INTERNATIONAL STANDARD ISO/IEC 14496-3:1999/Amd.1:2000
TECHNICAL CORRIGENDUM 1

Published 2001-08-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Coding of audio-visual objects —

Part 3: **Audio**

AMENDMENT 1: Audio extensions

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Codage des objets audiovisuels —

Partie 3: Codage audio

AMENDEMENT 1: Extensions audio

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to International Standard ISO/IEC 14496-3:1999/Amd.1:2000 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Voorbeeld
Preview

Dit document mag slechts op een stand-alone PC worden geïnstalleerd. Gebruik op een netwerk is alleen toestaan als een aanvullende licentieovereenkomst voor netwerkgebruik met NEN is afgesloten.
This document may only be used on a stand-alone PC. Use in a network is only permitted when a supplementary license agreement for us in a network with NEN has been concluded.

ICS 35.040 **Ref. No. ISO/IEC 14496-3:1999/Amd.1:2000/Cor.1:2001(E)**

© ISO/IEC 2001 – All rights reserved

Printed in Switzerland

Dit document is een voorbeeld van NEN / This document is a preview by NEN

Throughout the text of ISO/IEC 14496-3:1999/Amd.1:2000, replace all occurrences of "*AL PDU*" with "*SL packet*" and all occurrences of "*alPduPayload*" with "*slPacketPayload*".

In clause "Introduction", add

"

"MPEG-4 has no standard for transport. In all of the MPEG-4 tools for audio and visual coding, the coding standard ends at the point of constructing a sequence of access units that contain the compressed data. The MPEG-4 Systems (ISO/IEC 14496-1:2001) specification describes how to convert the individually coded objects into a bitstream that contains a number of multiplexed sub-streams.

There is no standard mechanism for transport of this stream over a channel; this is because the broad range of applications that can make use of MPEG-4 technology have delivery requirements that are too wide to easily characterize with a single solution. Rather, what is standardized is an interface (the Delivery Multimedia Interface Format, or DMIF, specified in ISO/IEC 14496-6:1999) that describes the capabilities of a transport layer and the communication between transport, multiplex, and demultiplex functions in encoders and decoders. The use of DMIF and the MPEG-4 Systems bitstream specification allows transmission functions that are much more sophisticated than are possible with previous MPEG standards.

However, LATM and LOAS have been defined to provide a Low overhead Audio multiplex and transport mechanism for natural audio applications, which do not require sophisticated object-based coding or other functions provided by MPEG-4 Systems.

The following table gives an overview about the multiplex, storage and transmission formats for MPEG-4 Audio currently available within the MPEG-4 framework:

	Format	Functionality defined in:	Functionality redefined in:	Description
Multiplex	FlexMux	ISO/IEC 14496-1:2001 (MPEG-4 Systems) (Normative)		Flexible multiplex scheme
	LATM	ISO/IEC 14496-3/Amd 1:2000 (MPEG-4 Audio V2) (Normative)		Low Overhead Audio Transport Multiplex
Storage	ADIF	ISO/IEC 13818-7:1997 (MPEG-2 Audio) (Normative)	ISO/IEC 14496-3:1999 (MPEG-4 Audio V1) (Informative)	(MPEG-2 AAC) Audio Data Interchange Format, AAC only
	MP4FF	ISO/IEC 14496-1/Amd. 1:2000 (MPEG-4 Systems V2) (Normative)		MPEG-4 File format
Transmission	ADTS	ISO/IEC 13818-7:1997 (MPEG-2 Audio) (Normative, Exemplarily)	ISO/IEC 14496-3:1999 (MPEG-4 Audio V1) (Informative)	Audio Data Transport Stream, AAC only
	LOAS	ISO/IEC I14496-3/Amd. 1:2000 (MPEG-4 Audio V2) (Normative, Exemplarily)		Low Overhead Audio Stream, based on LATM, three versions are available: AudioSyncStream() EPAudioSyncStream() AudioPointerStream()

".

Note: This text is intended to replace the first bullet in subclause 1.1.2 (New concepts in MPEG-4 Audio) as stated in MPEG-4 Audio (ISO/IEC 14496-3:1999).

iTeh STANDARD Preview

© ISO/IEC 2001 – All rights reserved
Dit document is een voorbeeld van NEN / This document is a preview by NEN

Replace the first row of Table 3 – Complexity of Audio Object Types in subclause 5.2.2 with

"

Object Type	Parameters	PCU (MOPS per channel)	RCU (kWords per channel)	Remarks

".

*In subclause 5.2.3, replace within table footnotes *1 below Table 4 (Levels for the High Quality Audio Profile), Table 5 (Levels for the Low Delay Audio Profile), Table 6 (Levels for the Natural Audio Profile) and Table 7 (Levels for the Mobile Audio Internetworking Profile)*

"

..., which has the longest frame length, in each profile & level.

