INTERNATIONAL STANDARD

ISO/IEC 18477-3

First edition
2015-12-15

Information technology — Scalable compression and coding of continuous-tone still images —

Part 3: Box file format

Technologies de l'information — Compression échelonnable et codage d'images plates en ton continu —

Partie 3: Format de la liste de fichiers

Reference number
ISO/IEC 18477-3:2015(E)

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

Contents

Page

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.

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 on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 18477 contains the following parts under the general title *Information technology — Scalable compression and coding of continuous-tone still images*:

— *Part 1: Scalable compression and coding of continuous-tone still images*

— *Part 2: Extensions for high dynamic range images*

— *Part 3: Box file format*

— *Part 6: IDR Integer Coding*

— *Part 7: HDR Floating-Point Coding*

— *Part 8: Lossless and Near-lossless Coding*

— *Part 9: Alpha Channel Coding*

The following parts are under preparation:

— *Part 4: Conformance testing*

— *Part 5: Reference software*

Introduction

This part of ISO/IEC 18477 specifies an extensible file format, denoted as JPEG XT, which is built on top of the existing Rec. ITU-T T.81 | ISO/IEC 10918-1 codestream definition. While typically file formats encapsulate codestreams by means of additional syntax elements such as boxes, the file format structure specified here rather embeds the syntax elements of the file format, called boxes, into the codestream. The necessity for this unusual arrangement is the backwards compatibility to the legacy standard and the application toolchain built around it; that is, legacy applications conforming to Rec. ITU-T T.81 | ISO/IEC 10918-1 will be able to decode image information embedded in files conforming to the family of ISO/IEC 18477 standards, though will only be able to recover a three component, 8 bits per sample, lower quality version of the image described by the full file.

For more demanding applications, it is not uncommon to use a bit depth of 16, providing 65 536 representable values to describe each channel within a pixel, resulting on over $2,8 \times 10^{14}$ representable colour values. In some less common scenarios, even greater bit depths are used, and sometimes the dynamic range of the image is so high that a floating point based encoding is desirable. In addition to image information, some applications also require an additional opacity channel, a feature not available from the legacy standard.

Most common photo and image formats use an 8-bit or 16-bit unsigned integer value to represent some function of the intensity of each colour channel. While it might be theoretically possible to agree on one method for assigning specific numerical values to real world colours, doing so is not practical. Since any specific device has its own limited range for colour reproduction, the device's range may be a small portion of the agreed-upon universal colour range. As a result, such an approach is an extremely inefficient use of the available numerical values, especially when using only 8 bits (or 256 unique values) per channel. To represent pixel values as efficiently as possible, devices use a numeric encoding optimized for their own range of possible colours or gamut.

JPEG XT is designed to extend the legacy JPEG standard towards higher bitdepth, higher dynamic range, wide colour gamut content while simultaneously allowing legacy applications to decode the image data in the codestream to a standard low dynamic range image represented by only eight bits per channel. The goal is to provide a backwards compatible coding specification that allows legacy applications and existing toolchains to continue to operate on codestreams conforming to the family of ISO/IEC 18477 standards.

JPEG XT has been designed to be backwards compatible to legacy applications while at the same time having a small coding complexity; JPEG XT uses, whenever possible, functional blocks of Rec. ITU-T T.81 | ISO/IEC 10918-1 to extend the functionality of the legacy JPEG Coding System.

This part of ISO/IEC 18477 is an extension of ISO/IEC 18477-1, a compression system for continuous tone digital still images which is backwards compatible with Rec. ITU-T T.81 | ISO/IEC 10918-1. That is, legacy applications conforming to Rec. ITU-T T.81 | ISO/IEC 10918-1 will be able to reconstruct streams generated by an encoder conforming to this part of ISO/IEC 18477, though will possibly not be able to reconstruct such streams in full dynamic range, full quality or other features defined in this part of ISO/IEC 18477.

The aim of this part of ISO/IEC 18477 is to provide a flexible and extensible framework to enrich ISO/IEC 18477-1 compliant codestreams with side-channels and metadata. The syntax chosen in this part of ISO/IEC 18477 defines a mechanism to embed syntax elements denoted as "Boxes" into Rec. ITU-T T.81 | ISO/IEC 10918-1 compliant codestreams. The box syntax used here is identical to that defined in the JPEG family of standards, for example JPEG 2000 (Rec. ITU-T T. 800 | ISO/IEC 15444-1). Boxes will then carry either additional image data, to enable encoding of images of higher bitdepth, high dynamic range, include alpha channels etc., or will carry metadata that describes the decoding process of the legacy Rec. ITU-T T.81 | ISO/IEC 10918-1 codestream and the side channels to an extended or high dynamic range image.

Information technology — Scalable compression and coding of continuous-tone still images —

Part 3: Box file format

1 Scope

This part of ISO/IEC 18477 specifies a coding format, referred to as JPEG XT, which is designed primarily for continuous-tone photographic content.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

ISO/IEC 18477-1:2015, *Information technology — Scalable compression and coding of continuous-tone still images — Part 1: Scalable compression and coding of continuous-tone still images*

Rec. ITU-T T.81 | ISO/IEC 10918-1, *Information Technology — Digital Compression and Coding of Continuous Tone Still Images – Requirements and Guidelines*

Rec. ITU-T T.871 | ISO/IEC 10918-5, *Information technology — Digital compression and coding of continuous-tone still images: JPEG File Interchange Format*

3 Terms, definitions, abbreviated terms and symbols

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
ASCII encoding
encoding of text characters and text strings according to ISO/IEC 10646

3.1.2
base decoding path
process of decoding legacy codestream and refinement data to the base image, jointly with all further steps until residual data is added to the values obtained from the residual codestream

3.1.3
base image
collection of sample values obtained by entropy decoding the DCT coefficients of the legacy codestream and the refinement codestream, and inversely DCT transforming them jointly

3.1.4
binary decision
choice between two alternatives

3.1.5
bit stream
partially encoded or decoded sequence of bits comprising an entropy-coded segment

3.1.6
block
8 × 8 array of samples or an 8 × 8 array of DCT coefficient values of one component

3.1.7
box
structured collection of data describing the image or the image decoding process embedded into one or multiple APP_{11} marker segments

Note 1 to entry: See Annex A for the definition of boxes

3.1.8
byte
group of 8 bits

3.1.9
coder
embodiment of a coding process

3.1.10
coding
encoding or decoding

3.1.11
coding model
procedure used to convert input data into symbols to be coded

3.1.12
(coding) process
general term for referring to an encoding process, a decoding process, or both

3.1.13
compression
reduction in the number of bits used to represent source image data

3.1.14
component
two-dimensional array of samples having the same designation in the output or display device. An image typically consists of several components, e. g. red, green and blue

3.1.15
continuous tone image
image whose components have more than one bit per sample

3.1.16
decoder
embodiment of a decoding process

3.1.17
decoding process
process which takes as its input compressed image data and outputs a continuous-tone image

3.1.18
encoder
embodiment of an encoding process

3.1.19
encoding process
process which takes as its input a continuous-tone image and outputs compressed image data

3.1.20
entropy decoder
embodiment of an entropy decoding procedure

3.1.21
entropy decoding
lossless procedure which recovers the sequence of symbols from the sequence of bits produced by the entropy encoder

3.1.22
entropy encoder
embodiment of an entropy encoding procedure

3.1.23
entropy encoding
lossless procedure which converts a sequence of input symbols into a sequence of bits such that the average number of bits per symbol approaches the entropy of the input symbols

3.1.24
high dynamic range
image or image data comprised of more than eight bits per sample

3.1.25
Intermediate dynamic range
image or image data comprised of more than eight bits per sample

3.1.26
Joint Photographic Experts Group
JPEG
informal name of the committee which created this part of ISO/IEC 18477

Note 1 to entry: The "joint" comes from the ITU and ISO/IEC collaboration.

3.1.27
legacy codestream
collection of markers and syntax elements defined by Rec. ITU-T T.81 | ISO/IEC 10918-1 bare any syntax elements defined by the family ISO/IEC 18477 standards

Note 1 to entry: That is, the legacy codestream consists of the collection of all markers except those APP_{11} markers that describe JPEG XT boxes by the syntax defined in Annex A.

