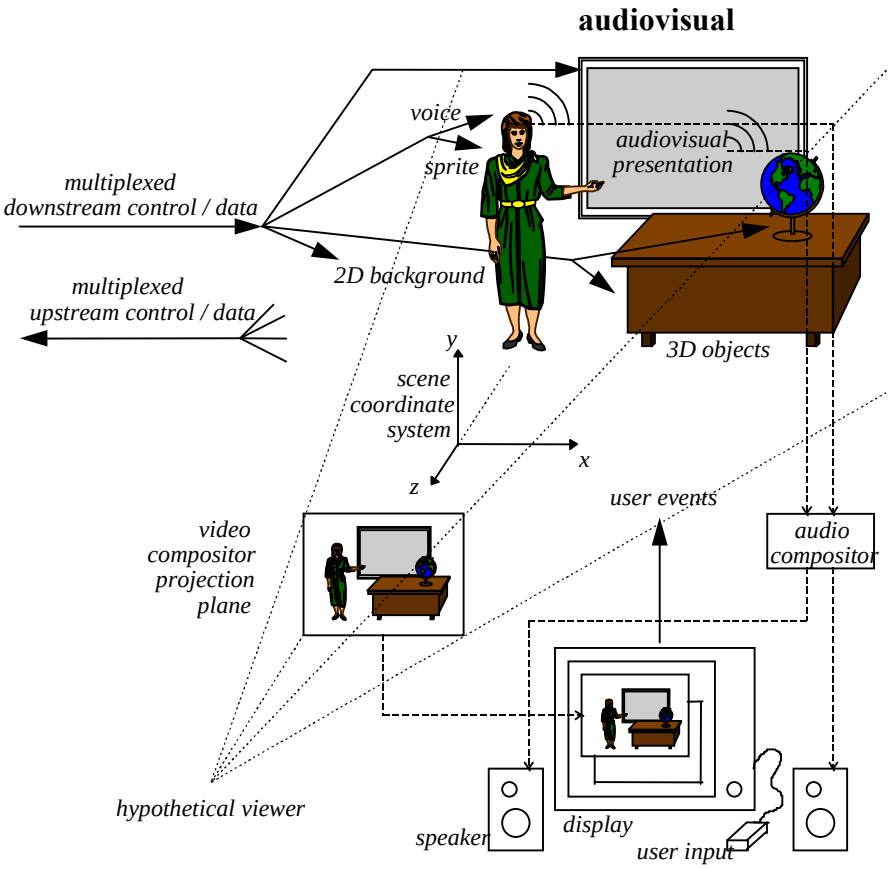

Figure 1 — An example of an object-based multimedia scene

Independent coding of different objects may achieve higher compression, and also brings the ability to manipulate content at the terminal. The behaviors of objects and their response to user inputs can thus also be represented in the scene description.

The scene description framework used in this part of ISO/IEC 14496 is based largely on ISO/IEC 14772-1:1998 (Virtual Reality Modeling Language – VRML).

1.1.2 Composition and Rendering

ISO/IEC 14496-11 defines the syntax and semantics of bitstreams that describe the spatio-temporal relationships of audio-visual objects. For visual data, particular composition algorithms are not mandated since they are implementation-dependent; for audio data, subclause 7.1.1.2.13 and the semantics of the AudioBIFS nodes normatively define the composition process. The manner in which the composed scene is presented to the user is not specified for audio or visual data. The scene description representation is termed “BInary Format for Scenes” (BIFS).

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

1.1.3 Scene Description

In order to facilitate the development of authoring, editing and interaction tools, scene descriptions are coded independently from the audio-visual media that form part of the scene. This permits modification of the scene without having to decode or process in any way the audio-visual media. The following clauses detail the scene description capabilities that are provided by ISO/IEC 14496-11.

1.1.3.1 Grouping of audio-visual objects

A scene description follows a hierarchical structure that can be represented as a graph. Nodes of the graph form audio-visual objects, as illustrated in Figure 2. The structure is not necessarily static; nodes may be added, deleted or be modified.

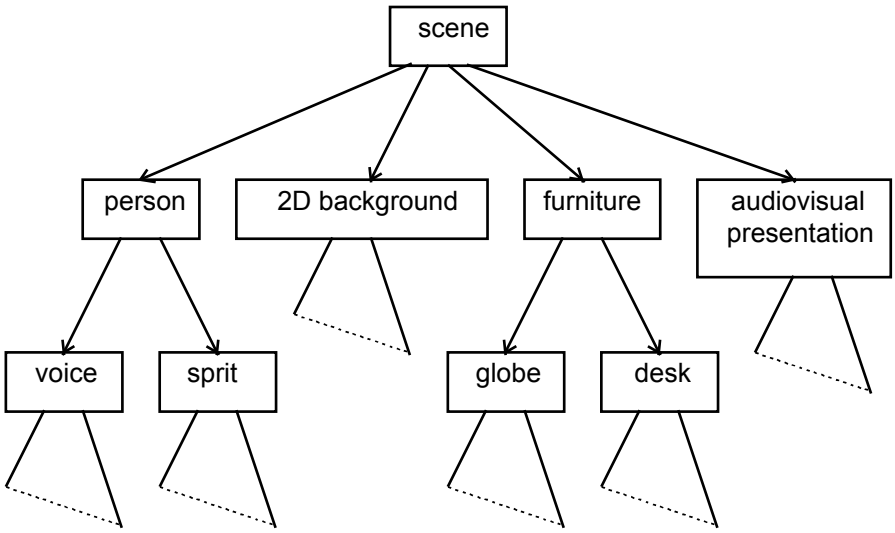

Figure 2 — Logical structure of example scene

1.1.3.2 Spatio-Temporal positioning of objects

Audio-visual objects have both a spatial and a temporal extent. Complex audio-visual objects are constructed by combining appropriate scene description nodes to build up the scene graph. Audio-visual objects may be located in 2D or 3D space. Each audio-visual object has a local co-ordinate system. A local co-ordinate system is one in which the audio-visual object has a pre-defined (but possibly varying) spatio-temporal location and scale (size and orientation). Audio-visual objects are positioned in a scene by specifying a co-ordinate transformation from the object's local co-ordinate system into another co-ordinate system defined by a parent node in the scene graph.

1.1.3.3 Attributes of audio-visual objects

Scene description nodes expose a set of parameters through which aspects of their appearance and behavior can be controlled.

EXAMPLE — the volume of a sound; the color of a synthetic visual object; the source of a streaming video.

1.1.3.4 Behavior of audio-visual objects

ISO/IEC 14496-11 provides tools for enabling dynamic scene behavior and user interaction with the presented content. User interaction can be separated into two major categories: client-side and server-side. Client-side interaction is an integral part of the scene description described herein. Server-side interaction is not dealt with.

Client-side interaction involves content manipulation that is handled locally at the end-user's terminal. It consists of the modification of attributes of scene objects according to specified user actions.

EXAMPLE — A user can click on a scene to start an animation or video sequence. The facilities for describing such interactive behavior are part of the scene description, thus ensuring the same behavior in all terminals conforming to ISO/IEC 14496-11.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

1.2 Extensible MPEG-4 Textual Format

1.2.1 Overview

The Extensible MPEG-4 Textual format (XMT) is a framework (illustrated in Figure 3) for representing MPEG-4 scene description using a textual syntax. The XMT allows the content authors to exchange their content with other authors, tools or service providers, and facilitates interoperability with both the Extensible 3D (X3D) being developed by the Web3D and the Synchronized Multimedia Integration Language (SMIL) from the W3C.

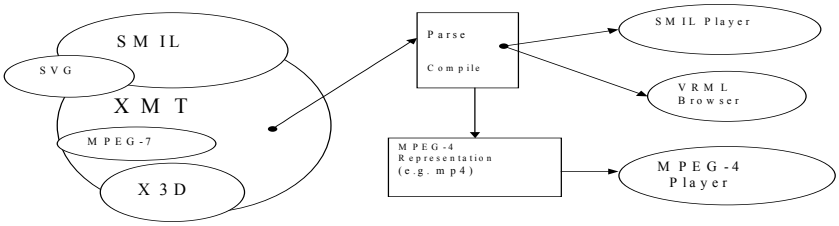

Figure 3 — Overview of the XMT Framework

1.2.2 Interoperability of XMT

The XMT format can be interchangeable between SMIL players, VRML players, and MPEG-4 players. The format can be parsed and played directly by a W3C SMIL player, preprocessed to Web3D X3D and played back by a VRML player, or compiled to an MPEG-4 representation such as MP4, which can then be played by an MPEG-4 player. See below for a graphical description of interoperability of the XMT.

1.2.3 Two-tier Architecture: XMT-A and XMT-Ω Formats

The XMT framework consists of two levels of textual syntax and semantics: the XMT-A format and the XMT-Ω format, which we will abbreviate by A and Ω, respectively, and use them interchangeably where there is no confusion.

The XMT-A is an XML-based version of MPEG-4 content, which contains a subset of the X3D. Also contained in XMT-A is an MPEG-4 extension to the X3D to represent MPEG-4 specific features. The XMT-A provides a straightforward, one-to-one mapping between the textual and binary formats.

The XMT-Ω is a high-level abstraction of MPEG-4 features designed based on the W3C SMIL. The XMT provides a default mapping from Ω to A, for there is no deterministic mapping between the two, and it also provides content authors with an escape mechanism from Ω to A.

In addition an XMT-C (Common) section contains the definition of elements and attributes that may be used within either XMT-A or XMT-Ω.

1.3 MPEG-J

1.3.1 Overview

MPEG-J is a flexible programmatic control system that represents an audio-visual session in a manner that allows the session to adapt to the operating characteristics when presented at the terminal. Two important characteristics are supported: first, the capability to allow graceful degradation under limited or time varying resources, and second, the ability to respond to user interaction and provide enhanced multimedia functionality.

More specifically, 9.7 normatively defines:

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

The format and delivery of Java byte code by specifying the MPEG-J stream format and the delivery mechanism of such a stream (Java byte code and associated data);

The MPEG-J Session and the MPEG-J application lifecycle; and

The interactions and behavior of byte code through the specification of Java APIs.

1.3.2 Organization MPEG-J specification

10.1 gives an overall architecture of the MPEG-J system. MPEG-J Session start-up is walked through in 10.2. The Delivery of MPEG-J data to the terminal is specified in 10.3. 10.4 specifies the different categories of APIs that a program in the form of Java bytecode would use. 10.5 is an informative annex on starting the Java Virtual Machine. The electronic annex attached to this document lists the normative MPEG-J APIs in the HTML format. 10.6 illustrates the usage of MPEG-J APIs through a few examples.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents.

The ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holder of these patent rights have assured the ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with the ISO and IEC. Information may be obtained from the companies listed in Annex D.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified in Annex D. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Information technology — Coding of audio-visual objects —

Part 11:
Scene description and application engine

1. Scope

This part of ISO/IEC 14496 specifies:

1. the coded representation of the spatio-temporal positioning of audio-visual objects as well as their behavior in response to interaction (scene description);
2. the Extensible MPEG-4 Textual (XMT) format, a textual representation of the multimedia content described in ISO/IEC 14496 using the Extensible Markup Language (XML); and
3. a system level description of an application engine (format, delivery, lifecycle, and behavior of downloadable Java byte code applications).

2. Normative references

The following referenced documents are indispensable for the application 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.

ISO 639-2:1998*, Codes for the representation of names of languages — Part 2: Alpha-3 code*

ISO 3166-1:1997, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 9613-1:1993, *Acoustics — Attenuation of sound during propagation outdoors — Part 1: Calculation of the absorption of sound by the atmosphere*

ISO/IEC 11172-2:1993, *Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 2: Video*

ISO/IEC 11172-3:1993, *Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 3: Audio*

ISO/IEC 13818-3:1998, *Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio*

ISO/IEC 13818-7: 2004, *Information technology — Generic coding of moving pictures and associated audio information — Part 7: Advanced Audio Coding (AAC)*

ISO/IEC 14496-2:2004, *Information technology — Coding of audio-visual objects — Part 2: Visual*

ISO/IEC 14772-1:1997, *Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding*

ISO/IEC 14772-1:1997/Amd.1:2003, *Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding — Amendment 1: Enhanced interoperability*

ISO/IEC 16262:2002, *Information technology — ECMAScript language specification*

ISO/IEC 13818-2:2000, *Information technology — Generic coding of moving pictures and associated audio information — Part 2: Video*

ISO/IEC 10918-1:1994, *Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines*

IEEE Std 754-1985, *Standard for Binary Floating-Point Arithmetic*

Addison-Wesley:September 1996, *The Java Language Specification,* by James Gosling, Bill Joy and Guy Steele, ISBN 0-201-63451-1

Addison-Wesley:September 1996, *The Java Virtual Machine Specification,* by T. Lindholm and F. Yellin, ISBN 0-201-63452-X

Addison-Wesley:July 1998, *Java Class Libraries Vol. 1 The Java Class Libraries,* Second Edition Volume 1, by Patrick Chan, Rosanna Lee and Douglas Kramer, ISBN 0-201-31002-3

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: BS ISO/IEC 14496-11:2015

Addison-Wesley:July 1998, *Java Class Libraries Vol. 2 The Java Class Libraries,* Second Edition Volume 2, by Patrick Chan and Rosanna Lee, ISBN 0-201-31003-1

Addison-Wesley, May 1996, *Java API, The Java Application Programming Interface, Volume 1: Core Packages,* by J. Gosling, F. Yellin and the Java Team, ISBN 0-201-63453-8

DAVIC 1.4.1 specification Part 9: Information Representation

ANSI/SMPTE 291M-1996, *Television — Ancillary Data Packet and Space Formatting*

SMPTE 315M -1999, *Television — Camera Positioning Information Conveyed by Ancillary Data Packets*

3. Additional reference

ISO/IEC 13522-6:1998, *Information technology — Coding of multimedia and hypermedia information — Part 6: Support for enhanced interactive applications.* This reference contains the full normative references to Java APIs and the Java Virtual Machine as described in the normative references above.

4. Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
Access Unit (AU)

individually accessible portion of data within an *elementary stream*

NOTE An access unit is the smallest data entity to which timing information can be attributed.

4.2
Alpha map

representation of the transparency parameters associated with a texture map

4.3
audio-visual object

representation of a natural or synthetic object that has an audio and/or visual manifestation

NOTE The representation corresponds to a node or a group of nodes in the BIFS scene description. Each audio-visual object is associated with zero or more *elementary streams* using one or more *object descriptors.*

4.4
audio-visual scene (AV scene)

set of audio-visual objects together with scene description information that defines their spatial and temporal attributes including behaviors resulting from object and user interactions

4.5
Binary Format for Scene (BIFS)

coded representation of a parametric scene description format

4.6
buffer model

model that defines how a terminal complying with ISO/IEC 14496 manages the buffer resources that are needed to decode a presentation

4.7
byte aligned

position in a coded bit stream with a distance of a multiple of 8-bits from the first bit in the stream

4.8
clock reference

special time stamp that conveys a reading of a time base

4.9
composition

process of applying scene description information in order to identify the spatio-temporal attributes and hierarchies of audio-visual objects

4.10
Composition Memory (CM)

random access memory that contains composition units

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 14496-11:2015

4.11
Composition Time Stamp (CTS)
indication of the nominal composition time of a composition unit

4.12
Composition Unit (CU)
individually accessible portion of the output that a decoder produces from access units

4.13
compression layer
layer of a system according to the specifications in ISO/IEC 14496 that translates between the coded representation of an elementary stream and its decoded representation

NOTE It incorporates the decoders.

4.14
decoder
entity that translates between the coded representation of an elementary stream and its decoded representation

4.15
decoding buffer (DB)
buffer at the input of a decoder that contains access units

4.16
decoder configuration
configuration of a decoder for processing its elementary stream data by using information contained in its elementary stream descriptor

4.17
Decoding Time Stamp (DTS)
indication of the nominal decoding time of an access unit

4.18
delivery layer
generic abstraction for delivery mechanisms (computer networks, etc.) able to store or transmit a number of multiplexed elementary streams or M4Mux streams

4.19
descriptor
data structure that is used to describe particular aspects of an elementary stream or a coded audio-visual object

4.20
Elementary Stream (ES)
consecutive flow of mono-media data from a single source entity to a single destination entity on the compression layer

4.21
Elementary Stream Descriptor (ESD)
structure contained in object descriptors that describes the encoding format, initialization information, sync layer configuration, and other descriptive information about the content carried in an elementary stream

4.22
M4Mux Channel (FMC)
label to differentiate between data belonging to different constituent streams within one M4Mux Stream

NOTE A sequence of data in one M4Mux channel within a M4Mux stream corresponds to one single SL-packetized stream.

