
**Information technology — JPEG 2000
image coding system —**

**Part 2:
Extensions**

*Technologies de l'information — Système de codage d'images JPEG
2000 —*

Partie 2: Extensions



Reference number
ISO/IEC 15444-2:2023(E)

Documents
Document Sharing Hub
© ISO/IEC 2023



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
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.

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by ITU-T (as Rec. ITU-T T.801) and drafted in accordance with its editorial rules, in collaboration with Joint Technical Committee ISO/IEC JTC 1, *Information technology, Subcommittee SC 29, Coding of audio, picture, multimedia and hypermedia information*.

This third edition cancels and replaces the second edition (ISO/IEC 15444-2:2021), which has been technically revised.

The main changes are as follows:

- Support for progression order extensions.

A list of all parts in the ISO/IEC 15444 series can be found on the ISO and IEC websites.

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 and www.iec.ch/national-committees.

CONTENTS

Page

1	Scope	1
2	Normative references	1
3	Definitions	2
4	Abbreviations	4
5	Conventions	4
6	General description	4
6.1	Extensions specified by this Recommendation International Standard	5
6.2	Relation between extensions	6
Annex A	Compressed data syntax, extension	8
A.1	Extended capabilities	8
A.2	Extensions to Rec. ITU-T T.800 ISO/IEC 15444-1 marker segment parameters	8
A.3	Extended marker segments	16
Annex B	Variable DC offset, extension	35
B.1	Variable DC offset flow	35
B.2	Inverse DC offset	35
B.3	Forward DC offset (informative)	35
Annex C	Variable scalar quantization, extension	37
C.1	Variable scalar quantization	37
C.2	Variable scalar dequantization for irreversible filters	37
C.3	Variable scalar quantization for irreversible filters (informative)	37
Annex D	Trellis coded quantization extensions	39
D.1	Introduction to TCQ	39
D.2	Sequence definition	40
D.3	Forward TCQ quantization (informative)	41
D.4	Inverse quantization (normative)	42
D.5	Lagrangian rate allocation (informative)	45
Annex E	Visual masking, extensions	50
E.1	Introduction to visual masking (informative)	50
E.2	Point-wise extended non-linearity (informative)	50
E.3	Decoding with visual masking	52
E.4	Encoding with visual masking (informative)	53
E.5	Setting parameters (informative)	53
E.6	Compatibility with other technologies (informative)	53
Annex F	Arbitrary decomposition of tile-components, extensions	54
F.1	Wavelet sub-bands	54
F.2	Equation, text and decomposition updates	55
F.3	Inverse discrete wavelet transformation for general decompositions	64
F.4	Forward discrete wavelet transformation for general decompositions (informative)	71
Annex G	Whole-sample symmetric transformation of images, extensions	78
G.1	Wavelet transformation parameters, definitions and normalizations	78
G.2	Whole-sample symmetric (WS) wavelet transformations reconstruction	78
G.3	Whole-sample symmetric (WS) wavelet transformation decomposition (informative)	81
G.4	Examples of WS wavelet transformations (informative)	82
Annex H	Transformation of images using arbitrary wavelet transformations	85
H.1	Wavelet transformation parameters and normalizations	85
H.2	Arbitrary (ARB) wavelet transformation reconstruction procedures	85
H.3	Arbitrary (ARB) wavelet transformation decomposition procedures (informative)	91
H.4	Examples of ARB wavelet transformations (informative)	94