3.1.28
legacy decoding path
collection of operations to be performed on the entropy coded data as described by Rec. ITU-T T.81 | ISO/IEC 10918-1 jointly with the Legacy Refinement scans before this data is merged with the residual data to form the final output image

3.1.29
legacy decoder
embodiment of a decoding process conforming to ITU. T Rec. T.81 | ISO/IEC 10918-1, confined to the lossy DCT process and the baseline, sequential or progressive modes, decoding at most four components to eight bits per component

3.1.30
legacy image
arrangement of sample values as described by applying the decoding process described by Rec. ITU-T T.81 | ISO/IEC 10918-1 on the entropy coded data as defined by said standard

3.1.31
lossless
descriptive term for encoding and decoding processes and procedures in which the output of the decoding procedure(s) is identical to the input to the encoding procedure(s)

3.1.32
lossy
descriptive term for encoding and decoding processes which are not lossless

3.1.33
low dynamic range
image or image data comprised of data with no more than eight bits per sample

3.1.34
marker
two-byte code in which the first byte is hexadecimal FF and the second byte is a value between 1 and hexadecimal FE

3.1.35
marker segment
marker together with its associated set of parameters

3.1.36
pixel
collection of sample values in the spatial image domain having all the same sample coordinates, e. g. a pixel may consist of three samples describing its red, green and blue value

3.1.37
point transform
scaling of a sample or DCT coefficient by a factor

3.1.38
precision
number of bits allocated to a particular sample or DCT coefficient

3.1.39
procedure
set of steps which accomplishes one of the tasks which comprise an encoding or decoding process

3.1.40
residual decoding path
collection of operations applied to the entropy coded data contained in the residual data box and residual refinement scan boxes up to the point where this data is merged with the base image to form the final output image

3.1.41
residual image
extension image
sample values as reconstructed by inverse quantization and inverse DCT transformation applied to the entropy-decoded coefficients described by the residual scan and residual refinement scans

3.1.42
residual scan
additional pass over the image data invisible to legacy decoders which provides additive and/or multiplicative correction data of the base scans to allow reproduction of high dynamic range or wide colour gamut data

3.1.43
refinement scan
additional pass over the image data invisible to legacy decoders which provides additional least significant bits to extend the precision of the DCT transformed coefficients. Refinement scans can be either applied in the base or residual decoding path

3.1.44
sample
one element in the two-dimensional image array which comprises a component

3.1.45
sample grid
common coordinate system for all samples of an image

Note 1 to entry: The samples at the top left edge of the image have the coordinates (0, 0), the first coordinate increases towards the right, the second towards the bottom.

3.1.46
superbox
box that carries other boxes as payload data

3.1.47
zero byte
0x00 byte

3.1.48
zig-zag sequence
specific sequential ordering of the DCT coefficients from (approximately) lowest spatial frequency to highest

3.2 Symbols

X width of the sample grid in positions

Y height of the sample grid in positions

Nf number of components in an image

$s_{i,x}$ subsampling factor of component i in horizontal direction

$s_{i,y}$ subsampling factor of component i in vertical direction

H_i subsampling indicator of component i in the frame header

V_i subsampling indicator of component i in the frame header

$v_{x,y}$ sample value at the sample grid position x, y

R_h additional number of DCT coefficient bits represented by refinement scans in the base decoding path, $8+R_h$ is the number of non-fractional bits (i. e. bits in front of the “binary dot”) of the output of the inverse DCT process in the base decoding path

R_r additional number of DCT coefficient bits represented by refinement scans in the residual decoding path. $P+R_r$ is the number of non-fractional bits of the output of the invers DCT process in the residual decoding path, where P is the frame-precision of the residual image as recorded in the frame header of the residual codestream

R_b additional bits in the HDR image. 8+Rb is the sample precision of the reconstructed HDR image

3.3 Abbreviated terms

For the purposes of this part of ISO/IEC 18477, the following abbreviations apply.

ASCII American Standard Code for Information Interchange

LSB Least Significant Bit

MSB Most Significant Bit

HDR High Dynamic Range

IDR Intermediate Dynamic Range

LDR Low Dynamic Range

TMO Tone Mapping Operator

DCT Discrete Cosine Transformation

4 Conventions

4.1 Conformance language

This part of ISO/IEC 18477 consists of normative and informative text.

Normative text is that text which expresses mandatory requirements. The word "shall" is used to express mandatory requirements strictly to be followed in order to conform to this part of ISO/IEC 18477 and from which no deviation is permitted. A conforming implementation is one that fulfils all mandatory requirements.

Informative text is text that is potentially helpful to the user, but not indispensable and can be removed, changed or added editorially without affecting interoperability. All text in this part of ISO/IEC 18477 is normative, with the following exceptions: the Introduction, any parts of the text that are explicitly labelled as "informative", and statements appearing with the preamble "NOTE" and behaviour described using the word "should". The word "should" is used to describe behaviour that is encouraged but is not required for conformance to this part of ISO/IEC 18477.

The keywords "may" and "need not" indicate a course of action that is permissible in a conforming implementation.

The keyword "reserved" indicates a provision that is not specified at this time, shall not be used, and may be specified in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be specified in the future.

4.2 Operators

NOTE Many of the operators used in this part of ISO/IEC 18477 are similar to those used in the C programming language.

4.2.1 Arithmetic operators

+	Addition
−	Subtraction (as a binary operator) or negation (as a unary prefix operator)
*	Multiplication
/	Division without truncation or rounding.

umod x umod a is the unique value y between 0 and a-1 for which y+Na = x with a suitable integer N.

4.2.2 Logical operators

|| Logical OR

&& Logical AND

! Logical NOT

∈ x ∈ {A, B} is defined as (x == A || x == B)

∉ x ∉ {A, B} is defined as (x != A && x != B)

4.2.3 Relational operators

\> Greater than

\>= Greater than or equal to

< Less than

<= Less than or equal to

== Equal to

!= Not equal to

4.2.4 Precedence order of operators

Operators are listed below in descending order of precedence. If several operators appear in the same line, they have equal precedence. When several operators of equal precedence appear at the same level in an expression, evaluation proceeds according to the associativity of the operator either from right to left or from left to right.

Operators	Type of operation	Associativity
(), [ ], .	Expression	Left to Right
–	Unary negation	
*, /	Multiplication	Left to Right
umod	Modulo (remainder)	Left to Right
+, –	Addition and Subtraction	Left to Right
< , >, <=, >=	Relational	Left to Right

4.2.5 Mathematical functions

$\lceil x \rceil$ Ceil of x. Returns the smallest integer that is greater than or equal to x.

$\lfloor x \rfloor$ Floor of x. Returns the largest integer that is lesser than or equal to x.

|x|	Absolute value, is –x for x < 0, otherwise x.
sign(x)	Sign of x, 0 if x is 0, +1 if x is positive, -1 if x is negative.
clamp(x,min,max)	Clamps x to the range [min, max]: returns min if x < min, max if x > max or otherwise x.
x^a	Raises the value of x to the power of a. x is a non-negative real number, a is a real number. x^a is equal to exp(a×log(x)) where exp is the exponential function and log() the natural logarithm. If x is 0 and a is positive, x^a is defined to be 0.

5 General

The purpose of this Clause is to give an informative overview of the elements specified in this part of ISO/IEC 18477. Another purpose is to introduce many of the terms which are defined in Clause 3. These terms are printed in *italics* upon first usage in this Clause.

There are three elements specified in this part of ISO/IEC 18477:

a) An *encoder* is an embodiment of an *encoding process*. An encoder takes as input *digital source image data* and *encoder specifications*, and by means of a specified set of *procedures* generates as output *a codestream*.

b) A *decoder* is an embodiment of a *decoding process*. A decoder takes as input a codestream, and by means of a specified set of procedures generates as output *digital reconstructed image data*.

c) The *codestream* is a compressed image data representation which includes all necessary data to allow a (full or approximate) reconstruction of the sample values of a digital image. Additional data might be required that define the interpretation of the sample data, such as colour space or the spatial dimensions of the samples.

5.1 High level overview on JPEG XT ISO/IEC 18477-3

The high-level syntax of an ISO/IEC 18477-3 compliant codestream is identical to that defined in ISO/IEC 18477-1, which is a subset of the syntax defined in Rec. ITU-T T.81 | ISO/IEC 10918-1. Marker definitions and the syntax of the markers defined in the above Recommendation remain in force and unchanged. However, this part of ISO/IEC 18477 defines the APP_{11} marker, reserved in the legacy Recommendation | Standard for encoding additional syntax elements. Legacy decoders will skip and ignore such marker elements, and hence will only be able to decode the image encoded by the legacy syntax elements. This part of ISO/IEC 18477 codestream will be denoted the **legacy codestream** in the following.

This part of ISO/IEC 18477 extends the legacy standard by a syntax element called "Box", using the APP_{11} marker to hide the extended syntax elements from legacy applications. Boxes and their encoding are specified in Annex A. A common set of boxes used by all subsequent parts of the family of ISO/IEC 18477 standards are defined in Annex B. A box may either include additional metadata required to decode the complete codestream to full precision, full dynamic range or without loss, or may contain entropy coded image data itself.

How entropy coded data from the side-channels contained in the boxes and entropy coded data in the legacy codestream are merged together is application dependent and defined in subsequent parts of the ISO/IEC 18477 standards family. It is beyond the scope of this part of ISO/IEC 18477 to define this process.

5.2 Encoder requirements

An encoder is only required to meet the compliance tests and to generate the codestream according to the syntax defined in this part of ISO/IEC 18477. How the codestream is algorithmically constructed and how the boxes are laid out is implementation specific and not within scope of this part of ISO/IEC 18477.