4.23
M4Mux packet
smallest data entity managed by the M4Mux tool, consisting of a header and a payload

4.24
M4Mux stream
sequence of M4Mux Packets with data from one or more SL-packetized streams that are each identified by their own M4Mux channel

4.25
M4Mux tool
tool that allows the interleaving of data from multiple data streams

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

4.26
graphics profile

profile that specifies the permissible set of graphical elements of the BIFS tool that may be used in a scene description stream

NOTE Note BIFS comprises both graphical and scene description elements.

4.27
inter

mode for coding parameters that uses previously coded parameters to construct a prediction

4.28
interaction stream

elementary stream that conveys user interaction information

4.29
intra

mode for coding parameters that does not make reference to previously coded parameters to perform the encoding

4.30
initial object descriptor

special object descriptor that allows the receiving terminal to gain initial access to portions of content encoded according to ISO/IEC 14496

NOTE It conveys profile and level information to describe the complexity of the content.

4.31
Intellectual Property Identification (IPI)

unique identification of one or more elementary streams corresponding to parts of one or more audio-visual objects

4.32
Intellectual Property Management and Protection System (IPMP)

generic term for mechanisms and tools to manage and protect intellectual property

NOTE Only the interface to such systems is normatively defined.

4.33
media node

following list of time dependent nodes that refers to a media stream through a URL field: **AnimationStream**, **AudioBuffer, AudioClip**, **AudioSource**, **Inline**, **MovieTexture**

4.34
media stream

one or more elementary streams whose ES descriptors are aggregated in one object descriptor and that are jointly decoded to form a representation of an AV object

4.35
media time line

time line expressing normal play back time of a media stream

4.36
Object Clock Reference (OCR)

clock reference that is used by a decoder to recover the time base of the encoder of an elementary stream

4.37
Object Content Information (OCI)

additional information about content conveyed through one or more elementary streams

NOTE It is either aggregated to individual elementary stream descriptors or is itself conveyed as an elementary stream.

4.38
Object Descriptor (OD)

descriptor that aggregates one or more elementary streams by means of their elementary stream descriptors and defines their logical dependencies

4.39
Object Descriptor Command

command that identifies the action to be taken on a list of object descriptors or object descriptor IDs, e.g. update or remove

4.40
Object Descriptor Profile

profile that specifies the configurations of the object descriptor tool and the sync layer tool that are allowed

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

4.41
Object Descriptor Stream
elementary stream that conveys object descriptors encapsulated in object descriptor commands

4.42
Object Time Base (OTB)
time base valid for a given elementary stream, and hence for its decoder

NOTE The OTB is conveyed to the decoder via object clock references. All time stamps relating to this object's decoding process refer to this time base.

4.43
Parametric Audio Decoder
set of tools for representing and decoding speech signals coded at bit rates between 6 Kbps and 16 Kbps, according to the specifications in ISO/IEC 14496-3

4.44
Quality of Service (QoS)
performance that an elementary stream requests from the delivery channel through which it is transported

NOTE QoS is characterized by a set of parameters (e.g. bit rate, delay jitter, bit error rate, etc.).

4.45
random access
process of beginning to read and decode a coded representation at an arbitrary point within the elementary stream

4.46
reference point
location in the data or control flow of a system that has some defined characteristics

4.47
rendering
action of transforming a scene description and its constituent audio-visual objects from a common representation space to a specific presentation device (i.e., speakers and a viewing window)

4.48
rendering area
portion of the display device's screen into which the scene description and its constituent audio-visual objects are to be rendered

4.49
Symbolic Music Representation (SMR)

A method of describing a logical structure consisting of: symbolic elements that represent audiovisual events; the relationship between those events; and aspects of rendering those events as defined by ISO/IEC 14496-23.

4.50
scene description
information that describes the spatio-temporal positioning of audio-visual objects as well as their behavior resulting from object and user interactions

NOTE The scene description makes reference to elementary streams with audio-visual data by means of pointers to object descriptors.

4.51
scene description stream
elementary stream that conveys scene description information

4.52
scene graph elements
elements of the BIFS tool that relate only to the structure of the audio-visual scene (spatio-temporal positioning of audio-visual objects as well as their behavior resulting from object and user interactions) excluding the audio, visual and graphics nodes as specified in 14496-11

4.53
scene graph profile
profile that defines the permissible set of scene graph elements of the BIFS tool that may be used in a scene description stream

NOTE Note BIFS comprises both graphical and scene description elements.

4.54
seekable
media stream is seekable if it is possible to play back the stream from any position

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

4.55
SL-Packetized Stream (SPS)
sequence of sync layer packets that encapsulate one elementary stream

4.56
stream object
media stream or a segment thereof

NOTE A stream object is referenced through a URL field in the scene in the form “OD:n” or “OD:n#<segmentName>”.

4.57
structured audio
method of describing synthetic sound effects and music as defined by ISO/IEC 14496-3

4.58
Sync Layer (SL)
layer to adapt elementary stream data for communication across the DMIF Application Interface, providing timing and synchronization information, as well as fragmentation and random access information

NOTE The sync layer syntax is configurable and can be configured to be empty.

4.59
Sync Layer Configuration
configuration of the sync layer syntax for a particular elementary stream using information contained in its elementary stream descriptor

4.60
Sync Layer Packet (SL-Packet)
smallest data entity managed by the sync layer consisting of a configurable header and a payload

NOTE The payload may consist of one complete access unit or a partial access unit.

4.61
Syntactic Description Language (SDL)
language defined by ISO/IEC 14496-1 that allows the description of a bitstream’s syntax

4.62
Systems Decoder Model (SDM)
model that provides an abstract view of the behavior of a terminal compliant to ISO/IEC 14496

NOTE It consists of the buffer model and the timing model.

4.63
System Time Base (STB)
time base of the terminal

NOTE Its resolution is implementation-dependent. All operations in the terminal are performed according to this time base.

4.64
terminal
system that sends, or receives and presents the coded representation of an interactive audio-visual scene as defined by ISO/IEC 14496-11

NOTE It can be a stand alone system, or part of an application system complying with ISO/IEC 14496.

4.65
time base
notion of a clock, equivalent to a counter that is periodically incremented

4.66
timing model
model that specifies the semantics of timing information, how it is incorporated (explicitly or implicitly) in the coded representation of information, and how it can be recovered at the receiving terminal

4.67
time stamp
indication of a particular time instant relative to a time base

4.68
interaction stream
elementary stream that conveys user interaction information

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

5. Abbreviations and Symbols

AFX	Animation Framework eXtension
AU	Access Unit
AV	Audio-visual
BIFS	Binary Format for Scene
CM	Composition Memory
CTS	Composition Time Stamp
CU	Composition Unit
DAI	DMIF Application Interface (see ISO/IEC 14496-6)
DB	Decoding Buffer
DTS	Decoding Time Stamp
ES	Elementary Stream
ESI	Elementary Stream Interface
ESID	Elementary Stream Identifier
FAP	Facial Animation Parameters
FAPU	FAP Units
FDP	Facial Definition Parameters
FIG	FAP Interpolation Graph
FIT	FAP Interpolation Table
FMC	M4Mux Channel
FMOD	The floating point modulo (remainder) operator which returns the remainder of x/y such that: Fmod(x/y) = x – k*y, where k is an integer, sgn(fmod(x/y)) = sgn(x), and abs(fmod(x/y)) < abs(y)
IP	Intellectual Property
IPI	Intellectual Property Identification
IPMP	Intellectual Property Management and Protection
NCT	Node Coding Tables
NDT	Node Data Type
NINT	Nearest INTeger value
OCI	Object Content Information
OCR	Object Clock Reference
OD	Object Descriptor
ODID	Object Descriptor Identifier
OTB	Object Time Base
PLL	Phase Locked Loop
QoS	Quality of Service
SAOL	Structured Audio Orchestra Language
SASL	Structured Audio Score Language
SDL	Syntactic Description Language
SDM	Systems Decoder Model
SL	Synchronization Layer
SL-Packet	Synchronization Layer Packet
SPS	SL-Packetized Stream
STB	System Time Base
TTS	Text-To-Speech
URL	Universal Resource Locator
VOP	Video Object Plane
VRML	Virtual Reality Modeling Language

6. Conventions

For the purpose of unambiguously defining the syntax of the various bitstream components defined by the normative parts of ISO/IEC 14496 a *syntactic description language* is used. This language allows the specification of the mapping of the various parameters in a binary format as well as how they are placed in a serialized bitstream. The definition of the language is provided in 14496-1.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

7. MPEG-4 Systems Node Semantics

7.1 Scene Description

7.1.1 Concepts

7.1.1.1 BIFS Elementary Streams

7.1.1.1.1 Overview

BIFS is a compact binary format representing a pre-defined set of audio-visual objects, their behaviors, and their spatio-temporal relationships. The BIFS scene description may, in general, be time-varying. Consequently, BIFS data is carried in a dedicated elementary stream and is subject to the provisions of the systems decoder model (see 8.2). Portions of BIFS data that become valid at a given point in time are contained in BIFS `CommandFrames` or `AnimationFrames` and are delivered within time-stamped access units. Note that the initial BIFS scene is sent as a BIFS-Command, although it is not required, in general, that a BIFS `CommandFrame` contains a complete BIFS scene description.

7.1.1.1.2 BIFS Decoder Configuration

BIFS configuration information is contained in a `BIFSConfig` (see 8.5.2) syntax structure, which is transmitted as `DecoderSpecificInfo` for the BIFS elementary stream in the corresponding object descriptor (see 7.2.6.7, ISO/IEC 14496-1). This gives basic information that must be known by the terminal in order to parse the BIFS elementary stream. In particular, it indicates whether the stream consists of BIFS-Command or BIFS-Anim entities.

7.1.1.1.3 BIFS Access Units

A BIFS data access unit consists of one BIFS `CommandFrame` or `AnimationFrame`, as defined in 8.6.2 and 8.8.2 respectively. The BIFS `CommandFrame` or `AnimationFrame` shall convey all the data that is to be processed at any given instant in time. Access units in BIFS streams shall be labelled and time-stamped by suitable means. This shall be done via the related flags and the composition time stamps (CTS), respectively, in the SL packet header (see 7.3.2.4, ISO/IEC 14496-1). The composition time indicates the point in time at which the `CommandFrame` or `AnimationFrame` embedded in a BIFS access unit shall become valid. This means that any changes to audio-visual objects that are described in the BIFS access unit will become visible or audible at precisely this time in an ideal compositor, unless a different behavior is specified by the fields of their nodes. Decoding and composition time for a BIFS access unit shall always have the same value.

An access unit does not necessarily convey a complete scene. In that case it just modifies the persistent state of the scene description. However, if an access unit conveys a complete scene as required at a given point in time it shall set the `randomAccessPointFlag` in the SL packet header to '1' for this access unit. Otherwise, the `randomAccessPointFlag` shall be set to '0'.

7.1.1.1.4 Time base for BIFS streams

The time base associated with a BIFS stream shall be indicated by suitable means. This shall be done by means of object clock reference time stamps in the SL packet headers (see 7.3.2.4, ISO/IEC 14496-1) for this stream or by indicating the elementary stream from which this BIFS stream inherits the time base (see 7.3.2.3, ISO/IEC 14496-1). All time stamps in the SL-packetized BIFS stream refer to this time base.

7.1.1.1.5 Multiple BIFS streams

Scene description data may be conveyed in more than one BIFS elementary streams. Two distinct mechanisms exist to associate a set of BIFS elementary streams to a single scene.

The first method uses **Inline** nodes (see 7.2.2.72) in a BIFS scene description. Each such node refers to further BIFS elementary streams. In this case, multiple BIFS streams have a hierarchical dependency. Each **Inline** node opens a new name scope for the identifiers used to label BIFS elements (`nodeID`, `ROUTEID`, `objectDescriptorID`). Therefore, it is not possible to pass events between parts of a scene that reside below different **Inline** nodes.

EXAMPLE 1 — An application of hierarchical BIFS streams is a multi-user virtual conferencing scene, where sub-scenes originate from different sources. Usually, it is neither possible nor useful to specify interaction between two such disjoint parts of the scene.

The second method to associate multiple BIFS elementary streams to a single scene is to group their elementary stream descriptors in a single object descriptor (see 7.2.7.2.2, ISO/IEC 14496-1). In this case, these BIFS streams share the same scope for the identifiers they use (`nodeID`, `ROUTEID`, `objectDescriptorID`). This allows a single scene to be partitioned into multiple streams.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

EXAMPLE 2 — An application may offer a presentation with different levels of detail, corresponding to different data rates and different computational complexity. By sharing the same name scope, the more detailed scene description can build on the simple one, rather than sending the entire scene again.

7.1.1.1.6 Time

7.1.1.1.6.1 Stream Objects

A Media Stream consists of one or more elementary streams whose ES descriptors are aggregated in one object descriptor and that are jointly decoded to form a representation of an AV object. Such streams may be streamed in response to player requests, in particular in the case of Media nodes that control play back of media. Streams may be seekable, in which case the stream can be played from any (randomly accessible) time position in the stream, or they may be non-seekable, in which case the player has no control over the playback of the stream, as is the case in broadcast scenarios.

7.1.1.1.6.2 Time-dependent Media Nodes

This specification defines the notion of a Media Node. Such nodes control the opening and playback of remote streams and are time-dependent nodes.

The **url** field of a media node shall contain at most one element which must point to a complete media stream, i.e. it is of the form “OD:n”. Media Nodes may become active or inactive based on the value of their **startTime** and **stopTime** fields. The mediaTime of the played stream is controlled by a MediaControl node, and is not dependent on the **startTime** and **stopTime** in the media Node.

The semantics of the **loop**, **startTime** and **stopTime** exposedFields and the **isActive** eventOut in time-dependent nodes are as described in ISO/IEC 14772-1:1998, Subclause 4.6.9. **startTime**, **stopTime** and **loop** apply only to the local start, pause and restart of these nodes. In the case of media Nodes, these fields affect the delivery of the stream attached to media nodes as described below. The following media nodes exist: **AnimationStream**, **AudioBuffer, AudioClip**, **AudioSource**, **MovieTexture.**

When a media node becomes active and the stream associated with that media node is already active, the media node simply joins the session. If the stream is not active when the media node becomes active, the stream becomes active; i.e. it is played.

When a media node becomes inactive, the stream shall become inactive if there are no other active media nodes referring to that stream, otherwise the stream remains active.

For media nodes that reference streams that are synchronized with the scene, reaching **startTime** and **stopTime** shall not cause the streams to become active or inactive. Instead, the media shall be muted (i.e. hidden) and unmuted, respectively.

Loop and **speed** in a **MediaControl** node shall over-ride the same fields, when they exist, in any media node referencing the controlled stream. These fields retain their semantics when no controlling **MediaControl** node is present in the scene.

7.1.1.1.6.3 Time fields in BIFS nodes

SFTime field in BIFS nodes define either a time duration or a point in time.

Durations expressed by relative SFTime values like the **cycleTime** field of the **TimeSensor** node are determined unambiguously by the time base of the BIFS stream as defined in 7.1.1.1.4. Note that the time base of a stream (and hence the scene time of a scene sub-tree) can be modified by the **TemporalTransform** node which is used to synchronize different streams.

The **startTime** and **stopTime** SFTime fields in BIFS Media nodes and the **TimeSensor** node represent an absolute position on the time line of the BIFS stream. This absolute position is defined as follows:

Each node in the scene description has an associated point in time at which it is inserted in the scene graph or at which an absolute-position SFTime field in such a node is updated through a `CommandFrame` in a BIFS access unit (see 7.1.1.1.3). The value of such an SFTime field in the scene graph shall be calculated as the sum of the value encoded in the BIFS command (or the default value, if no value is encoded in the BIFS command) and the scene time as evaluated in the scene graph (e.g. the value of the **time** field of a **TimeSensor**) when the command is composed.

Offset to **startTime** / **stopTime** fields shall be applied in the following cases:

- insertion of the node with explicit or default value for the **startTime** / **stopTime** fields;
- replacement of the value of the **startTime** / **stopTime** field through a BIFS Command ;
- replacement of the PROTO interface field to which the **startTime** / **stopTime** field is routed through an IS statement;

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

- insertion of a PROTO instance whose body contains **startTime** / **stopTime** fields.

No offset shall be applied to a startTime / **stopTime** field in the following cases:

- modification through a ROUTE;
- replacement of the value of the **startTime** / **stopTime** field through a Script or MPEG-J.

NOTE 1 — Absolute time in ISO/IEC 14772-1:1998 is defined slightly differently. Due to the non-streamed nature of the scene description in that case, absolute time corresponds to wallclock time in ISO/IEC 14772-1.

