INTERNATIONAL STANDARD

ISO/IEC 14496-4

Second edition
2004-12-15

AMENDMENT 46
2019-04

Information technology — Coding of audio-visual objects —

Part 4: Conformance testing

AMENDMENT 46: Conformance testing for internet video coding

Reference number
ISO/IEC 14496-4:2004/Amd.46:2019(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: 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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 14496 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Information technology — Coding of audio-visual objects —

Part 4: Conformance testing

AMENDMENT 46: Conformance testing for internet video coding

Clause 2

Add the following new references:

ISO/IEC 14496-5, *Information technology — Coding of audio-visual objects — Part 5: Reference software*

ISO/IEC 14496-33, *Information technology — Coding of audio-visual objects — Part 33: Internet video coding*

Clause 13

Add the following new clause:

13 Internet video coding

13.1 General

This clause specifies tests designed to verify whether bitstreams and decoders meet the normative requirements specified in ISO/IEC 14496-33.

Characteristics of coded bitstreams and decoders are defined in ISO/IEC 14496-33. An encoder can claim conformance to ISO/IEC 14496-33 if the bitstreams that it generates are conforming bitstreams. Decoder characteristics define the properties and capabilities of the applied decoding process. The capabilities of a decoder specify which bitstreams the decoder can decode and reconstruct. A bitstream can be decoded by a decoder if the characteristics of the bitstream are within the specified decoder capabilities.

Procedures are described for testing conformance of bitstreams and decoders to the requirements defined in ISO/IEC 14496-33. Given the set of characteristics claimed, the requirements that shall be met are fully determined by ISO/IEC 14496-33. This clause summarizes the requirements, cross references them to characteristics, and defines how conformance with them can be tested. This clause also gives guidelines on how to construct bitstream test suites to check or verify decoder conformance. In addition, a set of test bitstreams implemented according to those guidelines are provided at http://standards.iso.org/iso-iec/14496/-4/ed-2/amd/46/en.

13.2 Conformance for ISO/IEC 14496-33

13.2.1 Normative tests for verifying conformance

The following subclauses specify normative tests for verifying conformance of video bitstreams as well as decoders. These normative tests make use of test data (bitstream test suites) provided at http://standards.iso.org/iso-iec/14496/-4/ed-2/amd/46/en and the reference software decoder provided in ISO/IEC 14496-5 for ISO/IEC 14496-33 with source code included in ISO/IEC 14496-5 in electronic format.

13.2.2 Procedure to test bitstreams

A bitstream that claims conformance with ISO/IEC 14496-33 shall pass the following normative test.

The bitstream shall be decoded by processing it with the reference software decoder provided in ISO/IEC 14496-5 for ISO/IEC 14496-33. When processed by the reference software decoder, the bitstream shall not cause any error or non-conformance messages to be reported by the reference software decoder. This test should not be applied to bitstreams that are known to contain errors introduced by transmission.

Successfully passing the reference software decoder test provides only a strong presumption that the bitstream under test is conforming to the video coding specification, i.e. that it does indeed meet all the requirements for ISO/IEC 14496-33 that are tested by the reference software decoder.

Additional tests may be necessary to more thoroughly check that the bitstream properly meets all the requirements specified in ISO/IEC 14496-33. These complementary tests may be performed using other video bitstream verifiers that perform more complete tests than those implemented by the reference software decoder.

To check correctness of a bitstream, it is necessary to parse the entire bitstream and to extract all the syntactic elements and other values derived from those syntactic elements and used by the decoding process specified in ISO/IEC 14496-33.

A verifier may not necessarily perform all stages of the decoding process described in ISO/IEC 14496-33 in order to verify bitstream correctness. Many tests can be performed on syntax elements in a state prior to their use in some processing stages.

13.2.3 Procedure to test decoder conformance

13.2.3.1 Conformance bitstreams

Conformance bitstreams are available at http://standards.iso.org/iso-iec/14496/-4/ed-2/amd/46/en. The following information is included in a single zipped file for each such bitstream.

