INTERNATIONAL STANDARD ISO/IEC 13818-4:1998/Amd.1:1999
TECHNICAL CORRIGENDUM 1

Published 2003-03-01

Information technology — Generic coding of moving pictures and associated audio information —

Part 4:
Conformance testing

AMENDMENT 1: Advanced Audio Coding (AAC) conformance testing

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Codage générique des images animées et des informations sonores associées —

Partie 4: Essais de conformité

AMENDEMENT 1: Essai de conformité de codage audio avancé

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 13818-4:1998/Amd.1:1999 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information.*

―――――――――――――

Ref. No. ISO/IEC 13818-4:1998/Amd.1:1999/Cor.1:2003(E)

Throughout the text, replace
"

program_configuration_element
"

with
"

program_config_element
".

In subclause 2.6.4.2 (Decoder Modifications), replace
"

Bitstream Characteristic	Normative Clause	Variation
sampling rate	3	a decoder may support only a subset of possible sampling rates, decoder manufacturers must specify the rates which are supported
profile	2.1	a decoder may support additional channel elements beyond the minimums listed for its profile
program configuration	3.5	a decoder is only required to decode one program of a multi-program bitstream
data_stream_element	3.6	a decoder is not required to store or present data recovered from data_stream_elements
mono-mixdown element	3.3.8	a decoder is not required to present audio from the mono-mixdown element
stereo-mixdown element	3.3.8	a decoder is not required to present audio from the stereo-mixdown element
matrix-mixdown	3.3.8	a decoder is not required to calculate a matrix-mixdown signal

"

with

"

Bitstream Characteristic	Normative Clause	Variation/Restriction
sampling rate	8	a decoder may support only a subset of possible sampling rates, decoder manufacturers must specify the rates which are supported
profile	7.1	a decoder may support additional channel elements beyond the minimums listed for its profile
program configuration	8.5	a decoder is only required to decode one program of a multi-program bitstream
		the specified minimum decoder capabilities refer to the channels within a bitstream (not within a program)
		if program config elements are used, syntactic elements (other than ID_FILL or ID_END) not referenced by any program config element are not allowed
		a decoder conforming to the Main profile, the LC profile, or the SSR profile is not required to support bitstreams containing more than one program (in the sense of what is specified in a program config element)
data_stream_element	8.6	a decoder is not required to store or present data recovered from data_stream_elements
mono-mixdown element	8.3.8	a decoder conforming to the Main profile, the LC profile, or the SSR profile is not required to support bitstreams containing any mono-mixdown element
stereo-mixdown element	8.3.8	a decoder conforming to the Main profile, the LC profile, or the SSR profile is not required to support bitstreams containing any stereo-mixdown element
matrix-mixdown	8.3.8	a decoder is not required to calculate a matrix-mixdown signal

".

Add subclause 2.6.5.0 (Profile Specification)

"

2.6.5.0 Profile Specification

The same naming convention as used for decoders is applied to bitstreams (see 2.6.4.1). The channel specification within the naming convention is based on the bitstream, independent of any program specification. There shall be only one program (in the sense of what is specified in a program config element) in an Main, Low Complexity, or SSR profile conformant bitstream.

".

In subclause 2.6.5.1 (Parsing an ADIF header), add

"

copyright_id_present: no restrictions apply.
copyright_id: cross check with the Registration Authority.
original_copy: no restrictions apply.
home: no restrictions apply.
bitstream_type: no restrictions apply.
bitrate: no restrictions apply.
num_program_config_elements: shall be 0 for the profiles AAC Main, AAC LC and AAC SSR.
adif_buffer_fullness: shall be in the range of 0 and 6144 * NCC - mean_frame_length (in bits).
".

In subclause 2.6.5.2.1 (adts_fixed_header), add

"

All data elements of the adts_fixed_header() shall not change from frame to frame.
protection_absent: no restrictions apply, but the data element shall not change from frame to frame.
private_bit: no restrictions apply, but the data element shall not change from frame to frame.
channel_configuration: no restrictions apply, but the data element shall not change from frame to frame.
original_copy: no restrictions apply, but the data element shall not change from frame to frame.
home: no restrictions apply, but the data element shall not change from frame to frame.
".

In subclause 2.6.5.2.2 (adts_variable_header), replace

"

frame_length: shall be encoded with the length of the frame, including headers and error check (if present)
"

with

"

copyright_identification_bit: cross check with the Registration Authority.
copyright_identification_start: ´1´ if new copyright id starts, otherwise ´0´.
frame_length: shall be encoded with the length of the adts_frame in bytes.
adts_buffer_fullness: shall be in the range of 0 and INT ((6144 - mean_frame_length / NCC) / 32).
number_of_raw_data_blocks_in_frame: no restrictions apply.
".

Add subclause 2.6.5.2.3 (adts_error_check)

"

2.6.5.2.3 adts_error_check

crc_check: refer to the definition of adts_error_check() in ISO/IEC 13818-7.
".

Add subclause 2.6.5.2.4 (adts_header_error_check)

"

2.6.5.2.4 adts_header_error_check

raw_data_block_position[i]: shall be encoded with the start position of raw_data_block(i) measured as an offset in bytes from start position of raw_data_block(0).
crc_check: refer to the definition of adts_header_error_check() in ISO/IEC 13818-7.
".

Add subclause 2.6.5.2.5 (adts_raw_data_block_error_check)
"

2.6.5.2.5 adts_raw_data_block_error_check

crc_check: refer to the definition of adts_raw_data_block_error_check() in ISO/IEC 13818-7.
".

In subclause 2.6.5.3 (Decoding of raw data blocks), add at the beginning
"

If a program_config_element() is used to specify the content of a raw_data_stream(), any subsequent raw_data_block() shall contain exactly the specified elements (by means of type, number and element_instance_tag).
".

In subclause 2.6.5.9 (Prediction), replace
"

predictor_reset: shall be encoded with the binary value of 1 for at least one out of every eight consecutive frames for programs in which predictor_data_present is 1.
predictor_reset_group_number: shall not be encoded with the binary values 00000 or 11111; every valid value shall occur once within a consecutive series of 30 occurrences of pred_reset_group_number.
"

with

"

predictor_reset: marks the presence of **predictor_reset_group_number** for a given frame. **Predictor_reset** on its own can not be tested for conformance, but conformant use of **predictor_reset_group_number** implies conformant use of **predictor_reset**.

predictor_reset_group_number: shall not be encoded with the binary values 00000 or 11111 (undefined reset group) and shall not violate the following set of rules (maximum period of 240 frames with predictor activity):
- Define a predictor reset group counter ('activity counter') for each predictor group and initialize all counters to zero.
- Increment the counter for a given group, whenever any (or all) of the associated **prediction_used[**sfb**]** flags for that particular group are active ('1') during a frame. Note that prediction groups may be active independently of each other and that frames with predictor activity may be interleaved with an arbitrary number of frames without any predictor activity.
- If window_sequence equals EIGHT_SHORT_SEQUENCE, all predictor reset group counters are reset to zero.
- If **predictor_reset** has the value '1', the counter indicated by **predictor_reset_group_number** is reset to zero.
- A predictor group reset shall be signaled sufficiently often that the associated predictor reset group counter ("activity counter") never reaches the value of 240 after all possible reset or increment operations have been considered. Note, that a 'predictor group reset' resets only one group at a time so that it takes 30 frames to reset all predictor reset groups.

".