Annex I – Single sample overlap discrete wavelet transform, code-block anchor point and progression order extensions	98
I.1 Introduction to single sample overlapping	98
I.2 The code-block anchor points (CBAP) extension	98
I.3 The SSO extension	102
I.4 The TSSO extension	110
I.5 Combining the SSO and TSSO extensions (informative)	111
Annex J – Multiple component transformations, extension	112
J.1 Introduction to multiple component transformation concepts	112
J.2 Overview of inverse processing	112
J.3 Transformations	118
Annex K – Non-linear transformation	128
K.1 Signalling the use of the non-linear transformations	128
K.2 Non-linear transformation specifications	129
Annex L – Region of interest coding and extraction, extensions	133
L.1 Decoding of ROI	133
L.2 Description of the Scaling based method	133
L.3 Region of interest mask generation	134
L.4 Remarks on region of interest coding	138
Annex M – JPX extended file format syntax	139
M.1 File format scope	139
M.2 Introduction to JPX	139
M.3 Greyscale/Colour/Palette/multi-component specification architecture	142
M.4 Fragmenting the codestream between one or more files	143
M.5 Combining multiple codestreams	145
M.6 Using reader requirements masks to determine how a file can be used	149
M.7 Extensions to the JPX file format	156
M.8 Differences from the JP2 binary definition	157
M.9 Conformance	157
M.10 Key to graphical descriptions (informative)	161
M.11 Defined boxes	161
M.12 Dealing with unknown boxes	210
M.13 Using the JPX file format in conjunction with other multi-media standards (informative)	210
M.14 Decomposing an XML document into multiple boxes	211
Annex N – JPX file format extended metadata definition and syntax	213
N.1 Introduction to extended metadata	213
N.2 Additional references for extended metadata	213
N.3 Scope of metadata definitions	213
N.4 Metadata syntax	214
N.5 Defined boxes	215
N.6 Metadata definitions	217
N.7 Fundamental type and element definitions	258
N.8 JPX extended metadata document type definition	284
N.9 JPX extended metadata XML Schema	304
Annex O – Examples and guidelines, extensions	347
O.1 Arbitrary decomposition examples	347
O.2 Odd Tile Low Pass First (OTLPPF) convention	368
O.3 Multiple component collection example	369
O.4 Background to enhancement of quantization	379
O.5 Wrapping JPEG XR (Rec. ITU-T T.832 ISO/IEC 29199-2) Codestreams by the JPX file format ...	379
O.6 Representing floating point numbers within JPEG 2000	382
O.7 Working with ROI Description boxes	383

	Page
Annex P – Block coder extensions	384
P.1 Selective arithmetic coding bypass (lazy mode)	384
P.2 Enhancement of selective arithmetic coding bypass (fast mode).....	384
Bibliography	386

List of Tables

Table A.1 – Syntax support for extensions	8
Table A.2 – Capability Rsiz parameter, extended	9
Table A.3 – Start of tile-part parameter values, extended.....	9
Table A.4 – Number of tile-parts, TNsot, parameter value, extended	9
Table A.5 – Coding style parameter values for the Scod parameter.....	10
Table A.6 – Coding style parameter values of the SGcod parameter	11
Table A.7 – Coding style parameter values of the SPcod and SPcoc parameters, extended	11
Table A.7bis – Progression order for the SGcod and Ppoc parameters	11
Table A.8 – Multiple component transformation for the SGcod parameters.....	11
Table A.9 – Decomposition for the SPcod and SPcoc parameters, extended	12
Table A.10 – Transformation for the SPcod and SPcoc parameters, extended.....	12
Table A.11 – SSO parameters, extended	12
Table A.11bis – SXcod parameter.....	12
Table A.11ter – Progression order change, tile parameter values.....	13
Table A.12 – Quantization default values for the Sqcd, Sqcc, Sqpd, and Sqpc parameters, extended	14
Table A.13 – Quantization values (irreversible transformation only), extended	14
Table A.14 – SPqcd, SPqcc, SPqpd, and SPqpc parameters (irreversible transformation only), extended	14
Table A.15 – SPqcd, SPqcc, SPqpd, and SPqpc parameters (irreversible transformation only), extended	15
Table A.16 – Region-of-interest parameter values for the Srgn parameter	15
Table A.17 – Component index parameter value for the Crgn parameter	15
Table A.18 – Region-of-interest values from SPrgn parameter (Srgn = 1 or Srgn = 2)	15
Table A.19 – List of markers and marker segments	16
Table A.20 – Variable DC offset parameter values	17
Table A.21 – Variable DC offset parameter values for the Sdco parameter.....	17
Table A.22 – Visual masking parameter values	18
Table A.23 – Component parameter value for the Cvms parameter.....	18
Table A.24 – Visual masking for the Svms parameters.....	18
Table A.25 – Downsampling factor styles parameter values.....	19
Table A.26 – Arbitrary decomposition styles parameter values	20
Table A.27 – Arbitrary transformation parameter values	21
Table A.28 – Arbitrary transformation values for the Satk parameter.....	22
Table A.29 – Component bit depth definition parameter values	23
Table A.30 – Component bit depth definition values for the Ncbd parameter	23
Table A.31 – Component bit depth definition values for the BDCbdi parameter.....	23
Table A.32 – Multiple component transformation definition parameter values	24
Table A.33 – Multiple component transformation definition values for the Imct parameter	24
Table A.34 – Multiple component collection parameter values	26
Table A.35 – Multiple component collection values for the Xmcci parameter	26