Subsequent Recommendations | Standards of the ISO/IEC 18477 family may, however, define additional restrictions and requirements, either within the standard itself, or within profiles that restrict the freedom of the encoder further.

An encoder claiming to be compliant to one of these profiles then shall conform to the syntax constraints defined in the corresponding profile of the corresponding part of ISO/IEC 18477.

5.3 Decoder requirements

A decoding process converts compressed image data to reconstructed image data. A decoder shall interpret the syntax of the box structures, namely the packaging of boxes into APP markers specified in Annex A correctly. It is **not required**, though, that a conforming decoder is capable of interpreting the semantics of all box types defined in this or subsequent members of the ISO/IEC 18477 family of standards. A decoder implementation should skip over boxes it is unable or not willing to support unless such a box is indicated as a mandatory box in the profile and part of ISO/IEC 18477 the decoder claims to be compliant with.

Annex A
(normative)

JPEG XT marker segment

A.1 General

This Annex extends the compressed bitstream syntax of ISO/IEC 18477-1:2015, Annex B by introducing additional markers and marker segments carrying side channel and coding parameters that control the decoding process. While the corresponding decoding processes are specified in subsequent parts of the ISO/IEC 18477 family of standards, this Annex defines a generic mechanism by which such syntax elements are embedded into ISO/IEC 18477-1 compliant files.

The syntax element and the building block defined in this Annex is called a Box. This part of ISO/IEC 18477 defined several types of boxes; the definition of each specific box type defines the kind of information that may be found within a box of that type. Some boxes will be defined to contain other boxes. Box types are specified in Annex B, or in subsequent members of the ISO/IEC 18477 family of standards.

Boxes are, unlike in other Recommendations | International Standards not a top-level syntax elements, but themselves wrapped in JPEG XT Marker segments introduced in A.2. Since boxes may logically carry more than 64K (65536) bytes of payload data, but marker segments can at most carry 64K of data, a single **logical box** may need to be broken up into several marker segments. Syntax elements within the marker segment then instruct the decoder how to put the contents in the marker segment back into a single box.

Additionally, a JPEG XT file may contain several boxes of the same box type, though with differing content. The syntax of the marker segment provides a mechanism to distinguish between two logically different boxes of the same box type.

A.2 Marker assignments

The following additional marker is defined in this part of ISO/IEC 18477:

Table A.1 — Additional markers and marker segments

Code Assignment	Symbol	Description	Defined in
0xFFEB	APP_{11}	JPEG XT Marker	This part of ISO/IEC 18477

Each box is encapsulated in at least one JPEG XT marker segment, and may extend over several marker segments if the size of its payload data exceeds the capacity of the JPEG XT marker. See A.4 for how to merge JPEG XT marker segments to logical boxes.

A.3 Codestream syntax

The high-level syntax of ISO/IEC 18477-3 codestreams shall follow the syntax specified in the ISO/IEC 18477-1 standard, which is a subset of Rec. ITU-T T.81 | ISO/IEC 10918-1. Specifically, since JPEG XT boxes are represented by APP_{11} marker segments, ISO/IEC 18477-1 conforming implementations that do not implement this or any subsequent Recommendation | International Standard of the ISO/IEC 18477 family will ignore them.

NOTE Note that by the above paragraph, byte stuffing and padding as defined in Rec. ITU-T T.81 | ISO/IEC 10918-1 also applies to entropy coded data contained in APP_{11} markers. Note further that due to the segmentation of entropy coded data into application markers, it may happen that the last byte of an APP_{11} marker segment is 0xff, and that the corresponding "stuffed" zero byte is part of a subsequent application marker segment. This does not cause a problem for legacy decoders since they are required to skip over unknown application marker segments in first place, without interpreting their content.

A.4 JPEG XT boxes

JPEG XT structures any additional data that remains invisible to legacy decoders in JPEG XT boxes. A **box** is a generic data container that has both a type, and a body that carries its actual payload. The type is a four-byte identifier that allows decoders to identify its purpose and the structure of its content. A JPEG XT file may also carry several boxes of identical type. These boxes are logically distinct and differ in the value of the **Box Instance Number En** of the JPEG Extensions marker segment, see Figure A.1.

Boxes are embedded into the codestream format by encapsulating them into one or several JPEG XT marker segments. Since boxes can grow large in size, a single box may extend over multiple JPEG XT marker segments, and decoders may have to merge multiple marker segments before they can attempt to decode the box content. JPEG XT Marker segments that belong to the same logical box and require merging prior to interpretation have **identical Box Instance Number fields En,** but differ in the **Packet Sequence Number Z.**

The JPEG XT marker segment consists of the APP_{11} marker that is reserved for this part of ISO/IEC 18477, the size of the marker segment in bytes (not including the marker), a common identifier identical for all boxes and box types, the box instance number field, the packet sequence number field, the box length, the box type and the actual box payload data. The box length field can be extended by a Box Length Extension field that allows box sizes beyond 2^{32}-1 bytes. Figure A.1 depicts the high-level syntax of a JPEG XT Marker segment.

0xFFEB	Le	CI Common Identifier	En Box Instance Number	Z Packet Sequence Number	LBox Box Length	TBox Box Type	*XLBox* *Box Length Extension (optional)*	Payload Data

Figure A.1 — Organization of the JPEG XT marker segment

The meaning of the fields of the JPEG XT Marker segment is as follows:

The Le field is the size of the marker segment, not including the marker. It measures the size from the Le field up to the end of the marker segment.

NOTE Since boxes may extend over several marker segments, the Le field is typically **not derived from** the Box Length field and care must be taken not to confuse the two. The Le field defines the amount of data carried by a single marker segment; the Box Length is the logical size of the box. If a box extends over multiple JPEG XT Extension marker segments, the Le field measures the total size of each individual marker segment and may differ from segment to segment, whereas the Box Length field remains identical in all segments that contribute to the same logical box.

The Common Identifier is a 16 bit field that allows decoders to identify an APP_{11} marker segment as a JPEG XT marker segment. Its value shall be 0x4A50. It is identical for all boxes and all box types.

The Box Instance Number is a 16-bit field that disambiguates between JPEG XT marker segments carrying boxes of identical type, but differing content. That is, data that belongs to logically distinct boxes with the same box type differ in their Box Instance Number. Encoders shall concatenate the payload data of those JPEG XT marker segments whose Box Instance Number **and** Type Identifier fields are identical in the order of increasing Packet Sequence Numbers.

NOTE A codestream containing multiple boxes of the same box type uses the box sequence number field to instruct the decoder which JPEG XT Extension marker segments to merge into one box. Refinement coding makes use of this process: The entropy coded data of each refinement scan is placed into its individual box, using the box instance number field to disambiguate the scans.

The Packet Sequence Number is a 32-bit field that specifies the order in which payload data shall be merged. Concatenation proceeds in the order of increasing Packet Sequence Numbers.

The **Box Length LBox** is a four byte field that specifies the box length. It measures the size of the payload data of all JPEG Extensions markers of the same box type and enumerator combined, plus the size of **a single copy** of the Box Type, plus the size of **a single copy** of the Box Length, plus the length of a **single copy** of the Box Length Extender if present. The box length does not include the size of the packet sequence number, the box instance number, the common identifier, the marker length or the marker.

NOTE A box having a payload data of 32 bytes will, by this, have a box length of 32+4+4 = 40. If this box is split evenly over two JPEG XT marker segments, each marker segment will have a Le value of 2+2+2+4+(4+4+16) = 50.

If the size of the box payload is less than 2^{32}-8 bytes, then all fields **except the XLBox field**, that is: Le, CI, En, Z, LBox and TBox, shall be present in **all JPEG XT marker** segment representing this box, regardless of whether the marker segments starts this box, or continues a box started by a former JPEG XT Marker segment.

The **Box Type TBox** is a 32-bit field that specifies the type of the payload data, and thus its syntax. Box types are specified in Annex B and in subsequent parts of the ISO/IEC 18477 family of standards. Since ITU | ISO/IEC may add additional box types that define additional meta-information on the image later, decoders shall disregard box types they do not understand.

If the box length is larger than 2^{32} bytes, the LBox field is no longer sufficient to encode the box length and the XLBox field is required additionally. In this case, the LBox field shall be one and the XLBox field carries the box size instead. If the box length is larger than 2^{32}, the XLBox field shall be present in all JPEG XT marker segments of the same box type and same box Instance Number, and its value shall be identical in all JPEG XTmarker segments of the same Box Type and same Box Instance Number.

The payload data carries the contents of the box. Its syntax is specified along with the corresponding box types in this Annex.

Profiles defined in subsequent parts of the ISO/IEC 18477 family of standards add additional constraints in how payload data may be broken up into individual JPEG XT marker segments.

Table A.2 — JPEG XT marker parameters and sizes

Parameter	Size (bits)	Value	Meaning
APP_{11}	16	0xFFEB	Identifies all JPEG XT Marker Segments.
Le	16	8..65535	Length of the marker segment, including the size itself, all parameters, and the size of the payload data contained in this marker segment alone. Does not include the marker itself.