"

with

"

..., which has the longest maximum frame length, in each profile & level. For audio object types supporting variable frame lengths and arbitrary bitrates (i.e. any AAC audio object type) this does just extend the required decoder input buffer but does not affect the amount of redundancy actually applied.

".

In Table 8 (Syntax of AudioSpecificInfo()) in subclause 6.2.1, replace

"

epConfig;	**2**	**bslbf**
if (epConfig == 2)		
ErrorProtectionSpecificConfig();		

"

with

"

epConfig;	**2**	**uimsbf**
if (epConfig == 2 || epConfig == 3) {		
ErrorProtectionSpecificConfig();		
}		
if (epConfig == 3) {		
directMapping;	**1**	**bslbf**
if (! directMapping) {		
/* tbd */		
}		
}		

".

In subclause 6.3.5, replace

"

This variable signals what kind of error robust configuration is used, i. e. how instances of error sensitivity categories are obtained on decoder site.

Zonder Preview

© ISO/IEC 2001 – All rights reserved

Dit document is een voorbeeld van NEN / This document is a preview by NEN

Table 10: epConfig

epConfig	Description
0	All instances of all sensitivity categories belonging to one frame are stored within one access unit.
1	Each instance of each sensitivity category belonging to one frame is stored separately within a single access unit, i.e. there are as many elementary streams existent as instances defined within a frame.
2	The error protection decoder has to be applied. Its input is an error protected access unit and its output are several error protection class instances. Each instance of each sensitivity category belonging to one frame corresponds to one of these error protection class instances.
3	Reserved

"

with

"

This data element signals what kind of error robust configuration is used.

Table 10: epConfig

epConfig	Description
0	All instances of all error sensitivity categories belonging to one frame are stored within one access unit. There exists one elementary stream per scalability layer, or just one elementary stream in case of non-scalable configurations.
1	Each instance of each sensitivity category belonging to one frame is stored separately within a single access unit, i.e. there exist as many elementary streams as instances defined within a frame.
2	The error protection decoder has to be applied. Its input are error protected access units. There exists one elementary stream per scalability layer, or just one elementary stream in case of non-scalable configurations. The output of the EP decoder is a set of several error protection classes. The concatenation of EP classes at the error protection decoder output is equivalent to epConfig=0 data.
3	The error protection decoder has to be applied. Its input are error protected access units. There exists one elementary stream per scalability layer, or just one elementary stream in case of non-scalable configurations. The output of the EP decoder is a set of several error protection classes. The concatenation of EP classes at the error protection decoder output is equivalent to epConfig=0 data. The mapping between EP classes and ESC instances is signaled by the data element directMapping.

".

After subclause 6.3.5, add

"

6.3.6 direct mapping

This data element identifies the mapping between error protection classes and error sensitivity category instances.

ZIP-ISO/IEC Preview

© ISO/IEC 2001 – All rights reserved
Dit document is een voorbeeld van NEN / This document is a preview by NEN

directMapping

directMapping	Description
0	Reserved
1	Each error protection class is treated as an instance of an error sensitivity category (one to one mapping), so that the error protection decoder output is equivalent to epConfig=1 data.

".

To the end of subclause 6.5.1, add

"

The mechanism defined in this subclause should not be used for transmission of TTSI object (12), Main Synthetic object (13), Wavetable Synthesis object (14), General MIDI object (15) and Algorithmic Synthesis and Audio FX object (16). For these object types, other multiplex and transport mechanisms might be used, e.g. those defined in MPEG-4 Systems.

".

To the description of audioMuxElementStartPointer in subclause 6.5.2.2, add

"

The maximum possible value of this field is reserved to signal that there is no start of an access unit in this sync frame.

".

Replace Table 17 (Syntax of AudioMuxElement()) in subclause 6.5.3.1 (Syntax) with

"

Syntax	No. of bits	Mnemonic
AudioMuxElement(muxConfigPresent)		
{		
if(muxConfigPresent) {		
useSameStreamMux;	**1**	**bslbf**
if (!useSameStreamMux)		
StreamMuxConfig();		
}		
if (audioMuxVersion == 0) {		
for(i=0; i<=numSubFrames; i++) {		
PayloadLengthInfo();		
PayloadMux();		
}		
if(otherDataPresent) {		
for(i=0; i<otherDataLenBits; I++) {		
otherDataBit;	**1**	**bslbf**
}		
}		
}		
else {		
/* tbd */		
}		
}		

".