Page

Table A.36 – Multiple component collection values for the Nmcci parameter	26
Table A.37 – Multiple component collection values for the Mmcci parameter.....	27
Table A.38 – Multiple component collection values for the Tmcci parameter (array-based).....	27
Table A.39 – Multiple component collection values for the Tmcci parameter (wavelet-based)	27
Table A.40 – Multiple component intermediate collection parameter values	28
Table A.41 – Non-linearity transformation parameter values	29
Table A.42 – Non-linearity transformation parameter values for the Cnlt parameter	29
Table A.43 – Decoded image component bit depth parameter values for the BDnlt parameter	29
Table A.44 – Non-linearity transformation parameter values of the Tnlt parameter.....	30
Table A.45 – Non-linearity transformation parameter values of the STnlt parameter (Tnlt = 1)	30
Table A.46 – Non-linearity transformation parameter values of the STnlt parameter (Tnlt = 2)	30
Table A.47 – Quantization default, precinct parameter values.....	32
Table A.48 – Quantization precinct component parameter values	33
Table A.49 – Ccap2 syntax and semantics	33
Table A.50 – Precinct length, tile-part header parameter values	34
Table A.51 – Srlt values and semantics	34
Table A.52 – Semantics of Jrlti values when Srlt is in the range [0, 231 – 1]	34
Table D.1 – Parent LUTs for $k > 0$ in the trellis of Figure D.3.....	42
Table D.2 – Description of functional blocks in Figure D.4.....	42
Table D.3 – Description of functional blocks in Figure D.5.....	43
Table D.4 – Look-up table for $A(s)$	44
Table D.5 – Look-up table for $S(s, qk)$	44
Table D.6 – Description of functional blocks for Figure D.6	45
Table D.7 – Sub-band statistics required for LRA.....	46
Table D.8 – ρ_b parameters for TCQ	46
Table D.9 – Δ_b parameters for TCQ	46
Table D.10 – ρ_b parameters for SQ	47
Table D.11 – Δ_b parameters for SQ	47
Table D.12 – Description of functional blocks in Figure D.7.....	49
Table F.1 – Updates to contexts for significance propagation and cleanup coding passes.....	56
Table F.2 – Quantities for sub-band info calculation.....	60
Table F.3 – $S(ab)$ and $J(ab)$ as a function of $dS(i)$	64
Table F.4 – $S(ab)$ and $J(ab)$ as a function of $dR(i)$	64
Table F.5 – Characteristics for sample wavelet decomposition in Figure F.14	65
Table G.1 – Parameters for wavelet transformations.....	78
Table G.2 – Parameters of the 5-3 reversible wavelet transformation.....	83
Table G.3 – Parameters of the 13-7 reversible wavelet transformation.....	83
Table G.4 – Parameters of the 5-3 irreversible wavelet transformation	84
Table G.5 – Parameters of the irreversible 7-5 wavelet transformation	84
Table G.6 – Parameters of the irreversible 9-7 wavelet transformation	84
Table H.1 – Additional parameters for arbitrary wavelet transformations	85
Table H.2 – Minimum left extension length.....	89
Table H.3 – Minimum right extension length.....	89
Table H.4 – Parameters of the reversible Haar 2-2 wavelet transformation	95