Table A.2 *(continued)*

Parameter	Size (bits)	Value	Meaning
CI	16	0x4A50 (ASCII encoding of “JP”)	The special value 0x4A50 (ASCII: ‘J’ ‘P’) allows readers to distinguish the JPEG Extensions marker segment from other uses of the APP_{11} marker. Readers shall ignore APP_{11} markers for the purpose of decoding JPEG extensions if this value does not match.
En	16	1..65535	The Box Instance Number disambiguates payload data of the same box type and defines which payload data is to be concatenated. Only payload data whose box type and enumerator is identical shall be concatenated. The value 0 is reserved for ITU | ISO/IEC purposes.
Z	32	$1..2^{32}$-1	Packet Sequence number defining the order in which the payload data shall be concatenated. Concatenation shall proceed in order of increasing Z values. The value 0 is reserved for ITU | ISO/IEC purposes.
LBox	32	1 or $8..2^{32}$-1	Box length. This is the total length of the concatenated payload data, including a single copy of the LBox and Tbox field, and a single copy of the XLBox field, if present. The values 0 and two to seven are reserved for ITU | ISO/IEC purposes and shall not be used.
TBox	32	$0..2^{32}$-1	Box type. The box type defines the syntax of the concatenated payload data. Also, the box type **and** the box instance number specify which payload data to merge.
XLBox	0 or 64	$16..2^{64}$-1	If the LBox field is one, this field contains the size of the concatenated payload data plus the box overhead instead. Otherwise, this field is missing. The values 0 to 15 are reserved for ITU | ISO/IEC purposes.

Table A.2 *(continued)*

Parameter	Size (bits)	Value	Meaning
Payload Data	Varies	Varies	The syntax of the concatenated payload data is defined in Annex B and subsequent members of the ISO/IEC 18477 family of standards.

NOTE The size of the XLBox field itself also contributes to the box length, hence creating a corner case for boxes larger than 4GB. If an encoder detects that the value of the LBox field, computed as the sum of the payload data size and the box overhead, overruns the 4GB boundary LBox is able to express, it is not sufficient to create an XLBox field and store the sum there. The box size needs to be enlarged by the size of the XLBox field as well, namely by eight bytes.

A.5 Boxes and superboxes

Some boxes may carry other boxes as payload data. Such boxes are denoted as superboxes. The payload size of a superbox is given by the sum of the box lengths of all the boxes it contains.

Boxes **within** superboxes **do not** consist of JPEG XT marker, neither a marker size, neither a Common Identifier, neither an Box Instance Number nor a Packet Sequence Number shall be present. They start with the LBox field. The additional fields are not required since their composition from markers into boxes is unambiguous.

NOTE The length of a box within a superbox is derived in the same way from the size of the payload data as for top-level boxes within JPEG XT marker segments. Note that neither top-level boxes nor boxes within superboxes count the Le, En and Z fields as part of their length. Note further that a box within a superbox may be a superbox again and may contain further boxes. The layout of such boxes is given by Figure A.2, too.

LBox Box Length	TBox Box Type	*XLBox* *Box Length* *Extension* *(optional)*	Payload Data

Figure A.2 — Organization of a box within a superbox

Annex B
(normative)

Common box types

B.1 General

This Annex defines box types that are common for all parts of the ISO/IEC 18477 family of standards and shared amongst them. Subsequent parts reference to this Annex as required.

B.2 Integer Table Lookup box

This box shall appear at the top level of the file, it shall not be a subbox of any superbox. This box defines a lookup process at the decoder and may be used to implement a point transformation as used in the base or range mapping operations which are part of the merging process for combining the low dynamic range and residual image information to reconstruct a high dynamic range image. This table carries integer data of up to 16 bit precision, and is indexed by integer values.

There shall be at most one Integer or Floating Point Table Lookup or Parametric Curve box for each value of M within the same superbox or within the codestream at top level.

The type of this box shall be 0x544f4e45, ASCII encoding of "TONE".

The box organization is defined in Figure B.1, the parameters and sizes in Table B.1.

M	R_d	D_k

Figure B.1 — Organization of the Integer Table Lookup box

Table B.1 — Integer Table Lookup box, parameters and sizes

Parameter	Size (in Bits)	Value	Meaning
M	4	0..15	Table destination. Up to 16 tables can be defined.
R_d	4	0..9	Specifies the output precision of the table entries. Table entries shall be integers in the range [0, 2^{8+R_d}-1]. R_d shall fit to the application of the table, see Annex C.

Table B.1 *(continued)*

Parameter	Size (in Bits)	Value	Meaning
D_k	16 or 32	$0..2^{Rt+8}$-1	Table contents. For $R_b \leq 8$, each entry is a 16 bit unsigned integer. For R_b>8, each entry is a 32 bit unsigned integer. All numbers shall be represented in big-endian format. The number of table entries can be derived from the box length. The size of the table shall be a power of two that fits to the application of the table, see Annex C for details.

The value of R_t defines the output range of the table lookup process in bits. The length of the table depends on the input precision required for the application of the table. The number of table entries shall be 2^l where l is an application dependent integer.

B.3 Floating Point Table Lookup box

This box shall appear at the top level of the file, it shall not be a subbox of any superbox. This box defines a lookup process at the decoder and may be used to implement a point transformation as used in the base or range mapping operations which are part of the merging process for combining the low dynamic range and residual image information to reconstruct a high dynamic range image. This table carries floating point data of 32 bit precision, and is indexed by integer values.

There shall be at most one Integer or Floating Point Table Lookup or Parametric Curve box for each value of M within the same superbox or within the codestream at top level.

The type of this box shall be 0x46544f4e, ASCII encoding of "FTON".

The box organization is defined in Figure B.2, the parameters and sizes in Table B.2.

M	R_d	D_k

Figure B.2 — Organization of the Floating Point Table Lookup box

Table B.2 — Floating Table Lookup box, parameters and sizes

Parameter	Size (in Bits)	Value	Meaning
M	4	0..15	Table destination. Up to 16 tables can be defined.
R_d	4	0	Reserved for ITU | ISO/IEC purposes.

Table B.2 *(continued)*

Parameter	Size (in Bits)	Value	Meaning
D_k	32	non-negative numbers	Table contents. Each entry is a ISO/IEC/IEEE 60559 big endian single precision 32 bit floating point number. The number of table entries can be derived from the box length. The size of the table shall be a power of two that fits to the application of the table. See Annex C for details.

The length of the table depends on the input precision required for the application of the table. The number of table entries shall be 2^l where l is an application dependent integer.

B.4 Parametric Curve box

This box shall either appear at the top level of the file, or it shall be a sub-box of the Merging Specification box or Alpha Merging Specification box. A Parametric Curve Box as a subbox shall replace a Parametric Curve box at the top level of the file with the same value of M.

This box defines a decoder mapping process in the form of a parameterized curve which might offer a more efficient coding than a look-up table. The parametric curve box maps integer, fixed-point or floating-point numbers to floating point numbers.

There shall be at most one Integer or Floating Point Table Lookup or Parametric Curve box for each value of M within the same superbox or within the codestream at top level.

The box type of this box shall be 0x43555256, the ASCII encoding of “CURV”.

The box organization is defined in Figure B.3, the parameters and sizes in Table B.3.

M	t	e	r	P_1	P_2	P_3	P_4

Figure B.3 — Organization of the Parametric Curve box

Table B.3 — Contents of the Parametric Curve box

Parameter	Size (in Bits)	Value	Meaning
M	4	0..15	Table destination. Up to 16 curves can be defined.
T	4	0..15	Curve type. Curve types are defined in Table B.4.
e	4	0..1	Rounding Mode, see C.1 for details.
R	4	0	Reserved for ITU | ISO/IEC purposes. This field shall always be 0.
P_1	32	Depends on t	Curve parameter 1, encoded as big-endian ISO/IEC/IEEE 60559 single precision floating point value

Table B.3 *(continued)*

Parameter	Size (in Bits)	Value	Meaning
P_2	32	Depends on t	Curve parameter 2, encoded as big-endian ISO/IEC/IEEE 60559 single precision floating point value
P_3	32	Depends on t	Curve parameter 3, encoded as big-endian ISO/IEC/IEEE 60559 single precision floating point value
P_4	32	Depends on t	Curve parameter 4, encoded as big-endian ISO/IEC/IEEE 60559 single precision floating point value

The parametric curve to be applied is defined by the t parameter. Depending on t, the parameters P_1 through P_4 further specify the curve. Note that all parameters are always present, regardless of the actual curve type. They are, however, eventually ignored. Table B.4 specifies the available parametric curve types.

Parametric curves are applied in four steps to map an input value to an output value:

-- First, the input value *x* is normalized to the input scale of the channel as specified in Annex C using a procedure that depends on the e parameter. The result of this process is a scaled input value *y*.

