FINAL
DRAFT

AMENDMENT

ISO/IEC 14496-4:2000 FDAM 1

ISO/IEC JTC **1**

Secretariat: **ANSI**

Voting begins on:
2002-01-10

Voting terminates on:
2002-03-10

Information technology — Coding of audio-visual objects —

Part 4: Conformance testing

AMENDMENT 1: Conformance testing extensions

Technologies de l'information — Codage des objets audiovisuels —

Partie 4: Essai de conformité

AMENDEMENT 1: Extensions pour l'essai de confirmité

Please see the administrative notes on page iii

Reference number
ISO/IEC 14496-4:2000/FDAM 1:2002(E)

In accordance with the provisions of Council Resolution 21/1986, this document is **circulated in the English language only**.

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 3.

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 Amendment may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to International Standard ISO/IEC 14496-4:2000 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Introduction

Parts 1, 2 and 3 of ISO/IEC 14496 specify a multiplex structure and coded representations of audio-visual information. Parts 1, 2 and 3 of ISO/IEC 14496 allow for large flexibility, achieving suitability of ISO/IEC 14496 for many different applications. The flexibility is obtained by including parameters in the bitstream that define the characteristics of coded bitstreams. Examples are the audio sampling frequency, picture size, picture shape, picture rate and bitrate parameters. Part 6 of ISO/IEC 14496 specifies a framework for uniform delivery of MPEG-4 content irrespecitve of its location and the transport technolgy.

This part of ISO/IEC 14496 specifies how tests can be designed to verify whether bitstreams and decoders meet the requirements as specified in parts 1, 2 and 3 of ISO/IEC 14496 and allow interoperability with remote terminals in interactive sessions as well as access to boadcast or stored. These tests can be used for various purposes such as:

- manufacturers of encoders, and their customers, can use the tests to verify whether the encoder produces valid bitstreams.

- manufacturers of decoders and their customers can use the tests to verify whether the decoder meets the requirements specified in parts 1,2 and 3 of ISO/IEC 14496 for the claimed decoder capabilities.

- manufacturers of terminals that wish to interact with other remote terminals in interactive sessions over a multitude of transport networks as well as with broadcast and storage delivery technologies.

Information technology — Coding of audio-visual objects —

Part 4:
Conformance testing

Amendment 1: Conformance testing extensions

Replace and add the following subclauses to Clause 3.

3.4.3 Test Suites

This paragraph describes the test suites to be used. A test suite is a suite of material and measurement algorithms and associated reference algorithms.

3.4.3.1 BIFS Feature List

The test suite shall verify the features in Table 3-1. For nodes, the following shall be tested:

- Presence in the scene tree after decoding.
- Appropriate value of the fields after decoding.
- Functionality that has an effect on the scene tree, e.g. for a ROUTE, if the source field value changes, the target field value shall change accordingly.

Table 3-1 — BIFS Test Suite Information

N°	Feature	Reference of Test sequence and associated method
	BIFS-Anim: position 3D animation	
	BIFS-Anim: position 2D animation	Anim-simple
	BIFS-Anim: color animation	
	BIFS-Anim: angle animation	Anim-circle
	BIFS-Anim: float animation	
	BIFS-Anim: bound float animation	
	BIFS-Anim: normal animation	
	BIFS-Anim: size 3D animation	
	BIFS-Anim: size 2D animation	
	BIFS-Anim: integer animation	
	BIFS-Anim: several fields in the same node	Anim-rect,
	BIFS-Anim: several nodes	
	BIFS-Anim: switch of a node (isActive mask)	
	BIFS-Anim: random access true	
	BIFS-Anim: random access false	
	Quantization: 3D position	Pos3d-4bit,
	Quantization: 2D position	
	Quantization: drawing order	
	Quantization: color	
	Quantization: texture coordinate	
	Quantization: angle	Angle-8bit,
	Quantization: scale	
	Quantization: interpolator keys	
	Quantization: normals	Normal-4bit,
	Quantization: rotations	
	Quantization: object size 3D	
	Quantization: object size 2D	
	Quantization: linear scalar quantization	
	Quantization: efficient float	
	Quantization: node default values	
	Quantization: isLocal mode	
	Quantization: DEF/USE	