EXAMPLE — The example in Figure 4 shows a BIFS access unit that is to become valid at CTS. It conveys a node that has an associated media elementary stream. The startTime of this node is set to a positive value Δt. Hence, startTime will occur Δt seconds after the CTS of the BIFS access unit that has incorporated this node (or the value of the startTime field) in the scene graph.

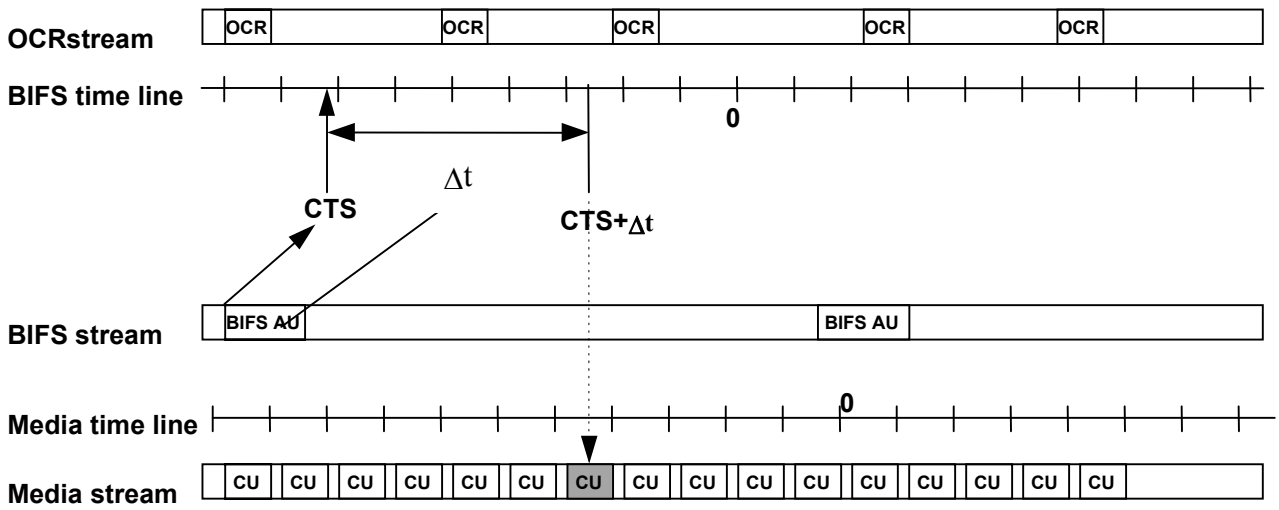

Figure 4 — Media start times and CTS

7.1.1.1.7 Sensor Reaction to Scene Changes

When the whichChoice field of a Switch node is changed or when a BIFS-command is executed, the current pointing device position shall be reevaluated with respect to all sensors in the scene.

7.1.1.2 BIFS Scene Graph

7.1.1.2.1 Structure of the BIFS scene graph

Conceptually, BIFS scenes represent (as in ISO/IEC 14772-1:1998) a set of visual and audio primitives distributed in a directed acyclic graph, in a 3D space. However, BIFS scenes may fall into several sub-categories representing particular cases of this conceptual model. In particular, BIFS scene descriptions support scenes composed of:

2D primitives (only)

3D primitives (only)

A combination of 2D and 3D primitives

Audio primitives (only)

In scenes combining 2D and 3D primitives, the following possibilities exist:

Complete 2D and 3D scenes layered in a 2D space with depth

2D and 3D scenes used as texture maps for 2D or 3D primitives

2D scenes drawn in the local X-Y plane of the local co-ordinate system in a 3D scene

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Figure 5 describes a typical BIFS scene structure.

A BIFS scene shall start with a one of the following nodes: **OrderedGroup**, **Group**, **Layer2D**, **Layer3D**. When the profile used enables visual elements to be composed, the first node indicates the co-ordinate system and context (2D or 3D) to be used for the children of that node. The following rules apply:

- Scene starts with a **Layer2D** or **OrderedGroup** node: A 2D co-ordinate system and context is assumed.
- Scene starts with a **Layer3D** or **Group** node : A 3D co-ordinate system and context is assumed.

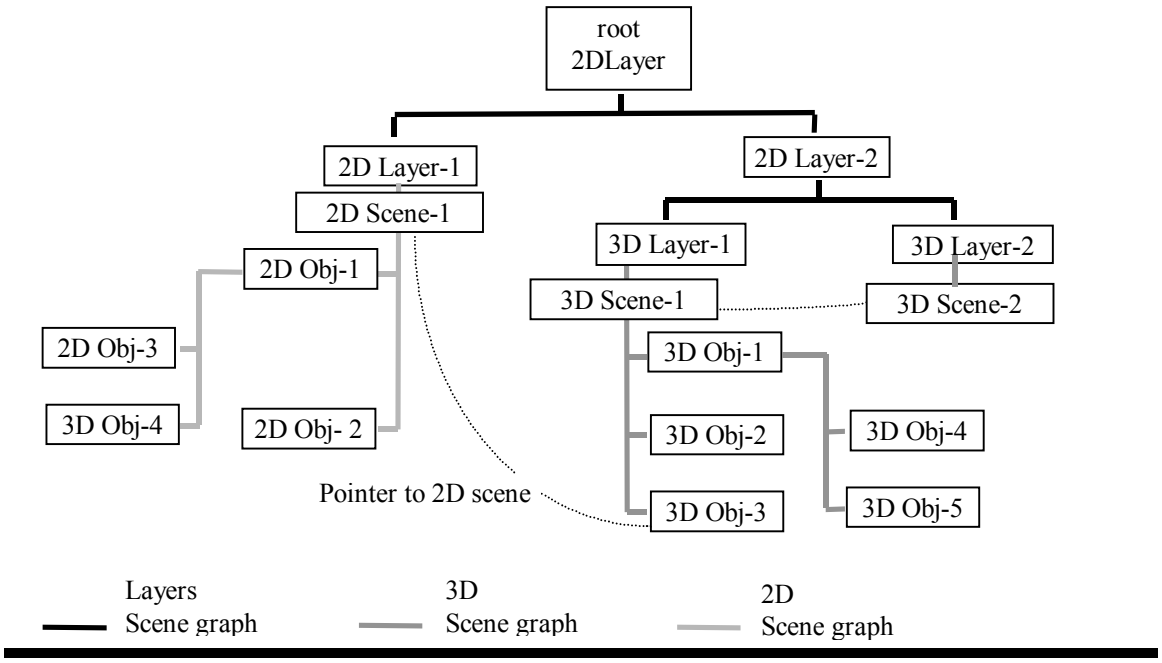

Figure 5 — Scene graph example.

The hierarchy of three different scene graphs is shown: a 2D graphics scene graph and two 3D graphics scene graphs combined with the 2D scene via layer nodes. As shown in the picture, the 3D Layer-2 is the same scene as 3D Layer-1, but the viewpoint may be different. The 3D Obj-3 is an Appearance node that uses the 2D Scene-1 as a texture node.

7.1.1.2.2 2D Co-ordinate System

The origin of the 2D co-ordinate system is positioned in the center of the rendering area, the x-axis is positive to the right, and the y-axis is positive upwards.

The width of the rendering area represents -1.0 to +1.0 (meters) on the x-axis (see Figure 6). The extent of the y-axis in the positive and negative directions is determined by the aspect ratio of the rendering area so that the unit of distance is equal in both directions. The rendering area is either the entire screen, or window on a computer screen, when viewing a single 2D scene, or the rectangular area defined by the texture used in a **CompositeTexture2D** node, or a **Layer2D** node that contains a subordinate 2D scene description.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

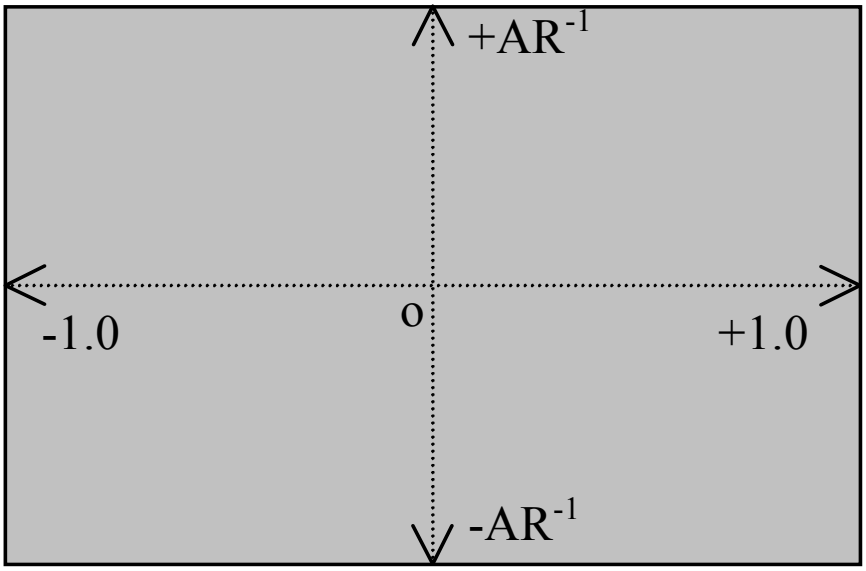

Figure 6 — 2D co-ordinate system (AR = Aspect Ratio)

7.1.1.2.3 3D Co-ordinate System

The 3D co-ordinate system is as described in ISO/IEC 14772-1:1998, subclause 4.4.5. When 2D objects are described in a 3D space, they are drawn in the local (x,y) plane (z=0), and the units used are those of the 3D co-ordinate system for the x and y directions.

7.1.1.2.4 Mixing 2D and 3D scenes

A single BIFS scene may contain both 2D and 3D elements. The following methods exist:

2D primitives may be placed in a 3D scene graph. In this cased, the 2D primitives are drawn in the local (x,y) plane, and use the local coordinate system, restricted to this (x,y) plane.

2D and 3D scenes may be composed and overlapped on the screen using **Layer2D** and **Layer3D** nodes. This is useful, for instance, when it is desirable to have 2D interfaces to 3D worlds ("head up" display), or a 3D insert in a 2D scene.

2D and 3D scenes may be mapped onto any given geometry using the **CompositeTexture2D** and **CompositeTexture3D** nodes. For instance, 2D scenes may be mapped onto animated 3D geometry to perform special effects.

7.1.1.2.5 Drawing Order

It is possible to specify the drawing order of elements of the scene, using the **OrderedGroup** node. This feature may be used for 2D or 3D scenes. 2D scenes are considered to have zero depth. Nonetheless, it is important to be able to specify the order in which 2D objects are composed, in order to describe their apparent depths. 3D scenes may use the drawing order facility to solve conflicts of coplanar polygons or other rendering optimizations.

The following rules determine the drawing order, including conflict resolution for objects having the same drawing order:

1. The object having the lowest drawing order shall be drawn first (taking into account negative values).

2. Objects having the same drawing order shall be drawn in the order in which they appear in the scene description.

7.1.1.2.6 Pixel and Meter metrics

In addition to meter-based metrics, it is also possible to use pixel-based metrics. In this case, 1 meter is set to be equal to the distance between two pixels. This applies to both the horizontal (x-axis) and vertical (y-axis) directions.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

The selection of the appropriate metrics is performed by the content creator. In particular, it is controlled by the `BIFSConfig` syntax (see 8.5.2).

When `pixelMetric` is set to 1, pixel metrics shall be used for the entire scene. This implies that rendered node sizes (such as for a Rectangle) and rendered node positions are integers. If non-integer values appear due to for example scaling, rounding shall be implied towards -infinity.

7.1.1.2.7 Nodes and fields

7.1.1.2.7.1 Nodes

The BIFS scene description consists of a collection of nodes that describe the scene structure. An audio-visual object in the scene is described by one or more nodes, which may be grouped together (using a grouping node). Nodes are grouped into node data types (NDTs) and the exact type of the node is specified using a `nodeType` field.

An audio-visual object may be completely described within the BIFS information, e.g. **Box** with **Appearance**, or may also require elementary stream data from one or more audio-visual objects, e.g. **MovieTexture** or **AudioSource**. In the latter case, the node includes a reference to an object descriptor that indicates which elementary stream(s) is (are) associated with the node, or directly to a URL description (see ISO/IEC 14772-1:1998, subclause 4.5.2). With the exception of the **Anchor** and **Script** nodes, a **url** field may only refer to content that conforms to a valid profile and level for the terminal. When a **url** field is not an OD ID url, the behaviour of the terminal is unspecified.

In the case of **InputSensor**, the node includes a reference to an object descriptor that indicates which user interaction stream is associated with the node.

7.1.1.2.7.2 Fields and Events

See ISO/IEC 14772-1:1998, subclause 5.1.

7.1.1.2.7.3 Object descriptor references in URL fields

The **url** fields in several nodes contain references to media streams. Depending on the profile and level settings (see subclause 7.9), references to media streams are made through object descriptor Ids. The textual syntax for the **url** fields in this case is as follows:

“od:<number>” - refers to the object descriptor with the id <number>.

“od:<number>#<segmentName>” - refers to the stream object defined within the object descriptor with the id <number> that has the name <segmentName>.

“od:<number>#<segmentName1>:<segmentName2>” - refers to all stream objects defined within the object descriptor with the id <number> that start at the same time or later as <segmentName1> and that end at the same time or earlier than <segmentName2>

“od:<number>#<segmentName1>+” - refers to all stream objects defined within the object descriptor with the id <number> start start at the same time or later as <segmentName1> until the end of the media stream.

7.1.1.2.7.4 Routing of Nodes

If a node is routed, the target node of this route becomes as if it were a USE of the source. If the source is a USE itself, then it becomes as if it were a USE of the original DEF. If the source node is not an existing DEF or USE then it becomes an implicit DEF with the target node on the route becoming a USE of it. This behavior for node routing is thus copy by reference, rather than copy by value for the basic fields. An implementation shall behave as if it holds a reference count for nodes that are DEF’d, whether implicit or explicit. The reference count will be the number of nodes that use the definition (the count starting at one for the original DEF). Any nodes overwritten by the routing will have their reference count decremented. When the node is no longer required, i.e. the reference count has been reduced to zero, the node is deleted and is no longer available for re-use.

7.1.1.2.8 Internal, ASCII and Binary Representation of Scenes

ISO/IEC 14496-1 describes the attributes of audio-visual objects using node structures and fields. These fields can be one of several types (see 7.1.1.2.7.2). To facilitate animation of the content and modification of the objects’ attributes in time, within the terminal, it is necessary to use an internal representation of nodes and fields as described in the node specifications (see 7.2). This is essential to ensure deterministic behaviour in the terminal’s compositor, for instance when applying ROUTEs or differentially coded BIFS-Anim frames. The observable behaviour of compliant terminals shall not be affected by the way in which they internally represent and transform data; that is, they shall behave as if their internal representation is as defined herein.

However, when encoding the BIFS scene description, different attributes may need to be quantized or compressed appropriately. Thus, the binary representation of fields may differ according to the types of fields, or according to the precision needed to represent a given audio-visual object's attributes. The semantics of nodes are described in subclause

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

7.2. The binary syntax which represents the binary format as transported in streams conforming to ISO/IEC 14496-11 is provided in clause 8 and uses the node coding parameters provided as an electronic attachment.

7.1.1.2.8.1 Binary Syntax Overview

7.1.1.2.8.1.1 Scene Description

The entire scene is represented by a binary encoding of the scene graph. This encoding restricts the VRML grammar as defined in ISO/IEC 14772-1:1997, Annex A, but still enables the representation of any scene that can be generated by this grammar.

EXAMPLE — One example of the grammatical differences is the fact that all ROUTEs are represented at the end of a BIFS scene, and that a global grouping node is required at the top level of the scene.

7.1.1.2.8.1.2 Node Description

Node types are encoded according to the context of the node. This improves efficiency by exploiting the fact that not all nodes are valid at all places in the scene graph. In many instances, only one of a subset of all BIFS nodes is valid at a particular place in the scene graph, and hence in the bitstream.

7.1.1.2.8.1.3 Fields description

Fields may be quantized to improve compression efficiency. Several aspects of the inverse quantization process can be controlled by adjusting the parameters of the **QuantizationParameter** node.

7.1.1.1.1.1 ROUTE description

All ROUTEs are described at the end of the scene. This improves bit efficiency by grouping these elements in a single location in the bitstream and removes the need for switches in the syntax to allow ROUTEs and nodes to be described in a mixed format.

7.1.1.2.9 Basic Data Types

There are two general classes of fields and events: fields/events that contain a single value (e.g. a single number or a vector), and fields/events that contain multiple values. Multiple-valued fields/events have names that begin with MF, whereas single valued begin with SF.