Bitstream

MD5 sum file (for each decoded frame)

The reference software decoder provided in ISO/IEC 14496-5 shall be used to generate the necessary reference decoded frames from the bitstream.

13.2.3.2 Requirements on output of the decoding process

It is a requirement that all of the decoded frames output by a conforming decoder shall be in the same order as output by the reference software decoder. It is a further requirement that all of the values of samples output by a conforming decoder, prior to post processing, shall be exactly equal to the values of the corresponding samples output by the reference software decoder.

13.2.4 Test bitstreams — IVC

13.2.4.1 Test bitstreams #TRANS_A

Specification: All slices are coded as I slices. Each frame contains only one slice. abt_enable is set to 1.

Functional stage: Test the reconstruction process of I slices with variable block-size transform and intra prediction.

Purpose: Check if decoder can correctly decode I slices with variable block-size transform and intra prediction.

13.2.4.2 Test bitstreams #TRANS_B

Specification: All slices are coded as I and P slices. Each frame contains only one slice. abt_enable is set to 1.

Functional stage: Test the reconstruction process of I and P slices with variable block-size transform.

Purpose: Check if decoder can correctly decode I and P slices with variable block-size transform.

13.2.4.3 Test bitstreams #TRANS_C

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. abt_enable is set to 1.

Functional stage: Test the reconstruction process of I, P, and B slices with variable block-size transform.

Purpose: Check if decoder can correctly decode I, P, and B slices with variable block-size transform.

13.2.4.4 Test bitstreams #MULQP_A

Specification: All slices are coded as I slices. Each frame contains only one slice. Each slice has a random QP value in the range of 0 to 63.

Functional stage: The reconstruction process of I slices with various QP values.

Purpose: Check if the decoder can handle I slices with various QP values.

13.2.4.5 Test bitstreams #MULQP_B

Specification: All slices are coded as I and P slices. Each frame contains only one slice. Each slice has a random QP value in the range of 0 to 63.

Functional stage: The reconstruction process of I and P slices with various QP values.

Purpose: Check if the decoder can handle I and P slices with various QP values.

13.2.4.6 Test bitstreams #MULQP_C

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. Each slice has a random QP value in the range of 0 to 63.

Functional stage: The reconstruction process of I, P, and B slices with various QP values.

Purpose: Check if the decoder can handle I, P, and B slices with various QP values.

13.2.4.7 Test bitstreams #MULSLICE_A

Specification: All slices are coded as I slices. Each frame contains a random number of slices from 1 to 10.

Functional stage: Test the reconstruction process of I frames with multiple slices.

Purpose: Check if the decoder can correctly decode I frames with multiple slices.

13.2.4.8 Test bitstreams #MULSLICE_B

Specification: All slices are coded as I and P slices. Each frame contains a random number of slices from 1 to 10.

Functional stage: Test the reconstruction process of I and P frames with multiple slices.

Purpose: Check if the decoder can correctly decode I and P frames with multiple slices.

13.2.4.9 Test bitstreams #MULSLICE_C

Specification: All slices are coded as I, P, and B slices. Each frame contains a random number of slices from 1 to 10.

Functional stage: Test the reconstruction process of I, P, and B frames with multiple slices.

Purpose: Check if the decoder can correctly decode I, P, and B frames with multiple slices.

13.2.4.10 Test bitstreams #DBLK_A

Specification: All slices are coded as I slices. Each frame contains only one slice. Each slice has a random QP value in the range of 22 to 63.

Functional stage: Test the deblocking filter process of I slices with various QP values.

Purpose: Check if the decoder can correctly decode I slices with various QP values.

13.2.4.11 Test bitstreams #DBLK_B

Specification: All slices are coded as I and P slices. Each frame contains only one slice. Each slice has a random QP value in the range of 22 to 63.

Functional stage: Test the deblocking filter process of I and P slices with various QP values.