-- Second, the parameter t is used to select one out of several curve types. The value of *y* and the curve parameters P_1 through P_4 compute then from the scaled input *y* a scaled output *f(y)*.

-- Third, the output *f(y)* is again scaled by a suitable factor depending on e to match the output range of the channel, where scaling is defined in Annex C again.

Table B.4 — Predefined parametric curves

Curve Type, Value of t	Inverse tone mapping to be performed	Remarks
0, 1		Reserved for ITU | ISO/IEC purposes.
2	$f(x) = x$	Identity transformation, input data is passed through, though scaling of input and output applies. P_1 through P_4 are ignored.
3		Reserved for ITU | ISO/IEC purposes
4	$f(x) = ((x+P_3)/(1+P_3))^{P_2}$ for $x \geq P_1$ and $f(x) = ((P_1+P_3)/(1+P_3))^{P_2} \times x/P_1$ for $x < P_1$	This is an inverse gamma mapping with parameters P_1 through P_3. For P_1=0. 04045, P_2=2. 4, P_3=0. 055, this is the sRGB nonlinearity.
5	$f(x) = x \times (P_2 - P_1) + P_1$	Linear ramp with start value P_1 and end value P_2. P_2 shall be larger than P_1. P_3 and P_4 are ignored.
6	$f(x) = P_3 \times \exp(x \times (P_2 - P_1) + P_1) + P_4$	Exponential map from a linear ramp.

Table B.4 *(continued)*

Curve Type, Value of t	Inverse tone mapping to be performed	Remarks
7	$f(x) = sign(P_1)\times log((\|P_1\|\times x)^{P2}+P_3)+P_4$	Logarithmic map. The parameters P_2 through P_3 shall be positive.
8	$f(x) = (P_2-P_1) \times x^{P3} + P_1$	Inverse gamma mapping without toe region followed by scaling to the range $[P_1, P_2]$.
9..15		Reserved for ITU | ISO/IEC purposes.

B.5 Fix-point Linear Transformation box

This box shall either appear at the top level of the file, or it shall be a sup-box of the Merging Specification box or Alpha Merging Specification box. A Fix-point Linear Transformation Box as a subbox shall replace a Fix-point Linear Transformation Box at the top level of the file with the same value of M.

There shall be at most one Floating-point Linear transformation box or Integer Linear Transformation box for each value of M within the same superbox, or within JPEG XT markers at top level of the codestream.

This box defines a free-form table based linear transformation. This box defines nine parameters which specify the entries in a 3×3 matrix:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Up to eleven free form transformations can be defined; its entries are fixed-point numbers that are preshifted by 13 bits.

The type of this box shall be 0x4D545258, ASCII encoding of "MTRX".

Figure B.4 describes the organization of this box segment, Table B.5 the parameters and parameter sizes.

M	t	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}

Figure B.4 — Organization of the Fixed-Point Linear Transformation box

Table B.5 — Fixed-Point Linear Transformation box contents

Parameter	Size (in bits)	Value	Meaning
M	4	5..15	Table Destination. Up to eleven free form linear transformations can be defined.
t	4	13	Reserved for ITU | ISO/IEC purposes.

Table B.5 *(continued)*

Parameter	Size (in bits)	Value	Meaning
a_{11}	16	-32768 to 32767	First parameter of the 3×3 transformation. This and all following parameters are represented by big-endian 16 bit integers
a_{12}	16	-32768 to 32767	Second parameter
a_{13} through a_{33}			All other parameters in the same format.

NOTE Due to scaling that is applied, the **integer** coefficients of the transformation matrix for t=13 can also be understood as fix point numbers which are pre-shifted by 13 bits.

B.6 Floating-point Linear Transformation box

This box shall either appear at the top level of the file, or it shall be a sup-box of the Merging Specification box or Alpha Merging Specification box. A Floating-point Linear Transformation Box as a subbox shall replace a Floating-point box at the top level of the file with the same value of M.

There shall be at most one Floating-point Linear transformation box or Integer Linear Transformation box for each value of M within the same superbox, or within JPEG XT markers at top level of the codestream.

This box defines a free-form table based linear transformation. This box defines nine parameters which specify the entries in a 3×3 matrix:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Up to eleven free form transformations can be defined; its entries are floating-point numbers encoded in ISO/IEC/IEEE 60559 single precision, big endian representation.

The type of this box shall be 0x46545258, ASCII encoding of “FTRX”.

Figure B.5 describes the organization of this box segment, Table B.5 the parameters and parameter sizes.

M	t	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}

Figure B.5 — Organization of the Floating Point Linear Transformation box

Table B.6 — Floating Point Linear Transformation box contents

Parameter	Size (in bits)	Value	Meaning
M	4	5..15	Table Destination. Up to eleven free form linear transformations can be defined.

Table B.6 *(continued)*

Parameter	Size (in bits)	Value	Meaning
t	4	0	Reserved for ITU | ISO/IEC purposes.
a_{11}	32	arbitrary	First parameter of the 3×3 transformation encoded as 32 bit single precision big-endian ISO/IEC/ IEEE 60559 number.
a_{12}	32	arbitrary	Second parameter
a_{13} through a_{33}			All other parameters in the same format.

B.7 Legacy Data Checksum box

This box keeps a checksum over the legacy codestream content, i.e. all data except those contained in application markers (APP_{00} through APP_{15}) using the algorithm defined in Annex D. Decoders may use this data to test for the integrity of the legacy codestream, and thus to test whether the base image data has been tampered with. A JPEG XT file shall contain at most one Legacy Data Checksum Box.

NOTE If a decoder detects that the checksum over the received data differs from the checksum recorded in this box, it may either abort decoding, or may only decode the base image and reject the residual stream or may attempt a full decoding at its discretion.

The type of this box shall be 0x4C43484B, ASCII encoding of “LCHK”.

Figure B.6 describes the organization of this box segment, Table B.7 the parameters and parameter sizes.

Chk

Figure B.6 — Organization of the Legacy Data Checksum box

Table B.7 — Legacy Data Checksum Box, Parameters and Sizes

Parameter	Size (bits)	Value	Meaning
Chk	32	Varies	Checksum over the legacy codestream, to be computed by the algorithm defined in Annex D.

B.8 File Type box

This box shall appear as first box in the first JPEG XT marker segment of a JPEG XT codestream. It shall not be a subbox of any other box and it shall not be broken up into multiple JPEG XT marker segments

The File Type box specifies the Recommendation | International Standard which completely defines all of the contents of this file, as well as a separate list of readers, defined by other Recommendations | International Standards, with which this file is compatible, and thus the file can properly interpreted within the scope of that other standard. This differentiates between the standard which completely describes the file, from other standards that interpret a subset of the file.

All files shall contain one and only one File Type box at top level of the file, and this box shall be the first box in the file.

The type of the File Type box shall be 0x66747970, ASCII encoding of “ftyp”. The organization of this box is depicted in Figure B.7, its parameters and fields in Table B.8.

BR	MinV	CL_0	...	CL_{N-1}

Figure B.7 — Organization of the File Type box

Table B.8 — File Type box, parameters and sizes

Parameter	Size (bits)	Value	Meaning
BR	32	0x6a707874 ASCII value of “jpxt”	This field specifies the Recommendation | International Standard which completely defines this file. This field is specified by a four byte ASCII string. The value of this field shall be “jpxt”, indicating that the file is compliant to JPEG XT.
MinV	32	0	Minor Version. This parameter defines the minor version number of this JPEG XT specification for which the file complies. The parameter is defined as a 4 byte big endian unsigned integer. The value of this field shall be 0. However, readers shall continue to parse and interpret this file even if the value of this field is nonzero.
CL_i	32	varies	Compatibility list. This field specifies a code representing this part of ISO/IEC 18477, another standard, or a profile of another standard, to which this file conforms. This field is encoded as a four byte string of ASCII characters. A file that conforms to this part of ISO/IEC 18477 shall have at least one CL_i field in the File Type box, and shall contain the value “jpxt” in one of the CL_i fields. All conforming readers shall properly interpret all files with “jpxt” in one of the CL_i fields. Other values of the Compatibility list fields are reserved for ITU | ISO/IEC use. The number of CL_i fields is determined by the length of this box.

B.9 Sample Unit box

This optional box shall appear on the top level of the JPEG XT codestream. If it is exists, it defines the unit of the sample values, i. e. a conversion factor from sample values to absolute radiance as given in cd/m^2 (nits). If this box is present, sample values represent absolute radiance values, otherwise only relative radiance is encoded. If this box is present, the absolute radiance L_a of a sample in candela per square metre is given by

$$L_a = \text{nits} \times L_r$$

where L_r is the decoded sample value and *nits* is the parameter in the Sample Unit box.