7.1.1.2.9.1 Numerical data and string data types

7.1.1.2.9.1.1 Introduction

For each basic data type, single field and multiple field data types are defined in ISO/IEC 14772-1:1998, subclause 5.2. Some further restrictions are described herein.

7.1.1.2.9.1.2 SFInt32/MFInt32

When routing values between two SFInt32s note shall be taken of the valid range of the destination. If the value being conveyed is outside the valid range, it shall be clipped to be equal to either the maximum or minimum value of the valid range, as follows:

if x > max, x := max

if x < min, x := min

7.1.1.2.9.1.3 SFTime

The SFTime field and event specifies a single time value. Time values shall consist of 64-bit floating point numbers indicating a duration in seconds or the number of seconds elapsed since the origin of time as defined in the semantics for each SFTime field.

7.1.1.2.9.2 Node data types

Nodes in the scene are also represented by a data type, namely SFNode and MFNode types. ISO/IEC 14496-1 also defines a set of sub-types, such as SFColorNode, SFMaterialNode. These node data types (NDTs) allow efficient binary representation of BIFS scenes, taking into account the usage context to achieve better compression. However, the generic SFNode and MFNode types are sufficient for internal representations of BIFS scenes.

7.1.1.2.10 Attaching nodeIDs to nodes

Each node in a BIFS scene graph may have a `nodeID` associated with it, to be used for referencing. ISO/IEC 14772-1:1998, subclause 4.6.2, describes the DEF statement which is used to attach names to nodes. In BIFS scenes, an

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

integer value is used for the same purpose for `nodeIDs`. The number of bits used to represent these integer values is specified in the `BIFSConfig` syntax (see 8.5.2).

The following restrictions apply:

a) Nodes are identified by the use of `nodeIDs`, which are binary numbers conveyed in the BIFS bitstream.

b) The scope of `nodeIDs` is given in 7.1.1.1.5.

c) No two nodes in the scene graph may have the same `nodeID` at any point in time.

Nodes that have been assigned a `nodeID` may be re-used, as described in ISO/IEC 14772-1:1998, subclause 4.6.3. Note that this mechanism results in a scene description that is a directed acyclic graph, rather than a simple tree.

The mechanisms that allow modifications to the BIFS scene also depend on the use of `nodeIDs`.

7.1.1.2.11 Standard Units

As described in ISO/IEC 14772-1:1998, subclause 4.4.5, the standard units used in the scene description are the following:

Table 1 — Standard units

Category	Unit
Distance	Meter
Color Space	RGB [0,1] [0,1] [0,1]
Time	Seconds
Angle	Radians

7.1.1.2.12 Mapping of Scenes to Screens

BIFS scenes may contain still images and videos that are to be pixel-copied to the rendering device using their native dimensions as produced at the output of their terminals. The **Bitmap** node (see 7.2.2.22) provides a screen-aligned geometry that has the pixel dimensions of the texture that is mapped onto it.

NOTE — When **Bitmap** is used, the same scene will appear differently on screens with different resolutions. BIFS scenes that do not use the **Bitmap** node are independent from the screen on which they are viewed.

7.1.1.2.12.1 Transparency of visual objects

Content complying with ISO/IEC 14496-1 may include still images or video sequences with representations that include alpha values. These values provide transparency information and are to be treated as specified in ISO/IEC 14772-1:1998, subclause 4.14. For video sequences represented according to ISO/IEC 14496-2, transparency is handled as specified in ISO/IEC 14496-2.

7.1.1.2.13 Special considerations for audio

7.1.1.2.13.1 Audio sub-graphs

Audio nodes are used to build audio scenes in the terminal from audio sources coded with tools specified in ISO/IEC 14496-3. The audio scene description capabilities provide two functionalities:

"Physical modelling" composition for virtual-reality applications, where the goal is to recreate the acoustic space of a real or virtual environment.

"Post-production" composition for traditional content applications, where the goal is to apply high-quality signal processing transformations.

Audio may be included in either 2D or 3D scene graphs. In a 3D scene, the audio may be spatially presented to sound as though it originates from a particular 3D direction, according to the positions of the object and the listener.

The **Sound**, **DirectiveSound, WideSound** and **SurroundingSound** nodes are used to attach audio to 3D scene graphs and the **Sound2D** node is used to attach audio to 2D scene graphs. However, by use of the **Transform3DAudio** node 3D audio nodes can also be used in 2D scenes. As with visual objects, an audio object represented by one of these nodes has a position in space and time, and is transformed by the spatial and grouping transforms of nodes hierarchically above it in the scene.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

The nodes below the **Sound, DirectiveSound, WideSound, SurroundingSound**, **Sound2D** nodes, however, constitute an audio sub-graph.This sub-graph is used to describe a particular audio object through the mixing and processing of several audio streams. Rather than representing a hierarchy of spatio-temporal transformations, the nodes within the audio sub-graph represent a signal flow graph that describes how to create the audio object from the audio coded in the AudioSource streams. That is, each audio sub-graph node (**AudioSource**, **AudioMix**, **AudioSwitch**, **AudioFX**, **AudioClip**, **AudioBuffer**, **AdvancedAudioBuffer**, **AudioDelay, AudioChannelConfig**) accepts one or several channels of input audio, and describes how to turn these channels of input audio into one or more channels of output. The only sounds presented in the audio-visual scene are those which are the output of audio nodes that are children of a **Sound** / **DirectiveSound** / **WideSound** / **SurroundingSound** / **Sound2D** node (that is, the "highest" outputs in the audio sub-graph). The remaining nodes represent "intermediate results" in the sound computation process and the sound represented therein is not presented to the user.

The normative semantics of each of the audio sub-graph nodes describe the exact manner in which to compute the output audio from the input audio for each node based on its parameters.

7.1.1.2.13.2 Overview of sound node semantics

This subclause describes the concepts for normative calculation of the audio objects in the scene in detail, and describes the normative procedure for calculating the audio signal which is the output of a **Sound, DirectiveSound**, **WideSound**, **SurroundingSound, Sound2D** node given the audio signals which are its input.

Recall that the audio nodes present in an audio sub-graph do not each represent a sound to be presented in the scene. Rather, the audio sub-graph represents a signal-flow graph which computes a single (possibly multi-channel) audio object based on a set of audio inputs (in **AudioSource** nodes) and parametric transformations. The only sounds which are presented to the listener are those which are the "output" of these audio sub-graphs, as connected to a **Sound**, **DirectiveSound**, **WideSound**, **SurroundingSound**, **Sound2D** node. This subclause describes the proper computation of this signal-flow graph and resulting audio object.

As each audio source is decoded, it produces data that is stored in composition memory (CM). At a particular time instant in the scene, the compositor shall receive from each audio decoder a CM such that the decoded time of the first audio sample of the CM for each audio source is the same (that is, the first sample is synchronized at this time instant). Each CM will have a certain length, depending on the sampling rate of the audio source and the clock rate of the system. In addition, each CM has a certain number of channels, depending on the audio source .

Each node in the audio sub-graph has an associated input buffer and output buffer, except for the **AudioSource** node which has no input buffer. The CM for the audio source acts as the input buffer of audio for the **AudioSource** with which the decoder is associated. As with CM, each input and output buffer for each node has a certain length, and a certain number of channels.

As the signal-flow graph computation proceeds, the output buffer of each node is placed in the input buffer of its parent node, as follows:

If an audio node, *N*, has *n* children, and each of the children produces *k(i)* channels of output, for 1 <= *i* <= *n*, then the node, *N*, shall have *k*(1) + *k*(2) + ... + *k*(*n*) channels of input, where the first *k*(1) channels [number 1 through *k*(1)] shall be the channels of the first child, the next *k*(2) channels [number *k*(1)+1 through *k*(1)+*k*(2)] shall be the channels of the second child, and so forth.

Then, the output buffer of the node is calculated from the input buffer based on the particular rules for that node.

7.1.1.2.13.2.1 Sample-rate conversion

If the various children of a **Sound**, **DirectiveSound**, **WideSound**, **SurroundingSound**, **Sound2D** node do not produce output at the same sampling rate, then the lengths of the output buffers of the children do not match, and the sampling rates of the children's' output must be brought into alignment in order to place their output buffers in the input buffer of the parent node. The sampling rate of the input buffer for the node shall be the fastest of the sampling rates of the children. The output buffers of the children shall be resampled to be at this sampling rate. The particular method of resampling is non-normative, but the quality shall be close in accuracy to the DAC that the signal is targeted for, i.e. according to the rule `dB SNR = 6 * (nbits –1)`, where `nbits` is the number of bits corresponding to the maximum bit depth of any of the signals being so converted and/or composited. Aliasing artifacts may be at this level of signal-to-noise ratio. The noise level due to arithmetic accuracy and other uncorrelated noise sources should be below the rule `dB SNR = 6* nbits`.

The output sampling rate of a node shall be the output sampling rate of the input buffers after this resampling procedure is applied.

Content authors are advised that content which contains audio sources operating at many different sampling rates, especially sampling rates which are not related by simple rational values, may produce scenes with a high computational complexity.

EXAMPLE — Suppose that node N has children M1 and M2, all three audio nodes, and that M1 and M2 produce output at S1 and S2 sampling rates respectively, where S1 > S2. Then if the decoding frame rate is F frames per second, then M1's output buffer will contain S1/F samples of data, and M2's output buffer will contain S2/F samples of data. Then, since M1 is the faster of the children, its output buffer values are placed in the input buffer of N. The output buffer of M2 is

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

resampled by the factor S1/S2 to be S1/F samples long, and these values are placed in the input buffer of N. The output sampling rate of N is S1.

7.1.1.2.13.2.2 Number of output channels

If the **numChan** field of an audio node, which indicates the number of output channels, differs from the number of channels produced according to the calculation procedure in the node description, or if the **numChan** field of an **AudioSource** node differs in value from the number of channels of an input audio stream, then the **numChan** field shall take precedence when including the source in the audio sub-graph calculation, as follows:

If the value of the **numChan** field is strictly less than the number of channels produced, then only the first **numChan** channels shall be used in the output buffer.

If the value of the **numChan** field is strictly greater than the number of channels produced, then the "extra" channels shall be set to all 0's in the output buffer.

7.1.1.2.13.3 Audio-specific BIFS Nodes

In the following table, nodes that are related to audio scene description are listed.

Table 2 — Audio-Specific BIFS Nodes

Node	Purpose	Subclause
AudioBuffer, AdvancedAudioBuffer	Interactively trigger snippets of sound	7.2.2.8, 7.2.2.2
AudioChannelConfig	Label channel configuration of audio data	7.2.2.9
AudioClip	Insert an audio clip into a scene	7.2.2.10
AudioDelay	Add delay to sound	7.2.2.11
AudioFX	Apply post-production effects to sound	7.2.2.12
AudioMix	Mix sounds	7.2.2.14
AudioSource	Define audio source input to a scene	7.2.2.15
AudioSwitch	Switching of audio sources in a scene	7.2.2.16
ListeningPoint	Define listening point in a scene	7.2.2.79
Sound, Sound2D, DirectiveSound, WideSound, SurroundingSound	Define properties of sound	7.2.2.121, 7.2.2.122, 7.2.2.47, 7.2.2.147, 7.2.2.127

7.1.1.2.13.4 Spatialization of sound sources according to the acoustic environment

This specification contains a set of nodes of extended node types, that can be used to include positional and directive sound sources to 3-D BIFS scenes, and process them in a way that the acoustics of the environment is taken into account. These nodes enable parametrization and rendering of the acoustic properties of a virtual environment according to the current relative positions of the sound source, the listening point, and the acoustically relevant objects in the BIFS scene. Such properties are, e.g., room reverberation time (and other statistical room acoustic parameters), speed of sound, acoustic properties of surfaces, and sound source directivity. Functionalities that are made possible with these parameters include immersive audiovisual rendering, room acoustic modeling, and enhanced 3-D sound presentation.

Two distinct approaches of acoustic environment rendering are incorporated in the 3-D sound processing. One is based on physical, or geometrical modeling of the acoustic scene while the second is based on the perceptual description of room acoustic effects. These two schemes of virtual acoustics rendering are referred to as the *physical* and the *perceptual* approach.

The nodes that are involved in the sound environment modeling are **AcousticScene**, **AcousticMaterial**, **DirectiveSound**, and **PerceptualParameters**, and their main functionalities are presented in the table below, and the rendering scheme where they are used is listed in the rightmost column:

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Table 3 — Nodes for environmental spatialization of sound

Node	Purpose	Approach	Subclause
AcousticScene	Restrict each audio rendering process to a defined 3-D region in the BIFS scene, and specify a reverberation time that is applied to the sound sources currently within that region.	physical	7.2.2.3
AcousticMaterial	Define sound reflectivity and transmission properties (along with the visual properties) for each acoustically relevant (flat, polygonal) surface.	physical	7.2.2.1
DirectiveSound	Define a directive sound source that also enables natural distance dependent attenuation and air absorption modeling, as well as rendering of the propagation delay between the source and the listener.	physical and perceptual	7.2.2.47
PerceptualParameters	Node for attaching perceptual properties to a directive sound source (DirectiveSound) in order to simulate virtual room effects that do not need to relate to the geometrical and/or visual BIFS scene.	perceptual	7.2.2.97

In the following, overviews of the physical and perceptual audio rendering schemes are presented.

7.1.1.2.13.4.1 Physical approach

In this approach the acoustics rendering is defined as creating a virtual auditory environment that models an existent or non-existent space. This rendering is called *auralization,* the relation of which to graphics (visualization) is understood as the creation of audiovisual scenes that are perceptually (visually and aurally) relevant. An example of this could be a virtual concert performance, where the acoustical behavior of the space as well as the graphical outlook is modeled. Another example could be a scene, where the listener moves from a very small room to a larger hall, and the changes in the acoustic and graphical rendering is immediately perceived. Also sound sources without a room acoustic response but with effects such as source directivity, Doppler effect, and echoes (distinctive sound reflections) can be modeled. The acoustical behaviors and properties are:

Acoustic properties of surface materials (walls), that enable modeling of sound reflections of surfaces, as well as transmission of sound through them. This way sound reflections are tracked and rendered according to the geometry of the walls and positions of the sound sources and the listener. Obstruction effects are automatically rendered when walls or obstacles are present between the source and the listener

Reverberation time of a specified region in the scene. This enables modeling of reverberating spaces by a simple parameter, and without the necessary need to describe the physical walls of a room.

Acoustic properties of the sound transmitting medium. These include the speed of sound, distance dependent attenuation and lowpass filtering effect caused by air absorption (see ISO 9613-1:1993). Speed of sound is used to control the sound propagation delay between source and the listener, and therefore also the strength of the Doppler effect which depends on the relative motion between the source and the listener.

Directivity characteristics of sound sources. This enables flexible modeling of different sound sources (e.g., human speaker, or a musical instrument). The directivity patterns can be frequency dependent, or it can be defined by a direction dependent coefficient, or in the simplest case the source can be omnidirectional.

In the physical approach, the *geometrical* and *physical sound propagation operator* is used in real time during playback in order to derive the auralization signal processing parameters to be applied to each sound source signal. This propagation operator exploits the knowledge of the positions of the sound sources and the listener relative to the walls to compute the arrival time, amplitude (and spectrum) and direction of arrival for each early reflection. This computation is performed in real time for a limited number of reflections per sound source, with dynamic refresh of reflection parameters according to movements of the sound sources or the listener.

7.1.1.2.13.4.2 Perceptual approach

In this model, the sound transformation associated with room reflections and reverberation is described by a set of perceptual attributes (such as source presence and brilliance, room reverberance, envelopment). These attributes may be manipulated directly and individually for each sound source in the scene.

This approach provides simple and intuitive parameters to the content provider, allowing:

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Manipulation of environmental effects for each sound event directly (without requiring that the source or the point of view be moved).

Sound design adjustments beyond the physical constraints implied by the graphic representation, for example:

Distorted or exaggerated distance sensation and room-related effects

Unconstrained spatial sound effects for audio-only scene nodes (no visual correspondence) or when the point of view is out of the room

In this approach, an absolute (exocentric) representation of the sound scene containing several sources and the listener can be manipulated as follows:

The environment (room) is described by setting the values of the perceptual attributes for a reference source-listener distance. These attributes and their values make up a "preset", which specifies, at that reference distance and for an omnidirectional sound source, the delay and intensity of the early reflection, as well as the delay, decay time and spectrum of the late reverberation.

The sound transformation to be applied to each sound event is derived from the above preset by use of a *perceptual sound propagation operator* which takes into account the relative positions and orientations of the sources and the listener, and a model of the directivity of sound sources.