Table H.5 – Parameters of the reversible 2-6 wavelet transformation.....	95
Table H.6 – Parameters of the reversible 2-10 wavelet transformation.....	95
Table H.7 – Parameters of the irreversible 6-10 wavelet transformation	96
Table H.8 – Parameters of the irreversible 10-18 wavelet transformation	96
Table M.1bis – Brand Values for JPEG XR Codestreams.....	142
Table M.1 – Example expression	151
Table M.2 – Expanded expression.....	151
Table M.3 – Example factored expression	151
Table M.4 – Example of a Reader Requirements expressions for Equations M-6 and M-7.....	153
Table M.5 – Example of a Reader Requirements box for Equations M-6 and M-7	153
Table M.6 – Reader Requirements table for Equations M-10 and M-11.....	154
Table M.7 – Reader Requirements box data for Equations M-10 and M-11.....	154
Table M.8 – Reader Requirements box data for Equations M-16 and M-17.....	155
Table M.9 – Example Reader Requirements box to test.....	155
Table M.10 – Table intentionally left blank	156
Table M.11 – Items which can be extended through Recommendations International Standards	156
Table M.12 – Items which can be extended by registration	157
Table M.13 – Boxes defined within this Recommendation International Standard.....	163
Table M.14 – Legal values of the SFi field	165
Table M.15 – Format of the contents of the Reader Requirements box	167
Table M.16 – Format of the contents of the Data Reference box	168
Table M.17 – Format of the contents of the Fragment List box	169
Table M.18 – Format of the contents of the Cross-Reference box	170
Table M.19 – Legal C values.....	171
Table M.20 – BPC and BPCi parameters	172
Table M.21 – Format of the contents of the Image Header box	172
Table M.22 – Legal METH values	176
Table M.23 – Legal APPROX values.....	177
Table M.24 – Format of the contents of the Colour Specification box.....	177
Table M.24bis – Nominal maximum sample values.....	177
Table M.25 – Additional legal EnumCS values	178
Table M.26 – Format of the contents of the METHDAT field for the Enumerated method	179
Table M.27 – Format of the contents of the METHDAT field for the Any ICC method	180
Table M.28 – Format of the contents of the METHDAT field for the Vendor Colour method.....	180
Table M.28bis – Format of the METHDAT field for the Parameterized method.....	181
Table M.29 – Standard illuminant values for CIELab.....	182
Table M.29bis – Default Offset Values and Encoding of Offsets for the CIELab Colourspace.....	182
Table M.30 – Format of the contents of the EP field for CIELab (EnumCS = 14)	183
Table M.30bis – Default Offset Values and Encoding of Offsets for the CIEJab Colourspace	184
Table M.31 – Format of the contents of the EP field for CIEJab (EnumCS = 19)	184
Table M.32 – Colours indicated by the Asoci field	185
Table M.33 – Otyp field values	186
Table M.34 – Format of the contents of the Opacity box	186
Table M.35 – Format of the contents of the Codestream Registration box	188