The type of the Sample Unit box shall be 0x554E4954, ASCII encoding of "UNIT". The organization of this box is depicted in Figure B.8, its parameters and fields in Table B.9.

nits

Figure B.8 — Organization of the Sample Unit box

Table B.9 — Sample Unit box, parameters and sizes

Parameter	Size (bits)	Value	Meaning
nits	32	>0	This multiplier converts the sample values into an absolute radiance, measured in cd/m^2 (nits) and hence defines the unit in which the sample values are represented in the codestream. This field is encoded as a 32-bit big-endian 32-bit ISO/IEC/IEEE 60559 number.

B.10 Primary Colour Coordinate box

This optional box shall appear on the top level of the JPEG XT codestream. If it is exists, it defines the x and y coordinates in the XYZ colour space of the primary colours encoded by the samples. This box shall only exist if the number of components Nf encoded in the codestream is 3. If this box does not exist, the colour coordinates of decoded sample values is defined by other means not specified by this part of ISO/IEC 18477.

The type of the Primary Colour Coordinate box shall be 0x50434f43, ASCII encoding of "PCOC". The organization of this box is depicted in Figure B.9, its parameters and fields in Table B.10.

x_0	y_0	x_1	y_1	x_2	y_2

Figure B.9 — Organization of the Sample Unit box

Table B.10 — Primary Colour Parameter box, parameters and sizes

Parameter	Size (bits)	Value	Meaning
x_0	32	≥0	This parameter specifies the x chroma coordinate in the XYZ colour space of the first component. It is encoded as a 32-bit big-endian ISO/IEC/IEEE 60559 floating point number.
y_1	32	≥0	This parameter specifies the y chroma coordinate in the XYZ colour space of the first component. It is encoded as a 32-bit big-endian ISO/IEC/IEEE 60559 floating point number. The z coordinate of the first colour primary is given as $z_0=1-x_0-y_0$
x_1	32	≥0	This parameter specifies the x chroma coordinate in the XYZ colour space of the second component. It is encoded as a 32-bit big-endian ISO/IEC/IEEE 60559 floating point number.
y_2	32	≥0	This parameter specifies the y chroma coordinate in the XYZ colour space of the second component. It is encoded as a 32-bit big-endian ISO/IEC/IEEE 60559 floating point number.
x_2	32	≥0	This parameter specifies the x chroma coordinate in the XYZ colour space of the third component. It is encoded as a 32-bit big-endian ISO/IEC/IEEE 60559 floating point number.
y_2	32	≥0	This parameter specifies the y chroma coordinate in the XYZ colour space of the third component. It is encoded as a 32-bit big-endian ISO/IEC/IEEE 60559 floating point number.

B.11 Residual Data box

This box encapsulates entropy coded segments extending the base image in the spatial domain. The process for merging the base image with the extension image contained in the residual data box is defined by the Merging Specification box and/or Alpha Merging Specification box and specified in further sub-parts of ISO/IEC 18477.

The Residual Data box shall only appear at the top-level of the codestream and not as a subbox of a superbox. Data contained in this box defines residual data that extends the base image encoded in the base JPEG stream to an IDR or HDR image. The sample precision of the samples within the codestream shall be either 8 or 12. While subsampling factors may be different from the base image, the number of components in the residual codestream and the image dimensions shall be identical to those signaled in the legacy codestream.

The type of this box shall be 0x52455349, ASCII encoding of “RESI”. There shall be at most one stream of residual data after concatenating all JPEG extension marker payload data belonging to this box type.

NOTE Unlike refinement coding, residual coding merges all data into one single box. This box may, however, extend over several JPEG XT marker segments.

The structure of this box is defined in Figure B.10, the parameters and sizes in Table B.11.

Entropy Coded Data, conforming to Rec. ITU-T T.81 | ISO/IEC 10918-1

Figure B.10 — Organization of the Residual Data box

Table B.11 — Residual Data box, parameters and sizes

Parameter	Size (bits)	Value	Meaning
Data	Varies	Varies	Entropy coded data segment of variable lengths.

B.12 Residual Refinement Data box

This box encapsulates entropy coded data segments that extend the bit-precision of the extension image(s) in the DCT domain. The syntax of the encapsulated data and the decoding algorithm of this data is specified in the part(s) of ISO/IEC 18477 that make use of this box. The Residual Refinement Data Box shall only appear at the top-level of the codestream and not as a subbox of a superbox. If the number of additional residual refinement bits, the R_r-Parameter of the Refinement Specification subbox is non-zero, residual refinement data encapsulated in Residual Refinement Data Boxes shall be present.

The type of this box shall be 0x5246494e, ASCII encoding of “RFIN”. The structure of the payload data of this box is defined in Figure B.11, the parameters and sizes in Table B.12.

Entropy Coded Data

Figure B.11 — Organization of the Residual Refinement Data box

Table B.12 — Residual Refinement Data box, parameters and sizes

Parameter	Size (bits)	Value	Meaning
Data	Varies	Varies	Entropy coded data segment of variable lengths.

B.13 Refinement Data box

This box encapsulates entropy coded data segments that extend the bit-precision of the base image(s) in the DCT domain. The syntax of the encapsulated data and the decoding algorithm of this data is specified in the part(s) of ISO/IEC 18477 that make use of this box. It shall only appear at the top level of the codestream and not as a subbox of a superbox. If the number of additional base refinement bits, the R_h-Parameter of the Refinement Specification subbox of the Merging Specification Box is non-zero, refinement data encapsulated in Residual Refinement Data Boxes shall be present.

The type of this box shall be 0x46494e45, ASCII encoding of “FINE”. The structure of the payload data of this box is defined in Figure B.12, the parameters and sizes in Table B.13.

Entropy Coded Data

Figure B.12 — Organization of the Refinement Data Box

Table B.13 — Refinement Data box, parameters and sizes

Parameter	Size (bits)	Value	Meaning
Data	Varies	Varies	Entropy coded data segment of variable lengths.

B.14 Output Conversion box

This box defines the final bitdepth of the reconstructed samples and defines the conversion process from the result of the base image/residual image merging process to the final output samples. This box shall never appear top level in the file, but it shall be a subbox of the Merging Specification Box or Alpha Merging Specification box. Exactly one Output Conversion Box shall appear in the Merging Specification Box if a Merging Specification Box exists, and exactly one Output Conversion Box shall appear in the Alpha Merging Specification box if such a box exists.

The operations performed by the Output Conversion process are as follows:

— First, the data shall be clipped to range if the Ce flag is set to one. The interval to clip to depends on the value of the Oc bit, and the algorithm for clipping is further specified in additional parts of ISO/IEC 18477. If the Ce flag is zero, no clipping is performed.

— Second, if the Oc flag is set, the data shall be converted from integer to floating point by a map that is further specified in additional parts of ISO/IEC 18477.

— If the Ol flag is set, the output shall be further transformed by a non-linear point transformation as specified in Annex C and which is defined by the curves or tables addressed by the parameters to_0 through to_3. The Ol flag and the Oc flag shall not be both set at the same time.

If Ol is set, the to_i value selects for component i a non-linear transformation by the algorithm specified in B.16. The non-linear transformation is either defined by a table lookup through the boxes specified in B.2 or B.3, or by applying a parametric curve, see B.4.

The non-linear transformation itself follow the specifications of Annex C. It requires four additional parameters, the input range R_w, R_e and the output range R_t, R_f. The parameters shall be as follows:

— $R_w = 1$ $R_e = 0$

— $R_t = 1$ $R_f = 0$

The value of the rounding mode e shall be 0. Note that its value is, however, ignored for the above choice of R_t and R_w.

The type of this box shall be 0x4F434F4E, ASCII encoding of "OCON".

Figure B.13 describes the organization of this box, Table B.14 the parameters and parameter sizes of the box.

R_b	Lf	Oc	Ce	Ol	to_0	to_1	to_2	to_3

Figure B.13 — Layout of the Output Conversion box

Table B.14 — Parameters of the Output Conversion box

Parameter	Size (in bits)	Value	Meaning
R_b	4	8	Number of additional bits available for high dynamic range data. The bit precision of the reconstructed high dynamic range image shall be computed as $8+R_b$.
Lf	1	0..1	This field indicates whether DCT compression is lossy or lossless/near lossless. This field may be used to indicate that the DCT implementation is constrained to particular algorithms defined in further parts of the ISO/IEC 18477 family of standards. If this field is 0, the only constraint on the DCT implementation is that it shall satisfy the error bounds of Rec. ITU-T T.83 | ISO/IEC 10918-2.
Oc	1	0..1	If this flag is set, clipped samples are rounded to integer and converted to floating point by a map that is further specified in additional parts of ISO/IEC 18477. If this flag is set Ol shall be 0
Ce	1	0..1	This field indicates whether the output shall be clipped to range before processing the data further. If the Ce flag and the Oc flag are both enabled, clipping is applied before conversion to floating point. The interval may on the value of the Oc flag. Additional parts of ISO/IEC 18477 define the interval to clip to.