	BIFS Command: insert node index	allupdates, update2
	BIFS Command: insert node begin	allupdates, update2
	BIFS Command: insert node end	Updatetest, Friday, allupdates
	BIFS Command: insert Idx value index	allupdates, update2
	BIFS Command: insert Idx value begin	allupdates, update2
	BIFS Command: insert Idx value end	allupdates, update2
	BIFS Command: insert ROUTE	jerusalem, slides2, update2
	BIFS Command: delete node	Bifs-deletenode, update2
	BIFS Command: delete Idx value index	Friday, allupdates, update2
	BIFS Command: delete Idx value begin	allupdates, update2
	BIFS Command: delete Idx value end	allupdates, update2
	BIFS Command: replace node	update2,
	BIFS Command: replace field	Bifs-2dfieldreplace1, Friday, allupdates, update2
	BIFS Command: replace Idx value index	Pae_raise, allupdates, update2
	BIFS Command: replace Idx value begin	allupdates, update2
	BIFS Command: replace Idx value end	allupdates, update2
	BIFS Command: replace ROUTE	update2,
	BIFS Command: replace scene	Ecran2, Updatetest, update2
	BIFS Command: several commands in same AU	Updatetest,
	BIFS Scene: mask node	
	BIFS Scene: list node	Jerusalem, Layout, Testlayout
	BIFS Scene: mask MFField	
	BIFS Scene: list MFField	
	BIFS Scene: ROUTE	Scaling3D, Jerusalem, Ecran2
	SFBool	Ecran2, Updatetest
	SFColor	Ecran2, Updatetest
	SFFloat	Ecran2, Updatetest
	SFInt32	Ecran2, Updatetest
	SFRotation	Normal-4bit,
	SFString	Ecran2, Updatetest
	SFTime	Jerusalem, OrientInterp3D
	SFUrl	Anchor, Audiotest
	SFVec2f	Ecran2, Updatetest
	SFVec3f	Bifs-deletenode,
	SFImage	
	SFCommandBuffer	Ecran2, Slider, Paeraise
	SFScript	Scaling3D, SFColor01, Value_changed3d, Qtvr
	BIFSConfig: BIFS Anim	Anim-rect, Anim-circle, Anim-simple
	BIFSConfig: BIFS Command	Ecran2, Jerusalem
	AcousticMaterial	AABphy65-80
	AcousticScene	AABphy55-64
	Anchor	Anchor, Frame1
	AnimationStream	Anim-rect, Anim-circle, Anim-simple
	Appearance	Bifs-deletenode, Bifs-2dfieldreplace1
	AudioBuffer	
	AudioClip	
	AudioDelay	
	AudioFX	
	AudioMix	
	AudioSource	Audiotest, Ifs
	AudioSwitch	
	Background	
	Background2D	
	Billboard	
	Bitmap	Ecran2, Jerusalem, Updatetest, Transition
	Box	Bifs-deletenode,
	Circle	Bifs-2dfieldreplace1, Ecran2, Simple
	Collision	
	Color	
	ColorInterpolator	Timestest, Anibut3