Page

Table M.35bis – Allowed values for the pixel format	189
Table M.35ter – Common floating point formats (informative).....	189
Table M.36 – Format of the contents of the Composition box	190
Table M.37 – Format of the contents of the Composition Options box.....	191
Table M.38 – Ityp field values.....	192
Table M.39 – Format of the contents of the Instruction Set box	192
Table M.40 – Format of the contents of the INSTi parameter in the Instruction Set box.....	194
Table M.40bis – Encoding of the ROT field	194
Table M.41 – Format of the contents of the Association box	196
Table M.42 – ANi field values	196
Table M.43 – Format of the contents of the Number List box	196
Table M.44 – Legal Filter types.....	197
Table M.45 – Format of the contents of the Binary Filter box	198
Table M.46 – Format of the contents of the Graphics Technology Standard Output box	199
Table M.47 – Legal Ri values.....	199
Table M.48 – Allowed Rtypi values	200
Table M.49 – Format of the contents of the ROI Description box	200
Table M.49bis – Interpreting the 2 bit D field of Rtypi for quadrilateral refinements.....	201
Table M.50 – Legal Styp values	202
Table M.51 – Legal Ptyp values	203
Table M.52 – Format of the contents of the Digital Signature box	203
Table M.53 – Format of the contents of the Multiple Codestream box	208
Table M.54 – Format of the contents of the Multiple Codestream Info box	209
Table N.1 – Format of the contents of the Image Creation box.....	215
Table N.2 – Format of the contents of the Content Description box	216
Table N.3 – Format of the contents of the History box	216
Table N.4 – Format of the contents of the Intellectual Property Rights box	217
Table N.5 – Format of the contents of the Image Identifier box.....	217
Table N.6 – Image Source values	219
Table N.7 – Scene type values.....	219
Table N.8 – Sensor technology values.....	222
Table N.9 – Exposure program values.....	228
Table N.10 – Metering mode values.....	228
Table N.11 – Scene illuminant values	229
Table N.12 – Back light values.....	229
Table N.13 – Auto focus values.....	230
Table N.14 – Name description values.....	252
Table N.15 – Date description values.....	254
Table N.16 – Additional name description values.....	258
Table N.17 – Address component type values.....	264
Table N.18 – Address type values	264
Table N.19 – Phone number type values	265
Table N.20 – Name component type values	268
Table N.21 – Latitude reference values	276

	Page
Table N.22 – Latitude values	276
Table N.23 – Longitude reference values	276
Table N.24 – Longitude values	277
Table N.25 – GPS Status values	277
Table N.26 – GPS Measure mode values	277
Table N.27 – GPS Speed reference unit values	277
Table N.28 – Direction reference values	278
Table N.29 – GPS Destination distance reference unit values	278
Table O.1 – Sub-band labels for Figure O.15.	352
Table O.2 – Mapping between ROT and SPATIAL_XFRM_SUBORDINATE	381
Table P.1 – Selective arithmetic coding bypass (default); (the same as Table D.9 of ITU-T T.800 ISO/IEC 15444-1)...	384
Table P.2 – Example of two bit planes (fast mode)	385

List of Figures

Figure 6-1 – Decoder block diagram	7
Figure A.1bis – Coding style default syntax	10
Figure A.1 – Variable DC offset syntax	16
Figure A.2 – Visual masking syntax	17
Figure A.3 – Downsampling factor styles syntax	18
Figure A.4 – Arbitrary decomposition styles syntax	19
Figure A.5 – Arbitrary transformation default syntax	20
Figure A.6 – Component bit depth definition syntax	22
Figure A.7 – Multiple component transformation definition syntax	23
Figure A.8 – Multiple component collection syntax	25
Figure A.9 – Multiple component transformation ordering syntax	28
Figure A.10 – Non-linearity point transformation syntax	28
Figure A.11 – Quantization default, precinct syntax	31
Figure A.12 – Quantization precinct component syntax	32
Figure A.13 – Precinct length, tile-part header syntax	34
Figure B.1 – Placement of the DC offset with multiple component transformation	35
Figure B.2 – Placement of the DC offset without multiple component transformation	35
Figure D.1 – Scalar quantizers used for TCQ	39
Figure D.2 – Union quantizers for TCQ	40
Figure D.3 – Trellis showing node indices	40
Figure D.4 – Forward TCQ processing	41
Figure D.5 – Full inverse processing for TCQ indices	43
Figure D.6 – Approximate dequantization of TCQ indices	45
Figure D.7 – Lagrangian rate allocation	48
Figure E.1 – System diagram for point-wise extended masking extension	50
Figure E.2 – Non-uniform quantization for self-contrast masking	51
Figure E.3 – Causal neighbourhood	52
Figure F.1 – Possible splits of sub-bands	55
Figure F.2 – Parameters for the GET_HOR_DEPTH and GET_VER_DEPTH procedures	56
Figure F.3 – The GET_HOR_DEPTH and GET_VER_DEPTH procedures	57