In this model, only the *relative* positions and orientations of the sound sources with respect to the listener are taken into account. The model does not exploit any knowledge of wall positions in order to compute the parameters of the early reflections. The temporal pattern of the early reflections is determined by the definition of the environment “preset”. The perceptual sound propagation operator adjusts one perceptual attribute (called "source presence") according to source-listener distance. Adjusting this single parameter produces a convincing sensation of proximity or remoteness of the sound source. Additionally, the operator takes into account the orientation of the source and its directivity pattern.

7.1.1.2.13.5 Channel configuration aspects in the audio subtree

Audio decoders require a channel configuration that is either known implicitly or is conveyed by some configuration information. Note that ISO/IEC 14496-3, Subpart 4 forbids the use of implicit channel mappings via the PCE (program config element) in subclause 4.5.1.2.1, if MPEG-4 Audio is used together with MPEG-4 Systems audio compositor (i.e., AudioBIFS). In an audio subtree the channel configuration of the decoders or from the DecoderSpecificInfo can normally be used for the loudspeaker mapping behind the sound node, especially in the multichannel case (numChan > 1) if the phase group flags in the audio nodes has been set. Therefore an MPEG-4 Player implementation has to pass this information from the decoder output via the audio nodes to the presenter. Some audio nodes (**AudioMix**, **AudioSwitch** and **AudioFX**) have channel-variant behavior. For these nodes conflicts in the channel configuration transmission can occur, as illustrated in Figure 7 where two different conflicts are shown.

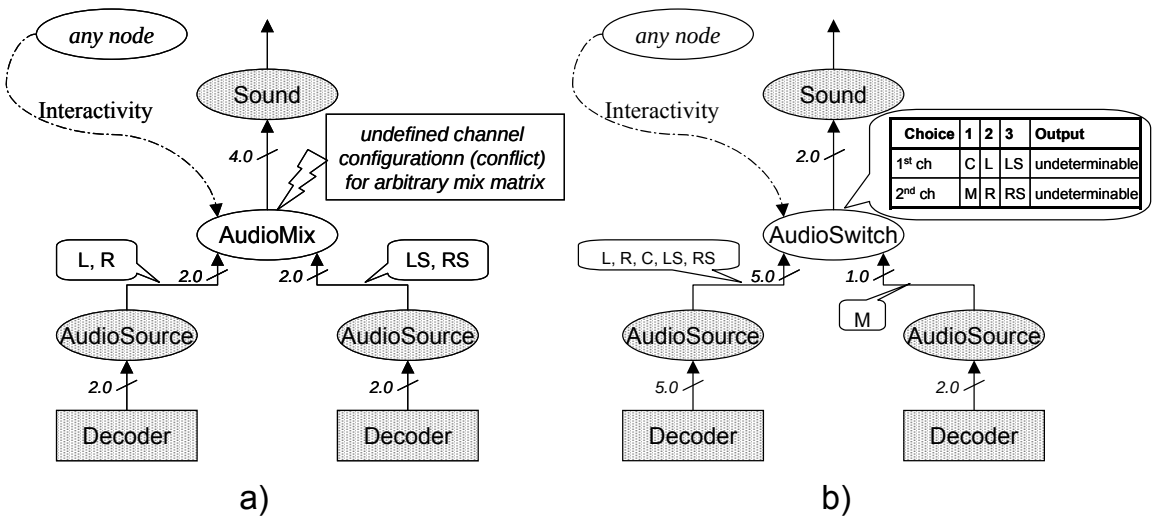

Figure 7 — Channel configuration flow conflicts

The first conflict occurs behind the mix node (Figure 7 a)), where a mix of a first stereo signal and a second stereo signal shall be mixed into 4 channels. The resulting channel configuration after the mix is undefined, since it is not obvious what kind of configuration has been produced. The second conflict occurs in a sequence of whichChoice field updates (Figure 7

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

b)) behind the **AudioSwitch** node (choice 1…3). In this sequence different channels from the Source node and a single channel from a second Source node are sequentially selected. There is no way to know the desired channel configuration at the output in both cases. Introucing an **AudioChannelConfig** node behind the conflicting node solves this problem by labeling the channels with a determined configuration.

The format of the transmitted channels is restricted to the formats of the decoder configurations. Despite these configurations, presets of other configurations are possible and can be composed with the help of an AudioBIFS scene. Two problems can arise in general: missing channel descriptions and composition of formats. It is not foreseen in the channel configuration table of the MPEG-1/2 encoders to transmit multichannel subsets, for example only the 2 surround channels LS, RS. The composition of multichannel formats with the help of several 2-channel streams, for example encoded in the MPEG-1/2 Layer 3 format, cannot be done unambiguously without the **AudioChannelConfig** node as illustrated in Figure 8: a composition of 3 stereo channels to a 6-channel mix can be labeled as a 5.1 multichannel set or as a 6-channel Ambisonics® signal.

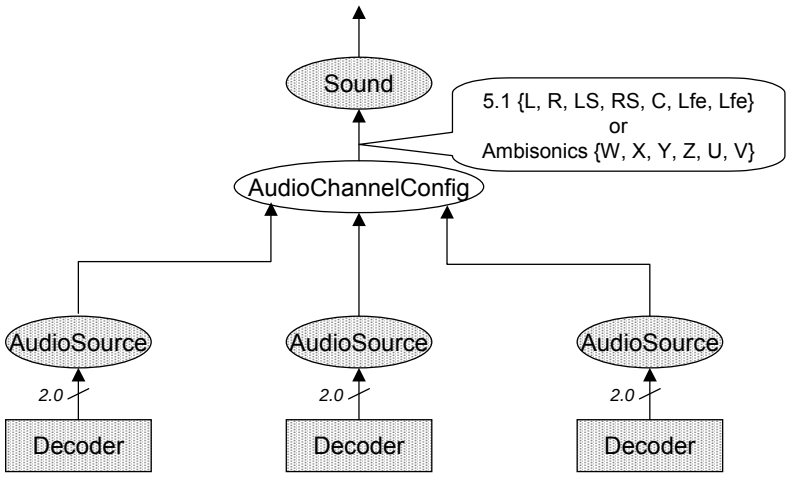

Figure 8 — The audioChannelConfig node for the composition of new formats

7.1.1.2.13.6 Implementation details on SurroundingSound transformations

For the case where the SurroundingSound node is "fed" with multi-channel flow labeled as a 1st order Ambisonics® sound field, transformations rely on formulae that already exist. Rotation effects are a quite trivial matter and involve basic trigonometry formulae (see Figure 9). Angular or perspective distortions derive from the existing "Forward dominance" effect.

Forward dominance is an effect to parametrically enlarge or narrow the frontal sound scene (and inversely the back scene). It derives from the "Lorentz Transform" (see Figure 9) applied to the 1st order spherical harmonics. This transform is parameterized by a value λ which range is]0; +inf[, the value 1 corresponding to "no effect".The angle distortion is such that a sound source that was localized at azimuth θ will be moved to azimuth θ', with: $\cos\theta' = (\mu + \cos\theta)/(1+\mu\cos\theta)$ and $\mu = (\lambda^2-1)/(\lambda^2+1)$.

If the ListeningPoint moves front or back in the scene, one can mimic the distortion of the angular perspective by applying the "Lorentz Transform" (see Figure 9) with a parameter being a function of the ListeningPoint distance (from the SurroundingSound center) weighted by the distortionFactor. If d is the displacement along the back-front axis, one can propose the following mapping law: $\lambda = \exp(-\alpha\, d)$, with α being the distortionFactor.

If the ListeningPoint moves along an arbitrary axis (not necessary the front-back axis), one has to rotate the soundfield towards this axis before applying the Lorentz Transform, then rotate it with the opposite angle, towards the front-back axis again.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

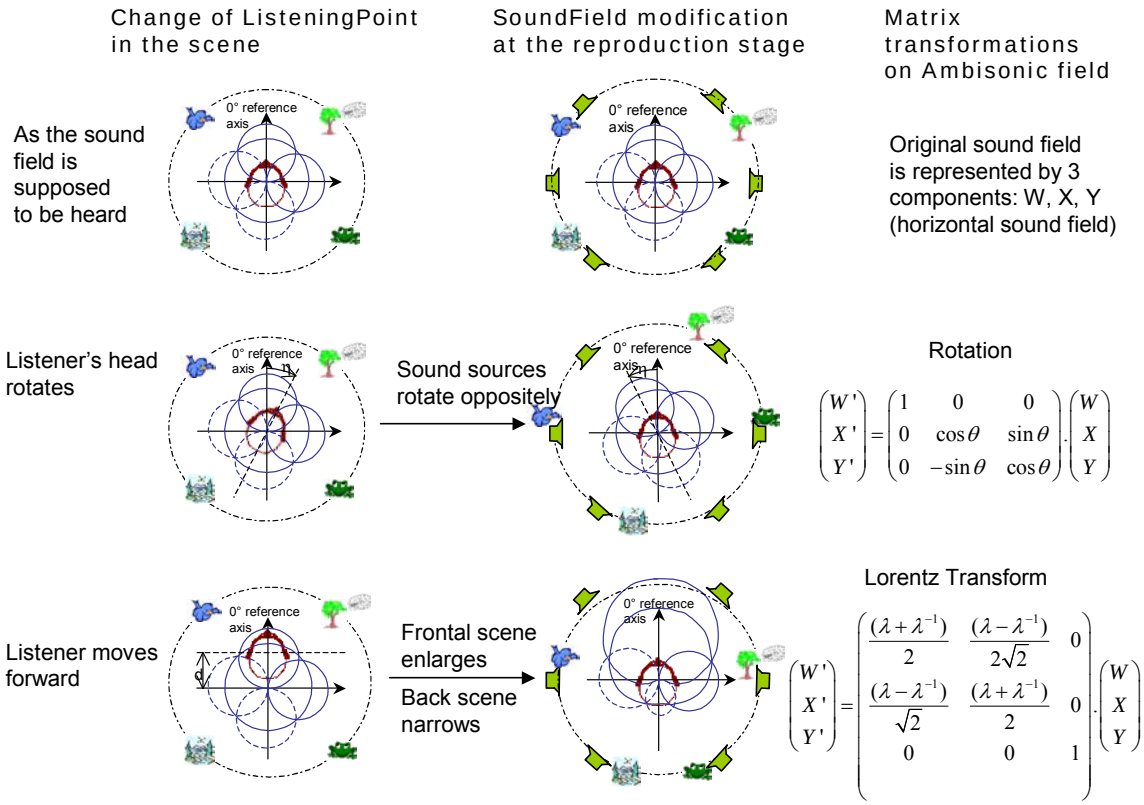

Figure 9 — Ambisonics® Sound Field transformations applied in connection with listener's moves

For non-Ambisonics® materials, angular transformations can be applied by panning the signals over the rendering layout, as if transformations would concern the originally dedicated loudspeakers as being the sources contained in the sound field.

7.1.1.2.14 Bindable Children Nodes

MPEG-4 extends the notion of bindable children nodes described in VRML, ISO/IEC 14772-1:1998. As in VRML, the scene maintains a collection of stacks, one for each type of bindable node. The bindable nodes are the **Viewpoint**, **Viewport**, **Background**, **Background2D**, **Fog**, and **NavigationInfo** nodes. In MPEG-4 some of these nodes may be bound to stacks held in the **Layer2D**, **Layer3D**, **CompositeTexture2D**, and **CompositeTexture3D** grouping nodes. Bindable nodes can be shared between these grouping nodes and the scene using the **set_bind** mechanism, but in the absence of such an event, the top most bindable node within a sub-scene will bind to its inner most grouping node that holds a stack for it.

The following rules shall apply in the absence of a set_bind event:

- The top-most **Background**, **Fog**, **NavigationInfo**, and **Viewpoint** nodes in a **Layer3D** or **CompositeTexture3D** node shall be bound to the respective stacks within that node.
- The top-most **Background2D** and **Viewport** nodes in a **Layer2D** or **CompositeTexture2D** node shall be bound to the respective stacks within that node.
- The top-most **Background2D** node in a **Layer3D** or **CompositeTexture3D** node shall be bound to the same stack as the **Background** node and shall appear in the **background** field of those nodes.
- The top-most bindable node that is not a child of any grouping node that holds a stack for it shall be bound to the respective scene stack for that node.

NOTE - For example, a **Background2D** node that is a child of an **OrderedGroup** or **Group** node would bind to the scene stack. A **Background2D** node that is a child of a **Layer3D** or **CompositeTexture3D** node would be bound

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

using the same stack as for **Background** in that node and would render behind all other geometries in the frame of that node. A **Background** node in a **Layer2D** would bind to the scene stack.

7.1.1.3 Sources of modification to the scene

7.1.1.3.1 Interactivity and behaviors

To describe interactivity and behavior of scene objects, the event architecture defined in ISO/IEC 14772-1:1998, subclause 4.10, is used. Sensors and routes describe interactivity and behaviors. Sensor nodes generate events based on user interaction or a change in the scene. These events are routed to interpolator or other nodes to change the attributes of these nodes. If routed to an interpolator, a new parameter is interpolated according to the input value, and is finally routed to the node which must process the event.

Events shall be generated and evaluated in the order in which their ROUTEs appear in or are inserted into the scene. Events are collected and applied to the scene together, with subsequently generated events collected and applied together repeatedly in an event cascade.

NOTE – the event cascade model differs from a depth-first, immediate execution model. In the latter, side effects from node deletions can create race conditions which would lead to different results on different implementations.

7.1.1.3.1.1 Attaching ROUTEIDs to routes

`ROUTEIDs` may be attached to routes using the DEF mechanism, described in ISO/IEC 14772-1:1998, subclause 4.6.2. This allows routes to be subsequently referenced in BIFS-Command structures. `ROUTEIDs` are integer values and the namespace for routes is distinct from that of `nodeIDs`. The number of bits used to represent these integer values is specified in the BIFS `DecoderConfigDescriptor`.

The scope of `ROUTEIDs` is defined in see 7.1.1.1.5. The following restrictions apply:

a) Routes are identified by the use of `ROUTEIDs`, which are binary numbers conveyed in the BIFS bitstream.

b) The scope of `ROUTEIDs` is given in 7.1.1.1.5.

c) No two routes in the scene graph may have the same `ROUTEID` at any point in time.

The mechanisms that allow modifications to the BIFS scene also depend on the use of `nodeIDs` (see 7.1.1.2.10). The USE mechanism shall not be used with routes.

7.1.1.3.1.2 Conditional node

The **Conditional** node (see 7.2.2.37) allows BIFS-Commands to be described in the scene which shall only be applied to the scene graph when an event is received on one of the **Conditional** node's inputs.

7.1.1.3.2 External modification of the scene: BIFS-Commands

The BIFS-Command mechanism enables the change of properties of the scene graph, its nodes and behaviors.

EXAMPLE — **Transform** nodes can be modified to move objects in space; **Material** nodes can be changed to modify an object's appearance, and fields of geometric nodes can be totally or partially changed to modify the geometry of objects.

7.1.1.3.2.1 Overview

BIFS-Commands are used to modify a set of properties of the scene at a given time instant in time. Commands are grouped into `CommandFrames` (see 8.6.2) in order to be able to send several commands in a single access unit. The following four basic commands are defined:

1. Replacement of an entire scene

2. Insertion

3. Deletion

4. Replacement

The first of these commands allows the replacement of the entire BIFS scene. The replacement of the entire scene requires a scene graph representing a valid BIFS scene to be transmitted. The `SceneReplace` command is the only random access point in the BIFS stream.

The other three commands can be used to update the following structures:

1. A node

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

2. An eventIn, exposedField or an indexed value in an MFField

3. A ROUTE

In order to modify the scene the sender must transmit a BIFS `CommandFrame` that contains one or more update commands. A single source of BIFS-Commands is assumed. The identification of a node in the scene is provided by a `nodeID`. Note that it is the sender's responsibility to provide this `nodeID`, which must be unique (see 7.1.1.1.5). The identification of a node's fields is provided by sending the `INid` of the field (see node coding tables in electronic attachment).