	CompositeTexture2D	Layout,
	CompositeTexture3D	
	Conditional	Ecran2, Layout, Friday
	Cone	Bifs-deletenode,
	Coordinate	
	Coordinate2D	Layout, Updatetest
	CoordinateInterpolator	
	CoordinateInterpolator2D	
	Curve2D	Layout, Polygontest
	Cylinder	Bifs-deletenode,
	CylinderSensor	
	DirectionalLight	PointLightPrimitive1-3D
	DirectiveSound	AABphy1-56 (physical approach) AABper1-46 (perceptual approach)
	DiscSensor	
	ElevationGrid	
	Expression	
	Extrusion	
	Face	
	FaceDefMesh	
	FaceDefTables	
	FaceDefTransform	
	FAP	
	FDP	
	FIT	
	Fog	Scaling3D,
	FontStyle	Scaling3D, Ecran2
	Form	Form_spread, Form_spread2, Testform
	Group	Anchor, Ecran2, Layout
	ImageTexture	Ecran2, Jerusalem, Pae_raise
	IndexedFaceSet	BB, Biliard
	IndexedFaceSet2D	Ifs, Facesetgalore
	IndexedLineSet	
	IndexedLineSet2D	Polygontest, Updatetest, Mosaic18
	Inline	
	LOD	
	Layer2D	Bifs-2dfieldreplace1, Transit
	Layer3D	Bifs-deletenode, Scaling3D
	Layout	Jerusalem, Layout, Testlayout
	LineProperties	Ecran2, Updatetest
	ListeningPoint	
	Material	Bifs-deletenode, Material3D
	Material2D	Bifs-2dfieldreplace1, Ecran2
	MovieTexture	Jerusalem, Friday, Av
	NavigationInfo	
	Normal	
	NormalInterpolator	
	OrderedGroup	Form_spread2, Pae_raise
	OrientationInterpolator	OrientInterp3D,
	PerceptualParameters	AABper47-76 (perceptual approach)
	PixelTexture	
	PlaneSensor	
	PlaneSensor2D	Slider, Valuator
	PointLight	PointLightPrimitive1-3D,
	PointSet	
	PointSet2D	
	PositionInterpolator	Value_changed3d,
	PositionInterpolator2D	Friday, Traj
	ProximitySensor2D	
	ProximitySensor	

	QuantizationParameter	
	Rectangle	Ecran2, Updatetest, Friday
	ScalarInterpolator	Trans-group, Mosaic41
	Script	Scaling3D, SFColor01, Value_changed3d, Qtvr
	Shape	Bifs-deletenode, Ecran2, Jerusalem
	Sound	
	Sound2D	Audiotest, Av
	Sphere	Bifs-InsertNodeStress
	SphereSensor	
	SpotLight	PointLightPrimitive1-3D
	Switch	Ecran2, Jerusalem, Friday
	TermCap	
	Text	Ecran2, Jerusalem, Updatetest
	TextureCoordinate	
	TextureTransform	
	TimeSensor	OrientInterp3D, Jerusalem, Trans-group, Timestest
	TouchSensor	Scaling3D, Ecran2, Jerusalem, Friday
	Transform	Bifs-deletenode,
	Transform2D	Bifs-2dfieldreplace1, Ecran2
	Valuator	Slider, Valuator
	Viewpoint	Scaling3D,
	VisibilitySensor	
	Viseme	
	WorldInfo	
	DEF / USE	SFColor01, Ecran2, Jerusalem
	BIFSConfig (DecoderSpecificInfo for BIFS)	Ecran2, Jerusalem

3.4.3.2 OD Feature List

Table 3-2 — OD Test Suite Information

N°	Feature	Reference of Test sequence and associated method
1.	IOD	Anchor, Audiotest, Ecran2
2.	OD Update (new)	Ecran2, Jerusalem
3.	OD Remove	
4.	ES Update (new)	
5.	ES Remove	
6.	IPMP Update	
7.	IPMP Remove	
8.	OD Update (modification)	
9.	ES Update (modification)	
10.	OCI descriptors	
11.	IPI descriptors	
12.	QoS descriptors	
13.	Extension descriptors	Ecran2, Slider

3.4.3.3 Bitstreams

Name	Provider	Content
Anchor	ENST	Anchor node
Audiotest	ENST	Audiosource and Sound2d
Ecran2	ENST	Medium size sample
Form_spread	ENST	Form node