Preview

© ISO/IEC 2001 – All rights reserved

Dit document is een voorbeeld van NEN / This document is a preview by NEN

Replace table 18 (Syntax of StreamMuxConfig) in subclause 6.5.3.1 (Syntax) with

"

Syntax	No. of bits	Mnemonic
StreamMuxConfig()		
{		
audioMuxVersion;	**1**	**bslbf**
if (audioMuxVersion == 0) {		
streamCnt = 0;		
allStreamsSameTimeFraming;	**1**	**uimsbf**
numSubFrames;	**6**	**uimsbf**
numProgram;	**4**	**uimsbf**
for (prog = 0; prog <= numProgram; prog++) {		
numLayer;	**3**	**uimsbf**
for (lay = 0; lay <= numLayer; lay++) {		
progSIndx[streamCnt]=prog; laySIndx[streamCnt]=lay;		
streamID [prog][lay] = streamCnt++;		
if (prog == 0 & lay == 0) {		
AudioSpecificInfo();		
}else {		
useSameConfig;	**1**	**uimsbf**
if (!useSameConfig)		
AudioSpecificInfo();		
}		
frameLengthType[streamID[prog][lay]];	**3**	**uimsbf**
if (frameLengthType[streamID[prog][lay] == 0) {		
bufferFullness[streamID[prog][lay]];	**8**	**uimsbf**
if (!allStreamsSameTimeFraming) {		
if ((AudioObjectType[lay]==6 ||		
AudioObjectType[lay]== 20) &&		
(AudioObjectType[lay-1]==8 ||		
AudioObjectType[lay-1]==24)) {		
coreFrameOffset;	**6**	**uimsbf**
}		
}		
} else if(frameLengthType[streamID[prog][lay]] == 1) {		
frameLength[streamID[prog][lay]];	**9**	**uimsbf**
} else if (frameLengthType[streamID[prog][lay]] == 4		
frameLengthType[streamID[prog][lay]] == 5		
frameLengthType[streamID[prog][lay]] == 3) {		
CELPframeLengthTableIndex[streamID[prog][lay]];	**6**	**uimsbf**
} else if (frameLengthType[streamID[prog][lay]] == 6		
frameLengthType[streamID[prog][lay]] == 7) {		
HVXCframeLengthTableIndex[streamID[prog][lay]];	**1**	**uimsbf**
}		
}		
}		
otherDataPresent;	**1**	**uimsbf**
if (otherDataPresent) {		
otherDataLenBits = 0; /* helper variable 32bit */		
do {		
otherDataLenBits = otherDataLenBits * 2^8;		
otherDataLenEsc;	**1**	**uimsbf**
otherDataLenTmp;	**8**	**uimsbf**
otherDataLenBits = otherDataLenBits + otherDataLenTmp;		
} while (otherDataLenEsc);		

LICENSED TO NEN Preview

© ISO/IEC 2001 – All rights reserved

Dit document is een voorbeeld van NEN / This document is a preview by NEN

```
        }
        crcCheckPresent;                                    1       uimsbf
        if( crcCheckPresent ) crcCheckSum;                  8       uimsbf
    }
    else {
        /* tbd */
    }
}
```

".

Voorbeeld
Preview

© ISO/IEC 2001 – All rights reserved

Dit document is een voorbeeld van NEN / This document is a preview by NEN

Replace Table 19 (Syntax of PayloadLengthInfo()) in subclause 6.5.3.1 (Syntax) with

"

Syntax	No. of bits	Mnemonic
PayloadLengthInfo()		
{		
if(allStreamsSameTimeFraming) {		
for (prog = 0; prog <= numProgram; prog++) {		
for (lay = 0; lay <= numLayer; lay++) {		
if(frameLengthType[streamID[prog][lay]] == 0) {		
do { /* always one complete access unit */		
tmp;	**8**	**uimsbf**
MuxSlotLengthBytes[streamID[prog][lay]] += tmp;		
} while(tmp==255);		
} else {		
if (frameLengthType[streamID[prog][lay]] == 5 ||		
frameLengthType[streamID[prog][lay]] == 7 ||		
frameLengthType[streamID[prog][lay]] == 3) {		
MuxSlotLengthCoded[streamID[prog][lay]];	**2**	**uimsbf**
}		
}		
}		
}		
} else {		
numChunk;	**4**	**uimsbf**
for (chunkCnt=0; chunkCnt <= numChunk; chunkCnt++) {		
streamIndx;	**4**	**uimsbf**
prog = progCIndx[chunkCnt] = progSIndx[streamIndx]**;**		
lay = layCIndx[chunkCnt] = laySIndx [streamIndx];		
if(frameLengthType[streamID[prog][lay]] == 0) {		
do { /* not necessarily a complete access unit */		
tmp;	**8**	**uimsbf**
MuxSlotLengthBytes[streamID[prog][lay]] += tmp;		
} while (tmp == 255);		
AuEndFlag[streamID[prog][lay]];	**1**	**bslbf**
} else {		
if (frameLengthType[streamID[prog][lay]] == 5 ||		
frameLengthType[streamID[prog][lay]] == 7 ||		
frameLengthType[streamID[prog][lay]] == 3) {		
MuxSlotLengthCoded[streamID[prog][lay]];	**2**	**uimsbf**
}		
}		
}		
}		
}		

".