Purpose: Check if the decoder can correctly decode I and P slices with various QP values.

13.2.4.12 Test bitstreams #DBLK_C

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. Each slice has a random QP value in the range of 22 to 63.

Functional stage: Test the deblocking filter process of I, P, and B slices with various QP values.

Purpose: Check if the decoder can correctly decode slices with various QP values.

13.2.4.13 Test bitstreams #DBLK_D

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. loop_filter_disable is randomly set to 0 or 1.

Functional stage: Test the deblocking filter process.

Purpose: Check if the decoder can correctly handle deblocking filter with the deblocking filter randomly turned on/off in each slice.

13.2.4.14 Test bitstreams #DBLK_E

Specification: All slices are coded as I, P, and B slices. Each frame contains more than one slice.

Functional stage: Test the deblocking filter process of frames with multiple slices.

Purpose: Check if the decoder can correctly handle deblocking filter with multiple slices.

13.2.4.15 Test bitstreams #DBLK_F

Specification: All slices are coded as I, P and B slices. Each frame contains more than one slice. Each slice has a random QP value in the range of 22 to 63.

Functional stage: Test the deblocking filter process of frames with multiple slices and various QP values.

Purpose: Check if the decoder can correctly handle deblocking filter with multiple slices and various QP values.

13.2.4.16 Test bitstreams #INFTYPE_A

Specification: All slices are coded as I and P slices. Each frame contains only one slice. if_type is set to 1.

Functional stage: Test reconstruction process of P slices by motion compensation with adaptive taps interpolation filter.

Purpose: Check if the decoder can correctly reconstruct P slices by motion compensation with adaptive taps interpolation filter.

13.2.4.17 Test bitstreams #INFTYPE_B

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. if_type is set to 1.

Functional stage: Test reconstruction process of P and B slices by motion compensation with adaptive taps interpolation filter.

Purpose: Check if the decoder can correctly reconstruct P and B slices by motion compensation with adaptive taps interpolation filter.

13.2.4.18 Test bitstreams #INFTYPE_C

Specification: All slices are coded as I, P and B slices. Each frame contains more than one slice. if_type is set to 1.

Functional stage: Test reconstruction process of P and B slices by motion compensation with adaptive taps interpolation filter and multiple slices within one frame.

Purpose: Check if the decoder can correctly reconstruct P and B slices by motion compensation with adaptive taps interpolation filter and multiple slices within one frame.

13.2.4.19 Test bitstreams #NONREFP_A

Specification: All slices are coded as I and P slices. Each frame contains only one slice. intra_period is set to 0. non-reference_P_frame_coding is set to 1, and non-adaptive_non-reference_P_frame_coding is set to 1.

Functional stage: Test the reconstruction process of slices with non-adaptive non-reference P frame coding.

Purpose: Check if the decoder can properly decode slices with non-adaptive non-reference P frame coding.

13.2.4.20 Test bitstreams #NONREFP_B

Specification: All slices are coded as I and P slices. Each frame contains only one slice. intra_period is set to 0. non-reference_P_frame_coding is set to 1, and non-adaptive_non-reference_P_frame_coding is set to 0.

Functional stage: Test the reconstruction process of slices with adaptive non-reference P frame coding.

Purpose: Check if the decoder can properly decode slices with adaptive non-reference P frame coding

13.2.4.21 Test bitstreams #MVRANGE_A

Specification: All slices are coded as I and P slices. Each frame contains only one slice. The range of motion vectors is randomly set from 16 to 512.

Functional stage: Test the decoding process of P slices with various motion ranges.

Purpose: Check if the decoder can properly handle the decoding process of P slices with various motion ranges.

13.2.4.22 Test bitstreams #MVRANGE_B

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. The range of motion vectors is set to a random value in the range of 16 to 512.

Functional stage: Test the decoding process of P and B slices with various motion ranges.

Purpose: Check if the decoder can properly handle the decoding process of P and B slices with various motion ranges.