Form_spread2	ENST	Form node
Form_spread3	ENST	Form node
Updatetest	ENST	Updates
Transit	ENST	Layer2D as clipping etc…
Valuator	ENST	Valuator
Simple	ENST	Simple2D sample
Jerusalem	ENST	Medium size sample
Layout	ENST	Medium size sample
Pae_raise	ENST	OrderedGroup, updates and interactivity
Polygontest	ENST	Polygons and lines
Slider	ENST	Valuator…
Timestest	ENST	ColorInterpolator
Trans-group	ENST	ScalarInterpolator
Testlayout	ENST	Layout
Testform	ENST	Form
Qtvr	ENST	Script, Valuator, Arb. Shape video
Friday	ENST	Medium size example
Traj	ENST	PositionInterpolator2D
Ifs	ENST	IndexedFaceSet2D
Anim-simple	FT	Animation of Transform2D.scale
Anim-rect	FT	Animation of Transform2D.translation and rotation
Anim-circle	FT	Animation of Transform2D.rotationAngle
Bifs-deletenode	FT	Delete node on 3d nodes
Bifs-2dfieldreplace1	FT	Replace field on 2d nodes
Bifs-InsertNodeStress	FT	Insert node
PointLightPrimitive1-3D	FT	3d lights
OrientInterp3D	FT	Orientation Interpolator
Material3D	FT	Material3D
Angle-8bit	FT	Quantization of angle
Normal-4bit	FT	Quantization of normal
Pos3d-4bit	FT	Quantization of position 3d
Scaling3D	FT	Script
SFColor01	FT	Script
Value_changed3d	FT	Script
BB	Optibase	IndexedFaceSet
Biliard	Optibase	IndexedFaceSet
Anibut3	ENST	ColorInterpolator, OrderedGroup, ScalarInterpolator, TimeSensor
Av	ENST	Sound2D, MovieTexture
Frame1	ENST	Anchor, etc…
Imabut	ENST	Image button
Interleaved_2s	ENST	Interleaved MP4 file (onechunk is the non-interleaved version)
Kang	ENST	Video with shape (static texture and shape)
Forme	ENST	Video with shape (static shape, moving texture)
Oiseau	ENST	PlaneSensor2D, Video with shape (static texture, moving shape)
Mosaic18	ENST	Background2D, Anchor, etc…
Mosaic41	ENST	Background2D, ScalarInterpolator, PlaneSensor2D…
Slides2	ENST	SlideShow
Facesetgalore	ENST	IndexedFaceSet2D and lots of updates on it.
Allupdates	ENST	Lots of updates encapsulated in Conditionals tied with text buttons.
Interactive	ENST	Tests the interactive starting of media.
Meteo2	ENST	Image and text interactivity.
Paepopup	ENST	A kind of popup menu.
Ultrasimple	ENST	For the simple profile.
Update2	ENST & FT	A set of 17 sequences covering all types of updates

AABphys1-80	HUT	Tests Advanced AudioBIFS physical approach, nodes DirectiveSound, AcousticScene, AcousticMaterial
AABper1-76	FT	Tests Advanced AudioBIFS perceptual approach, nodes DirectiveSound, PerceptualParameters

3.5 Advanced BIFS

3.5.1 Bitstream conformance

3.5.1.1 Conformance Requirements

BIFS streams shall comply with the specifications in Clause 5 of ISO/IEC 14496-1.

3.5.1.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of Clause 5 of ISO/IEC 14496-1.

3.5.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

3.5.2 Terminal conformance

3.5.2.1 Conformance Requirements

The terminal shall recover the BIFS Elementary Stream in the BIFS Decoding Buffer bit-exact as constructed by the BIFS encoder.

3.5.2.2 Measurement Procedure

The BIFS Access Units recovered from this conformance point shall be strictly identical to the Access Units stored in the corresponding BIFS track in the test MP4 file.

3.5.2.3 Tolerance

There is no tolerance. The diagnosis is pass or fail.