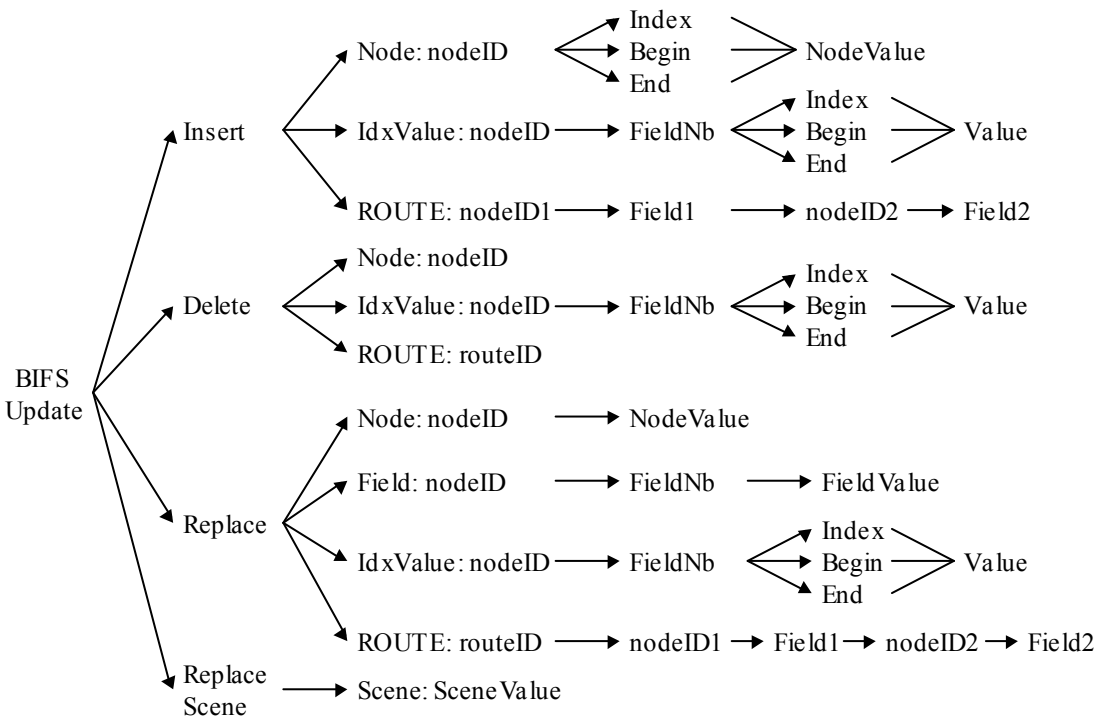

Figure 10 — BIFS-Command Types

7.1.1.3.2.2 Modification of indexed values

Insertion of an indexed value in a field implies that all later values in the field have their indices incremented and the length of the field increases accordingly. Appending a value to an indexed value field also increases the length of the field but the indices of existing values in the field do not change.

Deletion of an indexed value in a field implies that all later values in the field have their indices decremented and the length of the field decreases accordingly.

7.1.1.3.2.3 Timing of BIFS-Commands

The time at which a BIFS-Command is applied shall be the composition time stamp of the access unit in which the command is contained, as defined in the sync layer (see 7.3.2, ISO/IEC 14496-1).

7.1.1.3.3 External animation of the scene: BIFS-Anim

BIFS-Anim provides for the continuous update of the certain fields of nodes in the scene graph. BIFS-Anim is used to integrate different kinds of animation, including the ability to animate face models as well as meshes, 2D and 3D positions, rotations, scale factors, and color attributes. Although BIFS-Anim and BIFS-Command have the same elementary stream type (see Table 5, ISO/IEC 14496-1) they may not occupy the same elementary stream. BIFS-Anim information is conveyed in a separate elementary stream from that which carries BIFS-Command elements.

7.1.1.3.3.1 Overview

BIFS-Anim elementary streams consist of a sequence of `AnimationFrames`. The `AnimationMask`, which is required to interpret these `AnimationFrames`, is transmitted in the `DecoderSpecificInfo` for the BIFS-Anim elementary stream in the corresponding object descriptor (see 7.2.6.7, ISO/IEC 14496-1).

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

7.1.1.3.3.2 BIFS-Anim configuration

The `AnimationMask` contains one `ElementaryMask` for each node that is to be animated. These `ElementaryMasks` specify the fields that are contained in the `AnimationFrames` for a given animated node, and their associated quantization parameters. Only eventIn or exposedField fields that have an animation method (see node coding tables electronic attachment and 7.1.1.3.3.3) can be modified using BIFS-Anim. Such fields are called dynamic fields. In addition, the animated field must be part of an updateable node; that is, a node that has been assigned a `nodeID`. The `AnimationMask` is composed of several elementary masks defining these parameters.

7.1.1.3.3.3 BIFS-Anim animation parameters

Animation parameters are transmitted as a sequence of `AnimationFrames`. `AnimationFrames` specify the values of the dynamic fields of updateable nodes that are being animated in BIFS-Anim streams. An `AnimationFrame` contains the new values of all animated parameters at a specified time, unless if it is specified that, for some frames, these parameters are not sent. The parameters can be sent in Intra (the absolute value is sent) and Predictive modes (the difference between the current and previous values is sent).

Animation parameters can be applied to any eventIn or exposedField of any updateable node of a scene which has an assigned animation method (see node coding tables in electronic attachments).

NOTE — Some node tables, in the electronic attachment, contain an eventIn or exposedField that has an animation method but for which there is no associated `dynID`. This is the case when only one exposedField or eventIn in a node has an animation method. In such cases, it is not necessary for the field to have a `dynID` since the terminal can assume that BIFS-Anim animations for this type of node refer to the only dynamic field of the node.

The types of dynamic fields are:

SFInt32/MFInt32

SFFloat/MFFloat

SFRotation/MFRotation

SFColor/MFColor

SFVec2f/MFVec2f

SFVec3f/MFVec3f

SFVec4f/MFVec4f

7.1.1.3.4 Order of application of modifications to the scene

Where modifications to the scene graph, resulting from the use of more than one of the permitted methods, must be applied simultaneously, the following order of application shall be observed:

1. BIFS-Anim
2. Conditional node
3. BIFS-Command

7.2 Node Semantics

7.2.1 Overview

The BIFS nodes include nodes that have been defined in ISO/IEC 14772-1:1998. For these nodes, the semantic information is given by normative reference with any restrictions defined herein.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

7.2.2 Node specifications

7.2.2.1 AcousticMaterial

7.2.2.1.1 Node interface

AcousticMaterial {

field	SFFloat	**reffunc**	0
field	SFFloat	**transfunc**	1
field	MFFloat	**refFrequency**	[]
field	MFFloat	**transFrequency**	[]
exposedField	SFFloat	**ambientIntensity**	0.2
exposedField	SFColor	**diffuseColor**	0.8, 0.8, 0.8
exposedField	SFColor	**emissiveColor**	0, 0, 0
exposedField	SFFloat	**shininess**	0.2
exposedField	SFColor	**specularColor**	0, 0, 0
exposedField	SFFloat	**transparency**	0

}

NOTE - For the binary encoding of this node see node coding tables in electronic attachment.

7.2.2.1.2 Functionality and semantics

The **AcousticMaterial** node is used for attaching acoustic and visual properties to surfaces (planar polygons) defined by an **IndexedFaceSet** node that is a sibling or exist in a sub-graph of a sibling of an **AcousticScene** node. The fields of this node define the visual appearance properties, as well as sound reflection and transmission properties of the **IndexedFaceSet** surfaces it is attached to. It is used in the material field of an **Appearance** node that is attached to a **Shape** node under which the **IndexedFaceSet** is defined. Each polygon in an **IndexedFaceSet** that **AcousticMaterial** is associated with can produce a single specular reflection to sound whenever a corresponding sound image source is visible to the listening point (**Viewpoint** or **ListeningPoint**), or obstruct sound transmission when it appears between the sound source and the listener. Note that these reflectivity and sound transmission properties of a surface are only applied to sounds that are attached to a 3-D scene with a **DirectiveSound** and **WideSound** nodes. The delay of a reflection (a predelay that is added to sound) is computed from the relative distance between the image source corresponding to the reflection, and the speed of sound which is given as a field in the **DirectiveSound** and **WideSound** nodes (see 7.2.2.47).

There are two different ways of defining the reflectivity and transmission properties of **AcousticMaterial:**

The **reffunc** and **refFrequency** fields specify the sound reflectivity of the material. If **refFrequency** is an empty vector, **reffunc** is a system function representation of a linear, time-invariant system, the reflectivity transfer function of a digital filter for that material. Generally, a system function $H(z)$ is represented in the z-domain as a division of the z-transform of the output sequence $Y(z)$ with the z-transform of the input sequence $X(z)$:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}.$$

The reflection function is given as digital filter coefficients in the following order:

$$[b_0 b_1 b_2 \ldots a_1 a_2 \ldots]$$

Thus, a simple scalar value b_0 can be given to a material for frequency-independent reflectivity of a surface. On the other hand, complex reflection functions can also be represented using this formulation. For example, if the **reffunc** field is 1, the amplitude of the reflection of sound off a surface will be the same as that of the incident sound, and if the field is set to 0, no sound will reflect off that surface. The default value of this field is 0, implying no reflectivity.

If **refFrequency** is different from an empty vector, the semantics of the **reffunc** is different than described above. In this case **refFrequency** specifies a set of frequencies (in Hz) at which the gains in **reffunc** field are valid; The filter applied to sound when it is reflected off this surface implements a frequency magnitude response where at the given frequencies (in **refFrequency** field) the gains in **reffunc** field are valid. An example of **refFrequency** field is:

[250 1000 2000 4000],

and an example of **reffunc** in this approach is:

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

[0.75 0.9 0.9 0.2]

The **transfunc** and **transFrequency** fields specify the transmission properties of the material, e.g., the filtering that is applied to sound when it passes through an **IndexedFaceSet** surface this **AcousticMaterial** is attached to, when the **IndexedFaceSet** surface appears on the direct path between the sound source and the listener. The transmission function is given similarly as in the reflectivity in **reffunc** and **refFrequency** fields with two different ways of expressing the filtering.

The fields **ambientIntensity, diffuseColor, emissiveColor,** and **shininess** are used for the visual appearance rendering similarly as in the **Material** node.

7.2.2.2 AdvancedAudioBuffer

7.2.2.2.1 Node interface

AdvancedAudioBuffer {

eventIn	MFNode	**addChildren**	
eventIn	MFNode	**removeChildren**	
exposedField	MFNode	**children**	[]
exposedField	SFBool	**loop**	FALSE
exposedField	SFFloat	**pitch**	1.0
exposedField	SFTime	**startTime**	0
exposedField	SFTime	**stopTime**	0
exposedField	SFTime	**startLoadTime**	0
exposedField	SFTime	**stopLoadTime**	0
exposedField	SFInt32	**loadMode**	0
exposedField	SFInt32	**numAccumulatedBlocks**	0
exposedField	SFInt32	**deleteBlock**	0
exposedField	SFInt32	**playBlock**	0
exposedField	SFFloat	**length**	0.0
field	SFInt32	**numChan**	1
field	MFInt32	**phaseGroup**	[1]
eventOut	SFTime	**duration_changed**	
eventOut	SFBool	**isActive**	

}

NOTE — For binary encoding of this node see node coding tables in electronic attachment

7.2.2.2.2 Functionality and semantics

The **AdvancedAudioBuffer** node provides an interface for stored sound. It can be used instead of the **AudioBuffer** node. This node has corrected functionality and enhanced reload mechanism compared to the **AudioBuffer** node, e.g to accumulate snippets of sound in the **AdvancedAudioBuffer**. These snippets can be accessed directly or as the full accumulated content.

The functionality of the **children, loop**, **pitch, startTime, stopTime, numChan, phaseGroup, duration_changed** and **isActive** fields is described in 7.2.2.8, **AudioBuffer** node.

The **length** field specifies the length in seconds of the audio buffer. Audio data should be buffered at the instantiation of the node and whenever the **length** field changes. Additional modes to control loading, deletion and demand of data can be selected with **loadMode** as defined in Table 4.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Table 4 — loadMode association table

Value	LoadMode	Functionality
0	compatibility mode	Audio data should be buffered immediately after the node has been instantiated and whenever the **length** field changes. **startLoadTime, stopLoadTime, numAccumulatedBlocks, deleteBlock** and **playBlock** have no effect.
1	reload mode	**startLoadTime** and **stopLoadTime** are valid. When the time in **startLoadTime** is reached, the internal buffer is cleared and the samples at the input of the node are stored until **stopLoadTime** is reached or the stored data have the length defined in **length** . If **startLoadTime** >= **stopLoadTime** a data block with the length defined in the **length** field will be loaded when **startLoadTime** is reached. **numAccumulatedBlocks, deleteBlock** and **playBlock** have no effect.
2	accumulate mode	An audio data block defined by the interval between **startLoadTime** and **stopLoadTime** will be appended at the end of the buffer. The index of the internal audio data blocks has to be continous to be addressable. When the limit defined by **length** will be reached loading will be finished. **numAccumulatedBlocks** has no effect.
3	continuous accumulate mode	An audio data block defined by the interval between **startLoadTime** and **stopLoadTime** will be appended at the end of the buffer. The index of the internal audio data blocks has to be continous to be addressable. When the limit defined by **length** will be reached the oldest audio data block(s) has/have to be discarded until the data are stored. **numAccumulatedBlocks** has no effect.
4	accumulate mode with limited number of buffer blocks	In the accumulate mode the number of stored blocks are limited to **numAccumulatedBlocks.** **length** has no effect.
5…7	reserved for ISO use	

A transition from 0 to a value below 0 in the **deleteBlock** field starts deletion of a data block relative to the latest data block in the following modes: 'accumulate mode', 'continuous accumulate mode' and 'accumulate mode with limited number of buffer blocks'. The latest block will be addressed with –1 the previous audio data block with –2 etc.

playBlock defines the block to be played. If **playBlock** is set to 0 (default) the whole content will be played regarding to the **startTime** and **stopTime** conditions (compatibility mode). A value < 0 addresses a block relative to the latest block**.** The latest block will be addressed with –1 the previous audio data block with –2 etc.

7.2.2.3 AcousticScene

7.2.2.3.1 Node interface

```
AcousticScene {
    field          SFVec3f     center          0 0 0
    field          SFVec3f     Size            -1 –1 –1
    field          MFTime      reverbTime      0
    field          MFFloat     reverbFreq      1000
    exposedField   SFFloat     reverbLevel     0.4
    exposedField   SFTime      reverbDelay     0.5
}
```

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

NOTE - For the binary encoding of this node see node coding tables in electronic attachment.

7.2.2.3.2 Functionality and semantics

AcousticScene is a node the parameters of which are used for rendering of the acoustic response of the environment, together with the acoustic reflectivity or transmission defined in the siblings or their sub-graphs of this **AcousticScene**. **AcousticScene** also defines four fields (**reverbTime**, **reverbFreq**, **reverbLevel**, and **reverbDelay**) which can be used to add reverberation to sounds that are affected by this node. Only audio that has been attached to the scene through a **DirectiveSound** node performing the *physical* rendering scheme is spatialized according to these definitions.

Only those **IndexedFaceSet** nodes that **AcousticMaterial** node is associated with, and that are defined in the siblings of **AcousticScene** (or in the sub-graph of the siblings) have effect on the room acoustic response that is applied to sound sources. Only **DirectiveSound** nodes that are currently positioned in a 3-D rectangular region defined by **center** and **size** fields, are affected by these acoustic surfaces. The **size** field defines the size of a rectangular 3-D region where the parameters of **AcousticScene** and the acoustic surfaces in the siblings or sibling sub-graphs of the **AcousticScene** are taken into account in the auralization process (sound processing according to the acoustics of the environment). The default value of this field is –1, -1, -1.

The **center** field specifies the center of the above described region in the local coordinate system of the scene. Only when at the decoder a **DirectiveSound** and the **Viewpoint** (or **ListeningPoint**) are located within the same **AcousticScene** region defined by its **center** and **size** the sound attached to the **DirectiveSound** is heard. The default value (-1, –1, –1) of size equals to an infinite rectangular region (i.e., the sound is heard everywhere in the scene). **DirectiveSound** is rendered at one time only according to one **AcousticScene**, i.e., if the source and the viewpoint are in an overlapping area of several **AcousticScene**s, the one which is the first in the rendering order has effect on the **DirectiveSound.**

The **reverbTime** field specifies the reverberation time (time of 60 dB attenuation in the late reverberation response) at frequencies given in **reverbFreq** field to be applied to each **DirectiveSound** node that is within the 3-D region specified by the **AcousticScene**. This information is used for producing late reverberation at the maximum quality possible. With the default value 0, late reverberation is not added to the room response. It should be noted, however, that this field is useful for enabling simple room response modeling whenever there is not enough computational power to render several room reflections, or when the reflective properties of the surfaces are not specified. I.e., it is possible to specify a reverberant room with the boundaries defined by the **size** and **center** fields, even without specifying the reflectivity of individual surfaces. If only one value of **reverbTime** is given, it is taken as the reverberation time at the 1kHz frequency, and the decision about the frequency dependence of the reverberation time would be decided at the terminal (in natural environments the reverberation time decreases as a function of frequency). An example of **reverbTime** field is:

[2.0 0.5],

and example of **reverbFreq** is:

[0 16000],

yielding a late reverberation with a reverberation time of 2.0 s at 0 Hz frequency, and 0.5 s at 16000 Hz frequency.

reverbDelay specifies the time delay between the direct sound and the start of the reverberation in seconds. **reverbLevel** defines the level of the first output from the reverberator with respect to the direct sound.