13.2.4.23 Test bitstreams #MULH_A

Specification: All slices are coded as I and P slices. Each frame contains only one slice. Multiple-hypothesis prediction is not used.

Functional stage: Test the decoding process of P slices without multiple-hypothesis prediction.

Purpose: Check if the decoder can properly handle the decoding process of P slices without multiple-hypothesis prediction.

13.2.4.24 Test bitstreams #MULH_B

Specification: All slices are coded as I and P slices. Each frame contains only one slice. Multiple-hypothesis prediction is enabled.

Functional stage: Test the decoding process of P slices with multiple-hypothesis prediction.

Purpose: Check if the decoder can properly handle the decoding process of P slices with multiple-hypothesis prediction.

13.2.4.25 Test bitstreams #BLKSIZE_A

Specification: All slices are coded as I and P slices. Each frame contains only one slice. Only 16×16 macroblock partitions are used in inter prediction.

Functional stage: Test the decoding process of P slices with 16×16 motion compensation.

Purpose: Check if the decoder can properly handle the decoding process of P slices with 16×16 motion compensation.

13.2.4.26 Test bitstreams #BLKSIZE_B

Specification: All slices are coded as I and P slices. Each frame contains only one slice. Only 16×16 and 8×8 macroblock partitions are used in inter prediction.

Functional stage: Test the decoding process of P slices with 16×16 and 8×8 motion compensation.

Purpose: Check if the decoder can properly handle the decoding process of P slices with 16×16 and 8×8 blocks motion compensation.

13.2.4.27 Test bitstreams #BLKSIZE_C

Specification: All slices are coded as I and P slices. Each frame contains only one slice. 16×16,16×8, 8×16 and 8×8 macroblock partitions are used in inter prediction.

Functional stage: Test the decoding process of P slices with variable block-size motion compensation.

Purpose: Check if the decoder can properly handle the decoding process of P slices with variable block-size motion compensation.

13.2.4.28 Test bitstreams #BLKSIZE_D

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. Only 16×16 macroblock partitions are used in inter prediction.

Functional stage: Test the decoding process of slices with 16×16 block motion compensation.

Purpose: Check if the decoder can properly handle the decoding process of slices with 16×16 block motion compensation.

13.2.4.29 Test bitstreams #BLKSIZE_E

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. Only 16×16 and 8×8 macroblock partitions are used in inter prediction.

Functional stage: Test the decoding process of slices with 16×16 and 8×8 blocks motion compensation.

Purpose: Check if the decoder can properly handle the decoding process of slices with 16×16 and 8×8 blocks motion compensation.

13.2.4.30 Test bitstreams #BLKSIZE_F

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. 16×16, 16×8, 8×16 and 8×8 macroblock partitions are used in inter prediction.

Functional stage: Test the decoding process of slices with variable block-size motion compensation.

Purpose: Check if the decoder can properly handle the decoding process of slices with variable block-size motion compensation.

13.2.4.31 Test bitstreams #IPRED_A

Specification: All slices are coded as I slices. Each frame contains only one slice. All intra prediction modes (5 modes for luma, 4 modes for chroma) are used.

Functional stage: Test the reconstruction process of I slices.

Purpose: Check if the decoder can properly decode I slices with all intra prediction modes.

13.2.4.32 Test bitstreams #IPRED_B

Specification: All slices are coded as I slices. Each frame contains only one slice. All intra prediction modes (5 modes for luma, 4 modes for chroma) are used. Each slice has a random QP value in the range of 22 to 63.

Functional stage: Test the reconstruction process of I slices with multiple QP values.

Purpose: Check if the decoder can properly decode I slices with all intra prediction modes and multiple QP values.

13.2.4.33 Test bitstreams #PDIS_A

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. The distance between two neighbouring P frames is set to 2.

Functional stage: Test the decoding process of motion vector derivation for B_SKIP and Pred_Sym modes.

Purpose: Check if the decoder can properly handle the decoding process of slices with B_SKIP and Pred_Sym modes.