3.6 MPEG-J

3.6.1 MPEG-J Conformance Points

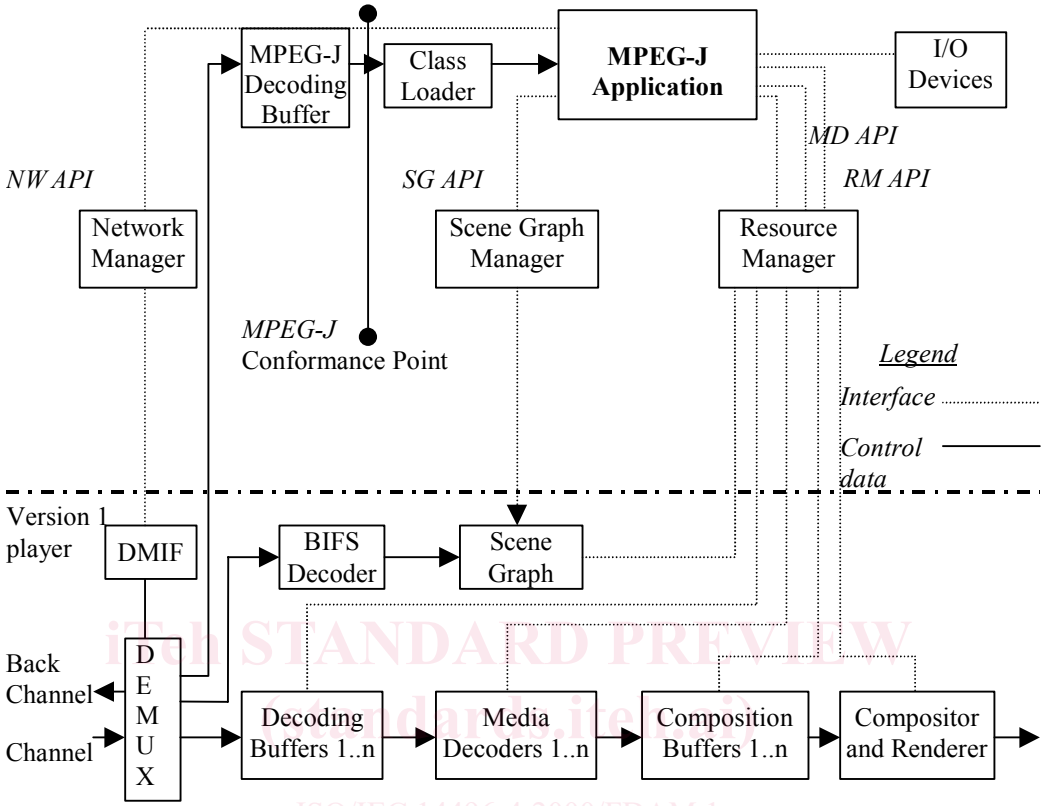

Figure 3-1 — MPEG-J Architecture with Conformance Point

Architecture of MPEG-J is explained in ISO/IEC 14496-1 subclause 11.2. MPEG-J data is defined and the delivery mechanism explained in ISO/IEC 14496-1 subclause 11.4. MPEG-J data is delivered as an elementary stream similar to video, audio and other elementary streams.

This is de-multiplexed and stored in MPEG-J Decoding Buffers. This buffer feeds the MPEG-J Decoder which "decodes" it. In the case of class (Java byte code), decoding means loading, while for the object and other data it is made available to the terminal.

The MPEG-J Decoding Buffer consists of MPEG-J Access Units defined in subclause. Each MPEG-J Access Unit contains either one class or one serialized object or one archive (a zip file) with a header. When this is decoded, the class file or the object data or the zip file is extracted and fed into the MPEG-J Class Loader as shown in ***Figure 3-1***

Bitstream conformance point for MPEG-J is:

- MPEG-J Decoding

At a bitstream conformance point, bitstreams will be acquired for use in testing.

Terminal conformance point for MPEG-J is:

- MPEG-J Decoding Buffer
- MPEG-J API conformance
- Java Platform conformance

An MPEG-J conformance point can be either an MPEG-J bitstream conformance point or an MPEG-J Terminal conformance point. The MPEG-J bitstream conformance points deal with the syntactic aspects while the MPEG-J terminal conformance points address the semantics.

3.6.2 Bitstream Conformance

Each bitstream shall meet the syntactic and semantic requirements specified in ISO/IEC 14496-1. This subclause describes a set of tests to be performed on bitstreams. In the description of the tests it is assumed that the tested bitstream contains no errors due to transmission or other causes. For each test the condition or conditions that must be satisfied are given, as well as the prerequisites or conditions in which the test can be applied. Note that the application of these tests requires parsing of the bitstream to the appropriate levels. Parsing and interpretation of ODs is also required. In some cases of IPMP-protected data, de-scrambling may be required before the tests can be performed on non IPMP-related features.