In order to define which **AcoustcScene** is applied to **DirectiveSound** in the case that it is positioned in an overlapping area of more than one **AcousticScene**, an **OrderedGroup** can be used above the various **AcousticScene**s.

7.2.2.4 Anchor

7.2.2.4.1 Node interface

```
Anchor {
    eventIn       MFNode    addChildren
    eventIn       MFNode    removeChildren
    exposedField  MFNode    children        []
    exposedField  SFString  description     ""
    exposedField  MFString  parameter       []
    exposedField  MFString  url             []
    eventIn       SFBool    activate
}
```

NOTE — For the binary encoding of this node see node coding tables in electronic attachment.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

7.2.2.4.2 Functionality and semantics

The semantics of the **Anchor** node are specified in ISO/IEC 14772-1:1998, subclause 6.2. ISO/IEC 14496-1 does not support the bounding box parameters (**bboxCenter** and **bboxSize**). Upon reception of an SFBool event of value TRUE on the **activate** eventIn the Anchor node will activate and behavior will be identical to that which would result from activating (e.g. clicking) any geometry contained within its **children** field.

7.2.2.5 AnimationStream

7.2.2.5.1 Node interface

```
AnimationStream {
    exposedField    SFBool      loop            FALSE
    exposedField    SFFloat     speed           1.0
    exposedField    SFTime      startTime       0
    exposedField    SFTime      stopTime        0
    exposedField    MFString    url             [""]
    eventOut        SFBool      isActive
}
```

NOTE — For the binary encoding of this node see node coding tables in electronic attachment.

7.2.2.5.2 Functionality and semantics

The **AnimationStream** node is designed to implement control parameters for a scene description stream.

The **loop**, **startTime**, and **stopTime** exposedFields and the **isActive** eventOut, and their effects on the **AnimationStream** node are described in 7.1.1.1.6.2.

The semantics of the **speed** exposedField are identical to those for the **MovieTexture** node (see 7.2.2.85).

The **url** field specifies the data source to be used. The data source referred to shall be a BIFS-Anim stream (see also 7.1.1.3.3) or a BIFS-Command stream. In both cases, the stream shall operate within the same name scope as the scene containing the **AnimationStream** node.

7.2.2.6 Appearance

7.2.2.6.1 Node interface

```
Appearance {
    exposedField    SFNode      material            NULL
    exposedField    SFNode      texture             NULL
    exposedField    SFNode      textureTransform    NULL
}
```

NOTE — For the binary encoding of this node see node coding tables in electronic attachment.

7.2.2.6.2 Functionality and semantics

The semantics of the **Appearance** node are specified in ISO/IEC 14772-1:1998, subclause 6.3.

The **material** field, if non-NULL, shall contain either a **Material** node or a **Material2D** node depending on the type of geometry node used in the geometry field of the **Shape** node that contains the **Appearance** node. The list below shows the geometry nodes that require a **Material** node, those that require a **Material2D** node and those where either may apply:

Material2D only: **Circle, Curve2D, IndexedFaceSet2D, IndexedLineSet2D, PointSet2D, Rectangle**;

Material only: **Box, Cone, Cylinder, ElevationGrid, Extrusion, IndexedFaceSet, IndexedLineSet, PointSet, Sphere;**

Material2D or **Material**: **Bitmap, Text.**

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Inside a **Shape** node in a 2D context, if no **Appearance** and therefore no **Material2D** is defined, the default values and behavior of the **Material2D** node shall be used. In a 3D context, the default behavior is specified in ISO/IEC14772-1 (the object is unlit and has color 1 1 1).

7.2.2.7 ApplicationWindow

7.2.2.7.1 Node interface

ApplicationWindow {

exposedField	SFBool	**isActive**	FALSE
exposedField	SFTime	**startTime**	0
exposedField	SFTime	**stopTime**	0
exposedField	SFString	**description**	“”
exposedField	MFString	**parameter**	[]
exposedField	MFString	**url**	[]
exposedField	SFVec2f	**size**	0, 0

}

NOTE - For the binary encoding of this node see node coding tables in electronic attachment.

7.2.2.7.2 Functionality and semantics

ApplicationWindow is an SF2DNode that allows an external application such as a web browser to exist within the MPEG-4 scene graph. Unlike a texture node, the windowed region is controlled and rendered by the external application, allowing natural user interaction with the application. The particular application to be opened is signaled in the **url** field, and any required parameters for starting the application may be placed in the **parameter** field.

The position of the application, its dimension and whether the application is active or not, is specified through BIFS scene authoring.

The **startTime** exposed field indicates when the application is to be started. The application is given control of the rendering window defined by the size field.

The **stopTime** exposedField indicates that the application is finished and should be shut down. The rendering window defined by the size field is returned to the MPEG-4 player.

The **isActive** exposedField signals the application to relinquish its rendering window to the MPEG-4 player, but to continue to run.

The **description** exposedField allows a prompt to be displayed as an alternative to the url in the **url** field. This choice should be user selectable.

The **parameter** exposedField carries parameters to be interpreted by the application decoder when the application window is instantiated.

The **url** exposedField carries the location of the windowed application.

The **size** exposedField provides the dimension (width and height) of the application window.

7.2.2.8 AudioBuffer

7.2.2.8.1 Node interface

AudioBuffer {

exposedField	SFBool	**loop**	FALSE
exposedField	SFFloat	**pitch**	1.0
exposedField	SFTime	**startTime**	0
exposedField	SFTime	**stopTime**	0
exposedField	MFNode	**children**	[]
exposedField	SFInt	**numChan**	1
exposedField	MFInt	**phaseGroup**	[1]
exposedField	SFFloat	**length**	0.0
eventOut	SFTime	**duration_changed**	
eventOut	SFBool	**isActive**	

}

NOTE — For the binary encoding of this node see node coding tables in electronic attachment.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

7.2.2.8.2 Functionality and semantics

The **AudioBuffer** node provides an interface to short snippets of sound to be used in an interactive scene.

EXAMPLE — Sounds triggered as “auditory icons” upon mouse clicks.

It buffers the audio generated by its children to support random restart capability upon interaction events. It differs from the **AudioClip** node in the following ways:

AudioBuffer can be used in broadcast and other one-way applications in which URLs from remote locations cannot be retrieved interactively

AudioBuffer can be used to trigger sounds made from processed sound (ie, with the other sound nodes) rather than only raw sound data as transmitted in the elementary stream

The **loop**, **startTime**, and **stopTime** exposedFields and the **isActive** eventOut, and their effects on the **AudioBuffer** node are described in 7.1.1.1.6.2.

The **length** field specifies the length in seconds of the audio buffer. Audio shall be buffered at the instantiation of the node, and whenever the **length** field changes.

The **pitch** field specifies a pitch-shift to apply to the output sound. The pitch-shift is calculated by simple resampling; that is, a pitch-shift of 2 corresponds to playing the sound twice as fast and an octave higher. If **pitch** is negative, the buffer is played backwards at the indicated speed, beginning at the last sample in the buffer and proceeding to the first, then returning to the last sample if **loop** is TRUE.

The **children** field specifies the child nodes that provide the sound for this node. Each child shall be an AudioBIFS node; that is, one of the following: **AudioSource**, **AudioDelay**, **AudioMix**, **AudioSwitch**, **AudioFX**, **AudioClip, AudioChannelConfig**, **AudioBuffer** or **AdvancedAudioBuffer**.

An event shall be generated via the **duration_changed** field whenever a change is made to the **startTime** or **stopTime** fields. An event shall also be triggered if these fields are changed simultaneously, even if the duration does not actually change.

The **numChan** field specifies the number of output channels of this node. If there are more output channels than input channels, the “extra” channels shall contain all 0s; if there are more input channels than output channels, the “extra” channels shall be ignored.

The **phaseGroup** field specifies phase relationships in the output of the node, see 7.1.1.2.13 and 7.2.2.15.

The output of this node is not calculated based on the current input values, but according to the **startTime** event, the **pitch** field and the contents of the clip buffer. When the **startTime** is reached (that is, the current scene time is greater than or equal to **startTime**), the sound output shall begin at the beginning of the clip buffer and **isActive** shall be set to TRUE. At each time step thereafter, the value of the output buffer shall be the value of the next portion of the clip buffer, upsampled or downsampled as necessary according to **pitch**. When the end of the clip buffer according to the value of **length** is reached, if **loop** is TRUE, the audio shall begin again from the beginning of the clip buffer; if **loop** is FALSE, the playback shall cease. This playback shall be continued until **stopTime** is reached. When the current scene time is greater than or equal to **stopTime**, the node shall cease to produce sound.

The clip buffer shall be calculated as follows. When the node is instantiated, or whenever the **length** field is changed, the first **length** seconds of the audio input to the **AudioBuffer** node shall be copied to the clip buffer. That is, after t seconds, where t < length, audio sample number t * S of channel i (where 0 <= i < **numChan**) in the buffer is set to contain the audio sample corresponding to time t of channel i of the input, where S is the sampling rate of this node. After the first length seconds, the input to this node has no effect. Changes to the **length** field that are received when **isActive** is TRUE shall be ignored.

When the playback is not active, the audio output of the node is all 0s.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

7.2.2.9 AudioChannelConfig

7.2.2.9.1 Node interface

AudioChannelConfig {

eventIn	MFNode	**addChildren**	
eventIn	MFNode	**removeChildren**	
exposedField	MFNode	**children**	[ ]
exposedField	SFInt32	**generalChannelFormat**	0
exposedField	SFInt32	**fixedPreset**	0
exposedField	SFInt32	**fixedPresetSubset**	0
exposedField	SFInt32	**fixedPresetAddInf**	0
exposedField	MFInt32	**channelCoordinateSystems**	[ ]
exposedField	MFFloat	**channelSoundLocation**	[ ]
exposedField	MFInt32	**channelDirectionalPattern**	[ ]
exposedField	MFVec3f	**channelDirection**	[ ]
exposedField	SFInt32	**ambResolution2D**	1
exposedField	SFInt32	**ambResolution3D**	0
exposedField	SFInt32	**ambEncodingConvention**	0
exposedField	SFFloat	**ambNfcReferenceDistance**	1.5
exposedField	SFFloat	**ambSoundSpeed**	340.0
exposedField	SFInt32	**ambArrangementRule**	0
exposedField	SFInt32	**ambRecombinationPreset**	0
exposedField	MFInt32	**ambComponentIndex**	[]
exposedField	MFFloat	**ambBackwardMatrix**	[]
exposedField	MFInt32	**ambSoundfieldResolution**	[]
field	SFInt32	**numChannel**	0

}

NOTE — For binary encoding of this node see node coding tables in electronic attachment

7.2.2.9.2 Functionality and semantics

This node is used to label the audio data in the audio subtree to supply the audio presenter with the required information for multichannel or soundfield signals. This is necessary in the following cases:

- channel configuration conflicts
  Some audio nodes have channel-variant behavior (**AudioMix**, **AudioSwitch** and **AudioFX**). For these nodes conflicts in the channel configuration can occur, e.g. if the **matrix** field in an **AudioMix** specifies an interchannel mixing.

- insufficient information
  A decoder can be used to transport subsets of multichannel formats that are not defined in the audioChannel config table (ISO/IEC 14496-3), e.g. the LS and RS channels of a 5.1 multichannel system.

- construction of soundfields
  E.g., a 5.1 soundfield can be transmitted with 3 two-channel audio decoders. This soundfield can be composed and labeled with the **AudioChannelConfig** node in the audio subtree.

- unsupported soundfield formats
  A decoder can be used to transport new or unsupported soundfields, for example an Ambisonics® soundfield. The soundfield formats are not defined in the audioChannel config table (ISO/IEC 14496-3). These soundfields can be composed and labeled with the AudioChannelConfig node in the audio subtree.

The node has the standard audio node interfaces, but no signal processing capability. The samples are passed through and get new channel configuration information.

A heterogenous audiochannel configuration cannot be described with this node – and does not make any sense. Therefore a **phaseGroup** does not exist in this node. Due to ambiguity between information provided in the **phaseGroup** field and the **AudioChannelConfig** node all nodes above the **AudioChannelConfig** node shall ignore the **phaseGroup** fields.

The **addChildren** eventIn specifies a list of nodes that shall be added to **children**.

The **removeChildren** eventIn specifies a list of nodes that shall be removed from **children**.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

The **children** array specifies the nodes affected by the **AudioChannelConfig**. Each child shall be one of the following nodes: **AudioSource**, **AudioDelay**, **AudioMix**, **AudioSwitch**, **AudioFX**, **AudioClip** or **AudioBuffer.**

The **numChan** field specifies the number of channels of audio output by this node.

The generalChannelFormat determines the general format of the channel label as described in Table 5**.**

Table 5 — generalChannelFormat table

Value	Configuration	Description	Valid fields
0	pass through	no labeling	none
1	ChannelPreset	Select the fixed values from the Channel Configuration Table in ISO/IEC 14496-3	**fixedPreset** **fixedPresetAddInf**
2	ChannelPreset-Subset	Select a subset from the Channel Configuration Table in ISO/IEC 14496-3 according to the mask in the **fixedPresetSubset** field	**fixedPreset** **fixedPresetSubset** **fixedPresetAddInf**
3	ParametricChan-nelOriented	parametric channel oriented configuration	**channelCoordinateSystems** **channelSoundLocation** **channelDirectionalPattern** **channelDirection**
4	ParametricAmbi-sonicsOriented	parametric Ambisonics® configuration	**ambResolution2D** **ambResolution3D** **ambEncodingConvention** **ambNfcReferenceDistance** **ambSoundSpeed** **ambArrangementRule** **ambRecombinationPreset** **ambComponentIndex** **ambBackwardMatrix** **ambSoundfieldResolution**
5..7	reserved for ISO use		

If generalChannelFormat is set to 1 (ChannelPreset) a fixed preset from the Channel Configuration in ISO/IEC 14496-3 is selected by the value of **fixedPreset**. Additional channel preset information can be specified from Table 6 by the value of **fixedPresetAddInf**. The order of the channels in AudioBIFS shall be adopted from Table 1.11 in ISO/IEC 14496-3, Subpart 1. I.e., the order starts with the center speaker (if present), continues with the stereo pairs, each beginning with the left followed by the right channel and ends with a back center speaker (if present) and the LFE channel (if present).

If **generalChannelFormat** is set to 2 (ChannelPresetSubset), a fixed preset of Channel Configuration in ISO/IEC 14496-3 is selected by the value of **fixedPreset.** A subset of the selected fixed preset is masked by **fixedPresetSubset**. Each bit of the value in this field represents an input channel according to the channel to speaker mapping column of the Channel Configuration Table in ISO/IEC 14496-3 where the first channel is indicated with the LSB of the **fixedPresetSubset** field.

EXAMPLE — fixedPreset=6 (5+1 multichannel) is the channel to speaker mapping C, L, R, LS, RS, LFE. This preset can be used to describe a 2-channel stream containing the LS and RS channels by masking the 4th and the 5th element of the fixed preset with **fixedPresetSubset** set to [MSB…LSB] […0 0 1 1 0 0 0] = 24_{10}

Additional channel preset information can be specified from Table 6 by the value of **fixedPresetAddInf.**

The **fixedPresetAddInf** can set additional properties of the channels labeled with **fixedPreset** and **fixedPresetSubset.** Values for these additional labels can be found in Table 6:

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Table 6 — fixedPresetAddInf table

Value	Configuration
0	none
1	Binaural
2	Transaural
3..7	reserved for ISO use

If **generalChannelFormat** is set to 3 (ParametricChannelOriented), **channelCoordinateSystems, channelSoundLocation**, **channelDirectionalPattern** and **channelDirection** for a generic format are valid. For its functional description see 7.2.2.9.2.1.

If **generalChannelFormat** is set to 4 (ParametricAmbisonicsOriented), **ambResolution2D, ambResolution3D, ambEncodingConvention, ambNfcReferenceDistance, ambSoundSpeed, ambArrangementRule, ambComponentIndex, ambBackwardMatrix,** and **ambSoundfieldResolution** are valid. For its functional description see 7.2.2.9.2.2.

7.2.2.9.2.1 Generic channel configuration parameters

The **channelCoordinateSystems** field determines the coordinate system for the corresponding channels. Each channel can have its own coordinate system as described in Table 7. Measurement units are meter for the cartesian coordinates and radians for the angle coordinates. In polar coordinates or cylindric coordinates a radius with the value r=0 indicates an unused component. An example is the 5.1 format where no radius is specified.