Table B.14 *(continued)*

Parameter	Size (in bits)	Value	Meaning
Ol	1	0..1	This field indicates whether an output lookup or point transformation is required. If enabled, the output transformation is specified by the to_0 through to_3 fields.
to_0	4	0	If Ol is one, this field defines the output table for component 0 in other parts of the ISO/IEC 18477 family of standards. If Ol is 0, this field is ignored.
to_1	4	0	If Ol is one, this field defines the output table for component 1 in other parts of the ISO/IEC 18477 family of standards. If Ol is 0, this field is ignored.
to_2	4	0	If Ol is one, this field defines the output table for component 2 in other parts of the ISO/IEC 18477 family of standards. If Ol is 0, this field is ignored.
to_3	4	0	Reserved for ITU/ISO purposes.

B.15 Refinement Specification box

This box defines the number of refinement scans for both the base and the residual coding process and delivers the parameters R_h and R_r required by other parts of the ISO/IEC 18477 family of standards. At most one Refinement Specification box shall exist as subbox of the Merging Specification Box and Alpha Merging Specification box. This box shall not appear at top-level of the file, and it shall present if and only if the file includes either Refinement Data Boxes or Residual Refinement Boxes.

The value R_h shall be identical to the number of refinement scans adding least significant bits to the base image, and is thus identical to the number of Refinement Data boxes. If the Refinement Specification Box is absent, its inferred value is 0.

The value R_r shall be identical to the number of residual refinement scans adding least significant bits to the residual image, and it is thus identical to the number of Residual Refinement boxes at top level of the file. If the Refinement Specification box is absent, the inferred value of R_r is 0.

The type of the Refinement Specification box shall be 0x52535043, ASCII encoding of “RSPC”.

Figure B.14 describes the organization of this box, Table B.15 the parameters and parameter sizes.

R_h	R_r

Figure B.14 — Layout of the Refinement Specification Box

Table B.15 — Parameters of the Base Transformation box

Parameter	Size (in bits)	Value	Meaning
R_h	4	0..4	Number of refinement scans in the base decoding path.
R_r	4	0..4	Number of refinement scans in the residual decoding path.

B.16 Non-linear Point Transformation Specification box

This subclause defines multiple boxes, see Table B.16, that are all used to specify a non-linear process in the merging process of the base image and the extension image. Details on the box types and their intended use are found in the corresponding parts of ISO/IEC 18477. All boxes defined in this subclause share the same box layout and the same syntax, see Figure B.15 and Table B.17.

Boxes of these types shall only exist as subboxes of the Merging Specification box or the Alpha Merging Specification box; they shall not appear at top-level of the file.

Table B.16 — Boxes and box types defined by this subclause

Box Name	Box Type	ASCII encoding of the Box Type
Base Non-linear Point Transformation Specification Box	0x4C505453	“LPTS”
Secondary Base Non-linear Point Transformation Specification Box	0x52505453	“CPTS”
Residual Non-linear Point Transformation Specification Box	0x5152505453	“QPTS”
Intermediate Residual Non-linear Point Transformation Specification Box	0x44505453	“DPTS”
Secondary Residual Non-linear Point Transformation Specification Box	0x52505453	“RPTS”
Prescaling Non-linear Point Transformation Specification Box	0x53505453	“SPTS”
Postscaling Non-linear Point Transformation Box	0x50505453	“PPTS”

The non-linear point transformation selected by a Non-Linear Point Transformation Specification box is an Integer Table Lookup box, a Floating Point Lookup box or a Parametric Curve box. The corresponding boxes referenced by this box appear at top level of the ISO/IEC 18477-3 compliant file or as subboxes of the Merging Specification Box or Alpha Merging Specification box.

The non-linear point transformation for component i is found by matching the value td_i of this box against the value of M of all Parametric Curve boxes, Integer Table Lookup boxes or Floating Point Lookup boxes found in the same super box or at the top-level of the file. If a box with a matching M value exists both as a subbox of the same encapsulating super box and on the top-level of the file, the box within the super box takes precedence.

The non-linear point transformation itself is given by the process specified in Annex C. It requires five additional parameters, the value of the parameter e, the input range R_w, R_e and the output range R_t, R_f. The value of these parameters depends on the box type and is specified in the corresponding part of ISO/IEC 18477 which makes use of this box.

Figure B.15 depicts this process: A JPEG XT decoder reserves 16 slots for non-linear transformations. Each slot may be occupied by a non-linearity that is either specified by a Integer Table Lookup box (see B.2), or a Floating Point Table Lookup box (see B.3), or a Parametric Curve box (see B.4). The M value, present in all the above box types, selects the slot to be populated by the corresponding box.

Each Non-linear Point Transformation Specification box and the Output Conversion box select now up to four non-linearities by referencing the slot indices of the transformation to be used for each component. Non-linearities may be re-used by referencing their slot numbers multiple times.

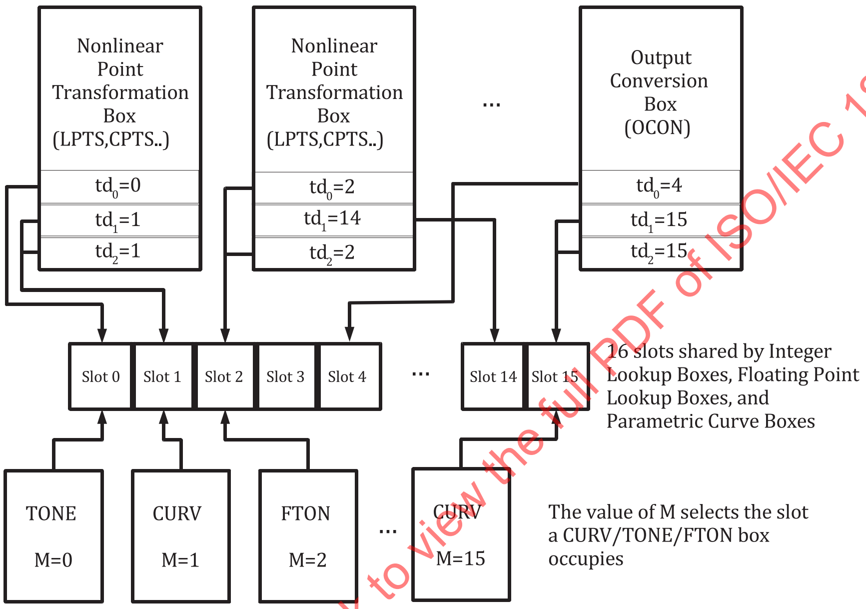

Figure B.15 — Selection of non-linearities through indirection by slot-indices

Figure B.16 describes the organization of this box, Table B.17 the parameters and parameter sizes.

td_0	td_1	td_2	td_3

Figure B.16 — Layout of the Non-linear Point Transformation Specification boxes

Table B.17 — Parameters and sizes of the Non-linear Point Transformation Specification boxes

Parameter	Size (in bits)	Value	Meaning
td_0	4	0..15	The value of td_0 selects the Integer or Floating point Table Lookup or Parametric Curve box whose M value matches the value td_0. This table or parametric curve is applied as non-linear point transformation of component 0.

Table B.17 *(continued)*

Parameter	Size (in bits)	Value	Meaning
td_1	4	0..15	The td_1 value is used to select the non-linear point transformation for component 1 if it is present. Otherwise, the value is ignored.
td_2	4	0..15	The td_2 value is used to select the non-linear point transformation for component 2 if it is present. Otherwise, the value is ignored.
td_3	4	0..15	Reserved for ITU | ISO/IEC purposes.

B.17 Linear Transformation Specification box

This subclause defines multiple boxes, see Table B.18, that all describe a linear transformation between the components of the base or extension image. Boxes of this type are used in multiple stages of the merging process of base and extension image defined in parts of ISO/IEC 18477. All boxes defined in this subclause share the same box layout and the same syntax, see Figure B.16 and Table B.19. Boxes of the types listed in Table B.18 shall only exist as subboxes of the Merging Specification box or Alpha Merging Specification box.

Table B.18 — Boxes and box types defined by this subclause

Box Name	Box Type	ASCII encoding of the Box Type
Residual Transformation Box	0x52545246	“RTRF”
Colour Transformation Box	0x43545246	“CTRF”
Residual Colour Transformation Box	0x44545246	“DTRF”
Prescaling Transformation Box	0x53545246	“STRF”

This box shall only exist as a subbox of the Merging Specification box. It shall only exist if the number of components in the image (base or extension) equals 3. If the number of components equals one, the transformation is the identity implicitly.

Xt values 0 to 4 correspond to pre-defined linear transformations that are defined in other sub-parts of ISO/IEC 18477, they are reserved for ITU | ISO/IEC purposes. Xt values between 5 and 15 are user-defined transformations.

The matrix for user-defined linear transformation is found by matching the value Xt of the Linear Transformation Specification box against the value of M of all Fixed Point and Floating Point Linear Transformation boxes found in the same super box or at the top-level of the file. If a box with a matching M value exists both as a subbox of the encapsulating super box and on the top-level of the file, the box within the super box is selected.