3.6.2.1 MPEG-J Conformance

3.6.2.1.1 Conformance Requirements

MPEG-J bitstreams shall comply with the specifications in ISO/IEC 14496-1 Clause 11. The terminal shall strictly adhere to the syntax specified in 11.4.3.

When the bitstream carries classes, these classes shall only use the classes, interfaces, or API (Application Programming Interface) calls from the following:

1. MPEG-J APIs defined in the ISO/IEC 14496-1 (org.iso.*) for the relevant profile.

2. Java APIs supported by the underlying Java Platform for the relevant profile. These are (typically) in the java.* packages.

3. Classes or Interfaces carried in the bitstream.

These classes shall obey the security rules defined in subclause 11.3.5 of ISO/IEC 14496-1.

3.6.2.1.2 Measurement procedure

Syntax of the bitstream shall meet the requirements of subclause 11.4.3 of ISO/IEC 14496-1.

The classes should compile with only the Java Platform APIs and the MPEG-J APIs relevant to that profile.

Verification mechanism: The API implementations should output a trace file for every bitstream. This trace files should be compared to see if the behavior is the same in two implementations. This idea is similar to the dump format used for BIFS.

`Method packagename.classname.methodName with parameter parameter`$_1$ `parameter`$_2$ `parameter`$_3$`… parameter`$_n$ `was called`

where: *method_name* is the name of the method, parameter$_n$ is:

- value of the parameter - when it is a primitive data type
- the instance name - otherwise.

E.g. a method foo(var$_1$, var$_2$) would print the trace

`Method org.iso.mpeg.mpegj.foo with parameter var`$_1$ `var`$_2$

`Exception packagename.`*`exception_name`* `was thrown ( or )`

`Exception packagename.`*`exception_name`* `was thrown or with parameter`*`var`*$_1$

3.6.2.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

3.6.3 Terminal Conformance

This subclause describes procedures to verify conformance of terminals. Each compliant decoder shall be able to decode all compliant ISO/IEC 14496-1 streams within the subset of the standard defined by the specified capabilities of the decoder.

All tests are performed using error free bitstreams. To test for correct interpretation of syntax and semantics, test sequences covering a wide range of parameters shall be supplied to the decoder under test and its output sequence shall be compared with the known expected output as described for the specific test sequence or bitstream. The comparison can be done, for example, by performing subjective evaluation, by verification of the expected result, or by comparing the timing performance. Such tests are necessary but not sufficient to prove conformance. They are helpful for discovering non-compliant implementations.

Tests are expected to be used for testing ISO/IEC 14496 decoders, including video and audio decoding, as it is generally not practical to test system decoders (or ISO/IEC 14496-1 decoders) alone. Practical test results depend on successful (or expected) output of the entire ISO/IEC 14496 decoder (systems, video, audio and DMIF).

3.6.3.1 MPEG-J conformance

3.6.3.1.1 Conformance Requirements

Figure 3-1 shows the architecture an MPEG-J Terminal and the conformance points. The terminal shall follow all the rules regarding:

- MPEG-J Session and Lifecycle specified in subclause 11.3 of ISO/IEC 14496-1.
- MPEG-J Decoding and Loading specified in subclause 11.4 of ISO/IEC 14496-1.
- Semantics of the timestamps specified in sub subclause 11.4.2 of ISO/IEC 14496-1.

All the defined and normatively referred APIs defined subclause 11.5 of ISO/IEC 14496-1 in shall be strictly followed.

3.6.3.1.1.1 MPEG-J Decoding

The Decoding process of MPEG-J data involves two steps:

a. Recovering the access unit data (class, object, or zip file) from the bit stream. This is input to the MPEG-J Class Loader.

b. Loading:

- If the data is a class file it is loaded according to the rules specified in subclause 11.4 of ISO/IEC 14496-1.
- If the data is a zip file the classes specified in the header are loaded according to the rules specified in subclause 11.4 of ISO/IEC 14496-1.
- If the data is neither a class or a zip file, it is made available according to the rules specified in subclause 11.4 of ISO/IEC 14496-1.

3.6.3.1.1.2 MPEG-J API conformance

The terminal shall implement all the APIs that are defined or normatively referenced by ISO/IEC 14496-1 for the relevant profile.