Table 7 — channelCoordinateSystems table

Value	Coordinate system	ChannelSoundLocation format
0 (default)	Cartesian origin = scene origin (locations relative to the scene)	{x, y, z}
1	Cartesian origin = user position (locations relative to the user)	{x, y, z}
3	Polar origin = scene origin (locations relative to the scene)	{r, azimuth, elevation}
4	Polar origin = user position (locations relative to the user)	{r, azimuth, elevation}
5	Cylindric origin = scene origin (locations relative to the scene)	{r, azimuth, z}
6	Cylindric origin = user position (locations relative to the user)	{r, azimuth, z}
7	reserved for ISO use	

channelSoundLocation determines the location of the sound of the corresponding channel. Each location is a 3-element vector. This field is in MFFloat-format instead of MFVec3f due to its multiple format possibility. Each vector consists of three elements.

For example in a 5-channel configuration the second channel can be configured as follows: **channelCoordinateSystems[1]** == 4 ➔ **channelSoundLocation[3:5]** is a 3D-vector in polar coordinates relative to the listener describing the position of the sound of the second channel.

The **channelDirectionalPattern** is an integer vector restricted to the values from Table 8. Each element describes the desired directional pattern of the sound of the corresponding channel.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Table 8 — channelDirectionalPattern association table

Value	Polarity
0	Sphere 0th order (monopole)
1	Sphere 1st order (dipole)
2	Cardioide
3..7	reserved for ISO use

This is useful for example in case of Dolby ProLogic where the front channels have monopole patterns and the surround channel has to have dipole characteristics (**channelDirectionalPattern[0:3]** = [0 0 0 1] for the four channels L, R, C, S).

The **channelDirection** field is valid for values of the **channelDirectionalPattern** > 0. It describes the direction of the **channelDirectionalPattern** in the coordinate system **channelCoordinateSystems** (see Table 7) defined for the corresponding channel.

7.2.2.9.2.2 Ambisonics® channel configuration parameters

These parameters are used to interpret the audio channels as "Ambisonics® components" B_{mn}^{σ}, which represent the spatial encoding/recording of the sound field according to the spherical harmonics Y_{mn}^{σ} (see Figure 11).

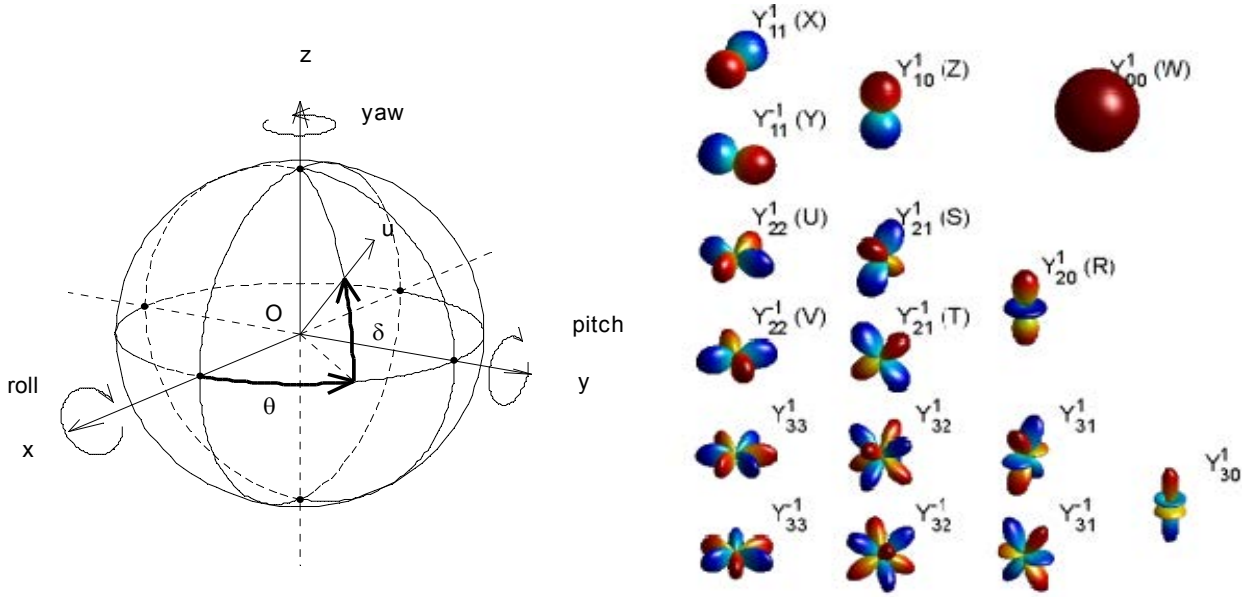

Figure 11 — Left part: spherical coordinate system (radius *r*, azimuth θ, elevation δ), X-axis being considered to be forward pointing while Y-axis is left-hand pointing. Right part: 3D view of spherical harmonics Y_{mn}^{σ} with usual designation of associated Ambisonics® components. 0th and 1st order components W, X, Y and Z compose the so-called "B-format"

Each component B_{mn}^{σ} is re-paired in terms of indexes m, n (with $0 \le n \le m$) and $\sigma = \pm 1$. Components with $\sigma = +1$ have an azimuth dependence in cos nθ while components with $\sigma = -1$ have an azimuth dependence in sin nθ. In case of increasing m and n the newly considered components provide a more accurate angular resolution to the 3D sound scene description. Note that for each m there's only one component with n=0 which has no azimuth dependence. Components with n=m are "horizontal" or "2D" components, and are generally privileged over other components since in many cases rendering may be performed on horizontal-only loudspeaker configurations. For convenience Ambisonics® components will be referred to by a single index SID (Table 9) derived by enumerating indexes (m,n,σ) according to this rule: increase m from 0; for each m, decrease n from m to 0; for each n, choose $\sigma = +1$, then $\sigma = -1$ (except if n=0).

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

Table 9 — Single Index Designation (SID) for Ambisonics® Components

Usual Name	W	X	Y	Z	U	V	No more usual names							
(m,n,σ) designation	0,0 +1	1,1 +1	1,1 -1	1,0 +1	2,2 +1	2,2 -1	…	m,m +1	…	m,n +1	m,n □1	…	m,0 +1	…
Single Index Designation	0	1	2	3	4	5	…	m^2	…	m^2 +2(m-n)	m^2 +2(m−n)+1	…	$(m+1)^2$ -1	…

Further explanation of concepts and parameters of this part can be found in [1]. To be complete, and in order to let no ambiguity (related to the presence/absence of the Condon-Shortley phase), the Legendre polynomials and associated Legendre functions used are also specified as follows.

Associated Legendre functions:

$$P_{mn} = (1 - x^2)^{\frac{n}{2}} \frac{d^n}{dx^n} P_m(x), \quad 0 < n < m$$

With Legendre polynomials defined as:

$$P_m(x) = \frac{1}{2^m m!} \frac{d^m}{dx^m} (x^2 - 1)^m$$

ambResolution2D: Upper order m of 2D (horizontal only) components (*i.e.* such that m=n).

ambResolution3D: Upper order m of 3D (non-horizontal) components (*i.e.* such that n<m); will be generally less than or equal to **ambResolution2D**.

ambEncodingConvention: The encoding convention characterized by the set of weighting factors applied to the components or spherical harmonics. Possible values are: 'N2D'=0, 'SN2D'=1, 'N3D'=2, 'SN3D'=3, 'MaxN'=4, 'FMH'=5, etc. Conversion formulae are described in details in [1]. Note that the specified convention applies to the Ambisonics® components finally obtained, for example after backward matrixing (or channel recombination) in case of occurrence (see comments on **ambBackwardMatrix**).

The intrinsic parameter for Near-Field Compensated Ambisonics® sound fields is the ratio of a reference distance (=**ambNfcReferenceDistance)** and the sound speed (=**ambSoundSpeed)** as measured in the real world. The reference distance refers to the loudspeaker array radius for which the sound field rendering is preferably dedicated to (in the sense that spatial decoding requires no sound field curvature adaptation).

ambArrangementRule: a value that describes how to interpret the conveyed channels as ambisonic components. Its binary coding can be interpreted as a "flag container". Bit number 0 is interpreted as "explicit ordering" flag: if binary masking of **ambArrangementRule** by 001 yields 001 (flag is true), then **ambComponentIndex** field is valid. Bits number 1 and number 2 are interpreted as respectively "recombination" flag and "explicit recombination" flag: if binary masking by 110 yields 010, **ambRecombinationPreset** field is valid and predefined recombination rules apply; if it yields 110, **ambBackwardMatrix** field is valid and explicit channel recombination applies.

ambRecombinationPreset: describes predefined rules (listed in Table 10) of channel recombination. Channel recombination operates the same way as described with **ambBackwardMatrix** below, but using parameters N and K of Table 10 and processing the first N channels with a predefined conversion rule instead of an explicit matrix.

Table 10 — Particular cases of channels recombination. "Value" is the 3-bit right-shifted value of the ambArrangementRule field. UHJ Format is described in [ref_UHJ]

Value	Special rules for channels recombination	N	K
0	2-channel UHJ: First 2 channels conform to UHJ format	2	2
1	3-channel UHJ: First 3 channels conform to UHJ format	3	3
2	4-channel UHJ: First 4 channels conform to UHJ format	4	4
3..15	reserved for ISO use		

ambComponentIndex: A vector containing the single indexes (as defined in Table 9) that identify Ambisonics® components, as resulting from the channel recombination described below if applied, or as being directly represented by the conveyed channels otherwise.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

ambBackwardMatrix: A vector describing the recombination matrix. First 2 elements are the number of lines (K) and the number of columns (N). The other elements are the line-wise listed matrix coefficients $a_{i,j}$ ($1 \le i \le K$, $1 \le j \le N$):

$$\text{ambBackwardMatrix} = \left[K\ N\ a_{1,1} \ldots a_{1,j} \ldots a_{1,N}\ \ldots a_{i,1} \ldots a_{i,j} \ldots a_{i,N}\ \ldots a_{K,1} \ldots a_{K,j} \ldots a_{K,N}\right]$$

If not null the length of **ambBackwardMatrix** shall be 2+N*K. The presenter may apply this matrix to the first N elementary channels {C_1 … C_N} in order to extract K Ambisonic components {B_1 … B_K}: $B_i = \sum_{j=1}^{N} a_{i,j} C_j$, which can also be written as:

$$\begin{bmatrix} B_1 \\ \vdots \\ B_i \\ \vdots \\ B_K \end{bmatrix} = \begin{bmatrix} a_{1,1} & \ldots & a_{1,j} & \ldots & a_{1,N} \\ \vdots & \ddots & \vdots & & \vdots \\ a_{i,1} & \ldots & a_{i,j} & \ldots & a_{i,N} \\ \vdots & & \vdots & \ddots & \vdots \\ a_{K,1} & \ldots & a_{K,j} & \ldots & a_{K,N} \end{bmatrix} \begin{bmatrix} C_1 \\ \vdots \\ C_j \\ \vdots \\ C_N \end{bmatrix}$$

Then the list of restored Ambisonic components is completed by the remaining (numChannel-N) conveyed channels {C_{N+1}, … $C_{numChannel}$} that represent "unmatrixed" Ambisonic components {B_{K+1}, … $B_{numChannel-N+K}$}:

$$\begin{bmatrix} B_{K+1} \\ \vdots \\ B_{i'} \\ \vdots \\ B_{\text{numChannel}-N+K} \end{bmatrix} = \begin{bmatrix} C_{N+1} \\ \vdots \\ C_{i'+N-K} \\ \vdots \\ C_{\text{numChannel}} \end{bmatrix}$$

This way, the total number of Ambisonic components is numChannel-N+K.

Remind that **ambComponentIndex**, **ambSoundfieldResolution** and **ambEncodingConvention** fields actually apply to this resulting set of Ambisonic components {B_1 … $B_{numChannel-N+K}$} rather than the conveyed channels {C_1, ... $C_{numChannel}$}.

ambSoundfieldResolution: A vector containing information on the global sound field resolution associated to each component. Indeed when mixing Ambisonic sound fields of different resolutions (as defined by their respective upper order) the content creator may have to provide several versions of the lower order component(s), in order to make possible some optimization of the spatial decoding at the rendering stage. For example: when mixing 1st and 2nd order sound fields, provide two W channels (SID=0), the one being the result of mixing both sound fields, the second being the one of either the 1st or the 2nd order sound field. The **ambSoundfieldResolution** should have the same number of elements as **ambComponentIndex** and requires **ambArrangementRule**=1 since several components might have the same “Single Index Designation”. Each element of **ambSoundfieldResolution**, when positive, refers to the upper sound field resolution the indexed channel belongs to (that means that it's the result of mixing sound fields having spatial resolutions up to the considered value). In the case of negative values each associated absolute value refers to a lower resolution instead of an upper resolution.

7.2.2.10 AudioClip

7.2.2.10.1 Node interface

AudioClip {

exposedField	SFString	**description**	""
exposedField	SFBool	**loop**	FALSE
exposedField	SFFloat	**pitch**	1.0
exposedField	SFTime	**startTime**	0
exposedField	SFTime	**stopTime**	0
exposedField	MFString	**url**	[]
eventOut	SFTime	**duration_changed**	
eventOut	SFBool	**isActive**	

}

NOTE — For the binary encoding of this node see node coding tables in electronic attachment.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015

7.2.2.10.2 Functionality and semantics

The semantics of the **Audioclip** node are specified in ISO/IEC 14772-1:1998, subclause 6.4.

The **loop**, **startTime**, and **stopTime** exposedFields and the **isActive** eventOut, and their effects on the **AudioClip** node are described in 7.1.1.1.6.2.

The **url** field specifies the data source to be used (see 7.1.1.2.7.1).

The use of this node should be avoided in new applications. It should only be used for VRML backward compatible applications.

7.2.2.11 AudioDelay

7.2.2.11.1 Node interface

```
AudioDelay {
    eventIn       MFNode    addChildren
    eventIn       MFNode    removeChildren
    exposedField  MFNode    children        []
    exposedField  SFTime    delay           0
    field         SFInt32   numChan         1
    field         MFInt32   phaseGroup      []
}
```

NOTE — For the binary encoding of this node see node coding tables in electronic attachment.

7.2.2.11.2 Functionality and semantics

The **AudioDelay** node allows sounds to be started and stopped under temporal control. The start time and stop time of the child sounds are delayed or advanced accordingly.

The **addChildren** eventIn specifies a list of nodes that shall be added to the **children** field.

The **removeChildren** eventIn specifies a list of nodes that shall be removed from the **children** field.

The **children** array specifies the nodes affected by the delay. Each child shall be an AudioBIFS node; that is, one of the following: **AudioSource**, **AudioDelay**, **AudioMix**, **AudioSwitch**, **AudioFX**, **AudioClip, AudioChannelConfig, AudioBuffer** or **AdvancedAudioBuffer**.

The **delay** field specifies the delay to apply to each child node.

The **numChan** field specifies the number of channels of audio output by this node.

The **phaseGroup** field specifies the phase relationships among the various output channels; see 7.1.1.1.6.2.

Implementation of the **AudioDelay** node requires the use of a buffer of size $d * S * n$, where d is the length of the delay in seconds, S is the sampling rate of the node, and n is the number of output channels from this node. At scene startup, a multichannel delay line of length d and width n is initialized to reside in this buffer.

At each time step, the $k * S$ audio samples in each channel of the input buffer, where k is the length of the system time step in seconds, are inserted into this delay line. If the number of input channels is strictly greater than the number of output channels, the extra input channels are ignored; if the number of input channels is strictly less than the number of output channels, the extra channels of the delay line shall be taken as all 0's.

The output buffer of the node is the $k * S$ audio samples which fall off the end of the delay line in this process. Note that this definition holds regardless of the relationship between k and d.

If the **delay** field is updated during playback, discontinuities (audible artefacts or "clicks") in the output sound may result. If the **delay** field is updated to a greater value than the current value, the delay line is immediately extended to the new length, and zero values inserted at the beginning, so that $d * S$ seconds later there will be a short gap in the output of the node. If the **delay** field is updated to a lesser value than the current value, the delay line is immediately shortened to the new length, truncating the values at the end of of the line, so that there is an immediate discontinuity in sound output. Manipulation of the **delay** field in this manner is not recommended unless the audio is muted within the terminal or by appropriate use of an **AudioMix** node at the same time, since it gives rise to impaired sound quality.

7.2.2.12 AudioFX

7.2.2.12.1 Node interface

```
AudioFX {
```


© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 14496-11:2015