ISO/IEC Preview

© ISO/IEC 2001 – All rights reserved

Dit document is een voorbeeld van NEN / This document is a preview by NEN

Replace Table 20 (Syntax of PayloadMux()) in subclause 6.5.3.1 (Syntax) with

"

Syntax	No. of bits	Mnemonic

PayloadMux()
{
if(allStreamsSameTimeFraming) {
for (prog = 0; prog <= numProgram; prog++) {
for (lay = 0; lay <= numLayer; lay++) {
payload [streamID[prog][lay]];
}
}
} else {
for (chunkCnt=0; chunkCnt <= numChunk; chunkCnt++) {
prog = progCIndx[chunkCnt];
lay = layCIndx [chunkCnt];
 	
payload [streamID[prog][lay]];
}
}
}		

".

In subclause 6.5.3.2 (Semantics), add

"

audioMuxVersion: A data element to signal the bitstream syntax version. possible values: 0 (default), 1 (reserved for future extensions).

".

In subclause 6.5.3.2 (Semantics), replace

"

Note 1: There might be more than one instance of error sensitivity category 1 and 2 depending on the value of the variable **numSubFrames** defined in **StreamMuxConfig()**. Figure 3 shows an example for the order of the instances assuming numSubFrames is two (2).

"

with

"

Note 1: There might be more than one instance of error sensitivity category 1 and 2 depending on the value of the variable **numSubFrames** defined in **StreamMuxConfig()**. Figure 3 shows an example for the order of the instances assuming numSubFrames is one (1).

".

In subclause 6.5.3.2 (Semantics), replace

"

numSubFrames A field indicating how many PayloadMux() frames are multiplexed. If more than one PayloadMux() frame are multiplexed, all PayloadMux() share a common StreamMuxConfig().The minimum value is 1.

numProgram A field indicating how many programs are multiplexed. The minimum value is 1.

numLayer A field indicating how many scalable layers are multiplexed. The minimum value is 1.

"

Withdrawn
Preview

© ISO/IEC 2001 – All rights reserved

Dit document is een voorbeeld van NEN / This document is a preview by NEN

Bestelformulier

NEN

Stuur naar:

NEN Standards Products & Services
t.a.v. afdeling Klantenservice
Antwoordnummer 10214
2600 WB Delft

NEN Standards Products & Services

Postbus 5059
2600 GB Delft

Vlinderweg 6
2623 AX Delft

T (015) 2 690 390
F (015) 2 690 271

www.nen.nl/normshop

Ja, ik bestel

__ ex. ISO/IEC 14496-3:1999/Amd 1:2000/Cor 1:2001 en Informatietechnologie - Codering van audio-visuele objecten - Deel 3: Audio — € 0.00

Wilt u deze norm in PDF-formaat? Deze bestelt u eenvoudig via www.nen.nl/normshop

Gratis e-mailnieuwsbrieven

Wilt u op de hoogte blijven van de laatste ontwikkelingen op het gebied van normen, normalisatie en regelgeving? Neem dan een gratis abonnement op een van onze e-mailnieuwsbrieven. www.nen.nl/nieuwsbrieven

Gegevens

Bedrijf / Instelling

T.a.v. O M O V

E-mail

Klantnummer NEN

Uw ordernummer BTW nummer

Postbus / Adres

Postcode Plaats

Telefoon Fax

Factuuradres (indien dit afwijkt van bovenstaand adres)

Postbus / Adres

Postcode Plaats

Datum Handtekening

Retourneren

Fax: (015) 2 690 271
E-mail: klantenservice@nen.nl
Post: NEN Standards Products & Services,
t.a.v. afdeling Klantenservice
Antwoordnummer 10214,
2600 WB Delft
(geen postzegel nodig).

Voorwaarden

- De prijzen zijn geldig tot 31 december 2016, tenzij anders aangegeven.
- Alle prijzen zijn excl. btw, verzend- en handelingskosten en onder voorbehoud bij o.m. ISO- en IEC-normen.
- Bestelt u via de normshop een pdf, dan betaalt u geen handeling en verzendkosten.
- Meer informatie: telefoon (015) 2 690 391, dagelijks van 8.30 tot 17.00 uur.
- Wijzigingen en typefouten in teksten en prijsinformatie voorbehouden.
- U kunt onze algemene voorwaarden terugvinden op: www.nen.nl/leveringsvoorwaarden.

Normalisatie: de wereld op één lijn.