
**Information technology — Coding of
audio-visual objects —**

Part 11:

Scene description and application engine

AMENDMENT 7: ExtendedCore2D profile

Technologies de l'information — Codage des objets audiovisuels —

Partie 11: Description de scène et moteur d'application

AMENDEMENT 7: Profil ExtendedCore2D



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.



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2010

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

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.

Amendment 7 to ISO/IEC 14496-11:2005 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 11:

Scene description and application engine

AMENDMENT 7: ExtendedCore2D profile

In 7.2, add the following subclauses in node alphabetical order:

7.2.2.ZZ CacheTexture

7.2.2.ZZ.1 Node interface

CacheTexture {			
Field	SFInt32	objectTypeIndication	0
Field	SFString	decoderSpecificInfo	NULL
Field	SFString	image	NULL
Field	SFString	cacheURL	NULL
Field	MFURL	cacheOD	[]
Field	SFInt32	expirationDate	0
Field	SFBool	repeatS	TRUE
Field	SFBool	repeatT	TRUE
}			

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

7.2.2.ZZ.2 Functionality and semantics

The CacheTexture allows carriage of visual data embedded inside the BIFS stream rather than using the OD framework. The **objectTypeIndication** field identifies the media type of the visual data. The compressed data is carried in the **image** field, as a single access unit. If a decoder configuration is needed, it may be carried in the **decoderSpecificInfo** field. The node can be used as a texture object in an **Appearance** node. The node can also be used in as a child node of a 2D or 3D container when it is only used for image caching. Additionally, the CacheTexture node allows for caching the embedded image by specifying a **cacheURL** name to be referred to by other nodes in the scene, as well as an **expirationDate** indicating the time in seconds the terminal should keep the data in its cache. If **expirationDate** is 0, the data shall not be cached. If **expirationDate** is strictly negative, the data should be cached for as long as possible. In any case, whether the data is cached or not is implementation specific.

The **cacheOD** field identifies an existing OD in the scene to be cached with the given **cacheName** and **expirationDate**. If **cacheOD** is set, **image**, **decoderSpecificInfo** and **objectTypeIndication** shall be ignored. Results are undefined if the OD indicated by the **cacheOD** is not a still image object such as JPEG or PNG.

The scoping of the CacheTexture node shall be done at the service level (same broadcast channel or same service URL of the initial scene). Sub-scenes opened through inline nodes are part of the same caching scope as the parent scene.

Example of cache usage

```

Shape {
  appearance Appearance {
    texture ImageTexture {
      url "some_cache_url_name"
    }
  }
}
...
CacheTexture {
  objectTypeIndication 0x6D
  image ...
  cacheURL "some_cache_url_name"
  expirationDate 3600 //one hour caching
}
....

```

7.2.2.ZZ EnvironmentTest

7.2.2.ZZ.1 Node interface

EnvironmentTest {			
eventIn	SFBool	evaluate	
exposedField	SFBool	enabled	TRUE
exposedField	SFInt32	parameter	0
exposedField	SFString	compareValue	NULL
exposedField	SFBool	evaluateOnChange	TRUE
eventOut	SFBool	valueLarger	
eventOut	SFBool	valueEqual	
eventOut	SFBool	valueSmaller	
eventOut	SFString	parameterValue	
}			

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

7.2.2.ZZ.2 Functionality and semantics

The EnvironmentTest node enables testing a **parameter** of the terminal environment, possibly comparing their values with the **compareValue**. The evaluation of the parameter triggers different eventOuts depending on the type of the parameter:

- If the **parameter** type is Boolean, the evaluation triggers a **valueEqual** eventOut, and the **compareValue** field is ignored.
- If the **parameter** type is a number and the **compareValue** represents a number, the two values are compared and the following eventOuts are generated:
 - **valueEqual** if **parameter** and **compareValue** are equal
 - **valueLarger** if **compareValue** is strictly larger than **parameter**
 - **valueSmaller** if **compareValue** is strictly less than **parameter**

The supported parameter types are defined in Table AMD7.1.

In any case, the **parameterValue** eventOut is triggered after evaluation.

If **evaluateOnChange** is set to FALSE, the node only evaluates upon receiving the **evaluate** eventIn; otherwise, the node evaluates on any change of **parameter** or **compareValue**.

The node evaluates and triggers events only when its **enabled** field is true.

Table AMD7.1 — Environmental parameters

Value	Definition	Type
0	Display region Aspect Ratio (larger dimension divided by smaller dimension, regardless of screen orientation)	Float
1	Portrait mode of the display region (TRUE if width<height)	Boolean
2	Display region width in pixels	Integer
3	Display region height in pixels	Integer
4	Horizontal DPI	Integer
5	Vertical DPI	Integer
6	Automotive Situation (terminal user drives a moving vehicle)	Boolean
7	User is Visually Challenged	Boolean
8	Touch Screen present on terminal	Boolean
9	Navigation Keypad present on terminal	Boolean
0x00000007-0xEEFFFFFF	ISO Reserved	
0xF0000000-0xFFFFFFFF	User Reserved	

The display region is the area onto which the BIFS content is rendered. This region may be the entire screen, some part of the screen or an off-screen memory region.

7.2.2.ZZ KeyNavigator

7.2.2.ZZ.1 Node interface

```

KeyNavigator {
    eventIn          SFBool          setFocus
    exposedField     SFNode          sensor          NULL
    exposedField     SFNode          left             NULL
    exposedField     SFNode          right            NULL
    exposedField     SFNode          up               NULL
    exposedField     SFNode          down             NULL
    exposedField     SFNode          select           NULL
    exposedField     SFNode          quit            NULL
    exposedField     SFFloat         step             0
    eventOut         SFBool          focusSet
}

```

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

7.2.2.ZZ.2 Functionality and semantics

The KeyNavigator node enables simple, pre-defined 2D navigation in the scene. Each KeyNavigator is associated with an existing sensor node (TouchSensor, PlaneSensor2D...) through the **sensor** field. The first KeyNavigator node found in the scene is used to determine the initial focusable object. If the attached **sensor** node is NULL or is disabled, the focus is not attached to any visual part of the scene. Focus can be changed by using the navigation pad of the terminal as follows:

- Pressing the left key will move focus to the **left** KeyNavigator node
- Pressing the right key will move focus to the **right** KeyNavigator node
- Pressing the up key will move focus to the **up** KeyNavigator node
- Pressing the down key will move focus to the **down** KeyNavigator node
- Pressing the validation key (OK, Enter, Select...) will move focus to the **select** KeyNavigator node