If Xt is below 5, the algorithm for the linear transformation is defined in other parts of the ISO/IEC 18477 family of standards. If Xt is 5 or above, the linear transformation is given by a matrix multiplication of the column vector of the three component input with the matrix contained in the Fixed Point or Floating Point Linear Transformation box whose M value matches the Xt value.

Figure B.17 depicts this process: A JPEG XT decoder reserves 16 slots for linear transformations. Five of the slots are pre-occupied with standard-defined transformations, all other slots can be user defined and can be occupied by either a Fixed Point or a Floating Point Linear Transformation box. The M value, present in both box types, selects the slot to be populated by the corresponding box.

Each Linear Transformation Specification box selects now one linear transformation by referencing the slot indices of the transformations they are supposed to apply to the data. Linear transformations may be re-used by referencing their slot numbers multiple times.

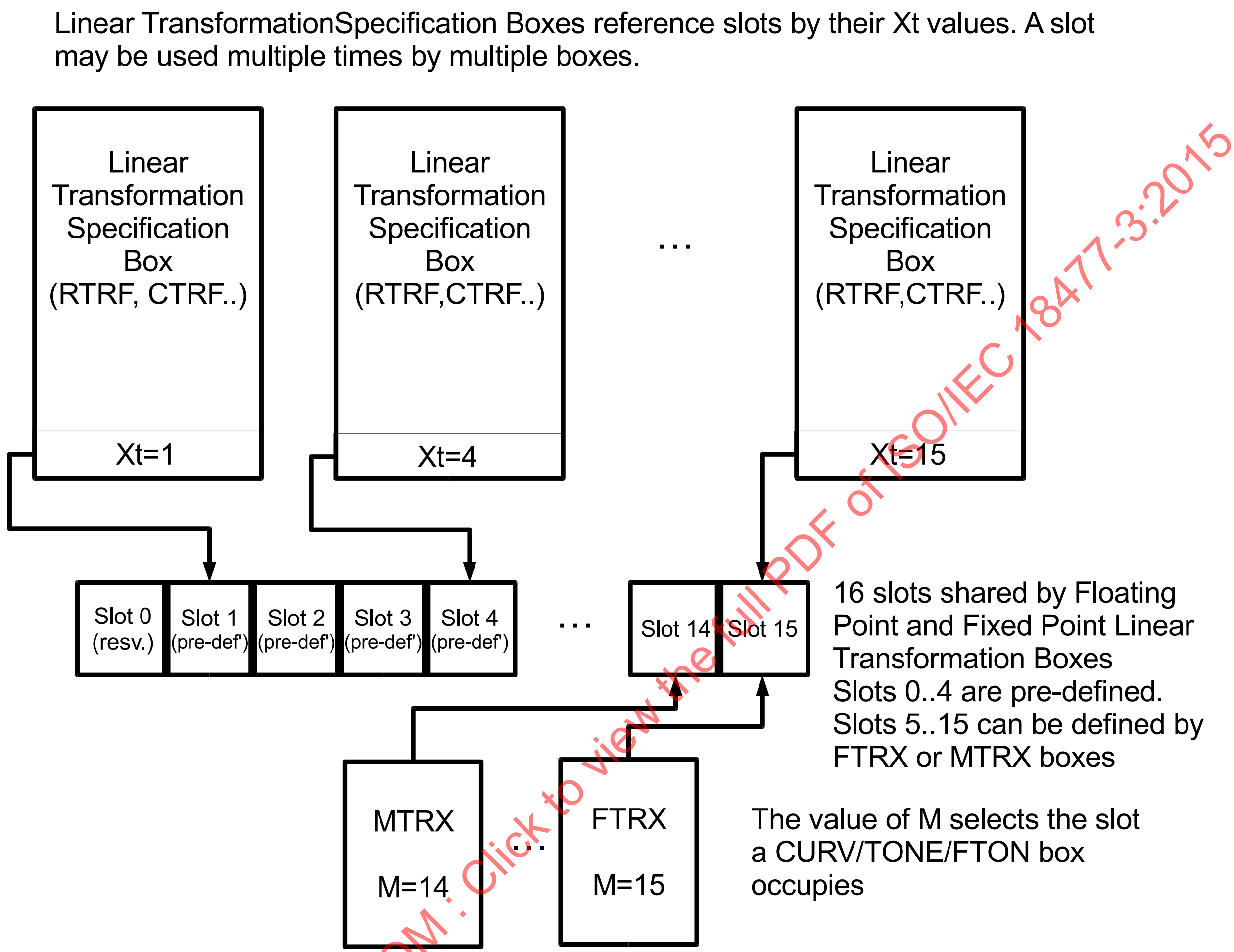

Figure B.17 — Selection of linear transformations through indirection by slot-indices.

Figure B.18 describes the organization of this box segment, Table B.19 the parameters and parameter sizes.

Xt	Re

Figure B.18 — Layout of the Linear Transformation specification boxes

Table B.19 — Parameters and sizes of the Linear Transformation Specification box

Parameter	Size (in bits)	Value	Meaning
Xt	4	0..15	Defines the linear transformation to be used. Valid choices and interpretation of this value are specified in other parts of ISO/IEC 18477.
Re	4	0	Reserved for ITU | ISO/IEC purposes

B.18 DCT Specification box

This subclause specifies multiple boxes, see Table B.20, that define the DCT transformation in either the base or residual decoding pass required for lossless or near-lossless decoding. Boxes of this type shall be present if and only if the Lf flag of the Output Conversion box is one. All boxes defined in this subclause share the same box layout and the same syntax, see Figure B.19 and Table B.21. Boxes of the types listed in Table B.20 shall only exist as subboxes of the Merging Specification box or Alpha Merging Specification box.

Table B.20 — Boxes and box types defined by this subclause

Box Name	Box Type	ASCII encoding of the Box Type
Base DCT Specification Box	0x4C444354	“LDCT”
Residual DCT Specification Box	0x52444354	“RDCT”

The encoding of the dct and Ns fields of this box, as well as the DCT itself is defined in other parts of ISO/IEC 18477. Valid choices for the parameters depend on the box type that is also specified in later parts of ISO/IEC 18477.

Figure B.19 describes the organization of this box, Table B.21 the parameters and parameter sizes.

dct	Ns

Figure B.19 — Layout of the DCT Specification boxes

Table B.21 — Parameters and sizes of the Base DCT Specification box

Parameter	Size (in bits)	Value	Meaning
dct	4	0..15	Selects the DCT that shall be used for reconstruction. Valid choices are defined in other parts of ISO/IEC 18477.
Ns	4	0..15	Defines other processing options in the absence of a DCT in the decoding path.

B.19 Merging Specification box

This box is a superbox that defines the process and parameters for merging low dynamic range image with the residual image to reconstruct the final output image. At most one Merging Specification box shall be present in the codestream. The contents of this superbox are specified in additional parts of the

ISO/IEC 18477 family of standards. This box shall be represented by a **single** JPEG XT marker segment in front of the first SOF marker of the ISO/IEC 18477-1 codestream.

The type of this box shall be 0x53504543, ASCII encoding of "SPEC".

Figure B.20 describes the organization of this box. Subboxes of this box are defined in Annex B.

Boxes (see A.5)

Figure B.20 — Organization of the Merging Specification box

B.20 Alpha Merging Specification box

This box is a superbox that encapsulates all boxes describing the process of generating the alpha channel from the Alpha Codestream Box, the Alpha Refinement Box, the Residual Alpha box and the Residual Alpha Refinement box. It does not contain the entropy coded data itself. Otherwise, this box is similar in content and purpose to the Merging Specification box.

At most one Alpha Merging Specification box shall be present in the codestream, and this box shall be present if and only if the JPEG XT file includes opacity information. This box shall be represented by a **single** JPEG XT marker segment in front of the first SOF marker of the ISO/IEC 18477-1 codestream.

The type of this box shall be 0x41535043, ASCII encoding of "ASPC".

Figure B.21 describes the organization of this box. Subboxes of this box are defined in Annex B.

Boxes (see A.5)

Figure B.21 — Organization of the Alpha Merging Specification box

B.21 Alpha Codestream box

This box encapsulates entropy coded segments forming the base layer of the opacity data of an image. The payload of this box shall form a codestream conforming to Rec. ITU-T T.81 | ISO/IEC 10918-1 confined to the baseline Huffman, extended Huffman or progressive Huffman coding modes, i. e. coding modes permitted by ISO/IEC 18477-1. The sample values reconstructed from this data define opacity information. This box shall be present if the Output Conversion box in the Alpha Merging Specification box is present.

The type of this box shall be 0x414C4641, ASCII encoding of "ALFA". There shall be at most one stream of opacity data after concatenating all JPEG extension marker payload data belonging to this box type.

NOTE This box is an optional box. This box may be missing because an image does not include opacity information or decoder implementations may ignore it if they do not intend to make use of the opacity information.

The structure of this box is defined in Figure B.22, the parameters and sizes in Table B.22.

Entropy Coded Data

Figure B.22 — Organization of the Alpha Codestream box