13.2.4.34 Test bitstreams #PDIS_B

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. The distance between two neighbouring P frames is set to 3.

Functional stage: Test the decoding process of motion vector derivation for B_SKIP and Pred_Sym modes.

Purpose: Check if the decoder can properly handle the decoding process of slices with B_SKIP and Pred_Sym modes.

13.2.4.35 Test bitstreams #PDIS_C

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. The distance between two neighbouring P frames is set to 4.

Functional stage: Test the decoding process of motion vector derivation for B_SKIP and Pred_Sym modes.

Purpose: Check if the decoder can properly handle the decoding process of slices with B_SKIP and Pred_Sym modes.

13.2.4.36 Test bitstreams #PDIS_D

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. The distance between two neighbouring P frames is set to 5.

Functional stage: Test the decoding process of motion vector derivation for B_SKIP and Pred_Sym modes.

Purpose: Check if the decoder can properly handle the decoding process of slices with B_SKIP and Pred_Sym modes.

13.2.4.37 Test bitstreams #PDIS_E

Specification: All slices are coded as I, P, and B slices. Each frame contains only one slice. The distance between two neighbouring P frames is set to 128.

Functional stage: Test the decoding process of motion vector derivation for B_SKIP and Pred_Sym modes.

Purpose: Check if the decoder can properly handle the decoding process of slices with B_SKIP and Pred_Sym modes.

13.3 Normative test suites for ISO/IEC 14496-33

Table AMD46-1 — Conformance test categories

Categories	Bitstream	Donated by	File name
Transform	TRANS_A	Peking University	TRANS_A
	TRANS_B	Peking University	TRANS_B
	TRANS_C	Peking University	TRANS_C
Quantization	MULQP_A	Peking University	MULQP_A
	MULQP_B	Peking University	MULQP_B
	MULQP_C	Peking University	MULQP_C
Multiple slices	MULSLICE_A	Peking University	MULSLICE_A
	MULSLICE_B	Peking University	MULSLICE_B
	MULSLICE_C	Peking University	MULSLICE_C
Deblocking	DBLK_A	Peking University	DBLK_A
	DBLK_B	Peking University	DBLK_B
	DBLK_C	Peking University	DBLK_C
	DBLK_D	Peking University	DBLK_D
	DBLK_E	Peking University	DBLK_E
	DBLK_F	Peking University	DBLK_F
Interpolation	INFTYPE_A	Peking University	INFTYPE_A
	INFTYPE_B	Peking University	INFTYPE_B
	INFTYPE_C	Peking University	INFTYPE_C
Non-reference P	NONREFP_A	Peking University	NONREFP_A
	NONREFP_B	Peking University	NONREFP_B

Categories	Bitstream	Donated by	File name
Motion vector	MERANGE_A	Peking University	MERANGE_A
	MERANGE_B	Peking University	MERANGE_B
Multiple-hypothesis	MULH_A	Peking University	MULH_A
	MULH_B	Peking University	MULH_B
MC block size	BLKSIZE_A	Peking University	BLKSIZE_A
	BLKSIZE_B	Peking University	BLKSIZE_B
	BLKSIZE_C	Peking University	BLKSIZE_C
	BLKSIZE_D	Peking University	BLKSIZE_D
	BLKSIZE_E	Peking University	BLKSIZE_E
	BLKSIZE_F	Peking University	BLKSIZE_F
Intra coding	IPRED_A	Peking University	IPRED_A
	IPRED_B	Peking University	IPRED_B
P frame distance	PDIS_A	Peking University	PDIS_A
	PDIS_B	Peking University	PDIS_B
	PDIS_C	Peking University	PDIS_C
	PDIS_D	Peking University	PDIS_D
	PDIS_E	Peking University	PDIS_E

Add the electronic attachment available at http://standards.iso.org/iso-iec/14496/-4/ed-2/amd/46/en to the electronic database of conformance test data.

ICS 35.040.40

Price based on 9 pages