Page

Figure F.4 – Parameters for the SET_SUBBAND_INFO procedure	58
Figure F.5 – The SET_SUBBAND_INFO procedure	58
Figure F.6 – Parameters for the RECUR_INFO procedure	59
Figure F.7 – The RECUR_INFO procedure	59
Figure F.8 – Parameters for the INIT_□ procedure.....	60
Figure F.9 – Procedure for setting maximum number of sub-levels, □(lev)	61
Figure F.10 – Parameters for the INIT_S_R procedure	61
Figure F.11 – Upper level procedure for defining S(ab) and R(lev).....	62
Figure F.12 – Parameters for the LEV_S procedure.....	63
Figure F.13 – Procedure for defining S(ab)	63
Figure F.14 – Sample wavelet decomposition with labelled sub-bands	64
Figure F.15 – Parameters for the MOD_IDWT procedure	65
Figure F.16 – The MOD_IDWT procedure.....	66
Figure F.17 – Parameters for the MOD_2D_SR procedure.....	66
Figure F.18 – The MOD_2D_SR procedure.....	67
Figure F.19 – Parameters for the MOD_2D_INTERLEAVE procedure.....	67
Figure F.20 – The MOD_2D_INTERLEAVE procedure.....	68
Figure F.21 – Parameters for the 2D_HV_INTERLEAVE procedure	68
Figure F.22 – The 2D_HV_INTERLEAVE procedure	69
Figure F.23 – Parameters for the 2D_H_INTERLEAVE procedure	70
Figure F.24 – The 2D_H_INTERLEAVE procedure	70
Figure F.25 – Parameters for the 2D_V_INTERLEAVE procedure	71
Figure F.26 – The 2D_V_INTERLEAVE procedure	71
Figure F.27 – Parameters for the MOD_FDWT procedure	71
Figure F.28 – The MOD_FDWT procedure	72
Figure F.29 – Parameters for the MOD_2D_SD procedure	72
Figure F.30 – The MOD_2D_SD procedure	73
Figure F.31 – Parameters for the MOD_2D_DEINTERLEAVE procedure.....	73
Figure F.32 – The MOD_2D_DEINTERLEAVE procedure	74
Figure F.33 – Parameters for the 2D_HV_DEINTERLEAVE procedure	74
Figure F.34 – The 2D_HV_DEINTERLEAVE procedure	75
Figure F.35 – Parameters for the 2D_H_DEINTERLEAVE procedure.....	76
Figure F.36 – The 2D_H_DEINTERLEAVE procedure	76
Figure F.37 – Parameters for the 2D_V_DEINTERLEAVE procedure.....	76
Figure F.38 – The 2D_V_DEINTERLEAVE procedure.....	77
Figure G.1 – Parameters of the 1D_SR_WS procedures	79
Figure G.2 – The 1D_SR_WS procedure	79
Figure G.3 – Parameters of the 1D_FILTR_WS procedure	80
Figure G.4 – Parameters of the 1D_FILTR_WS procedure	80
Figure G.5 – Parameters of the 1D_SD_WS procedure	81
Figure G.6 – The 1D_SD_WS procedure	81
Figure G.7 – Parameters of the 1D_FILTD_WS procedure	82
Figure G.8 – Parameters of the 1D_FILTD_WS procedure	82
Figure H.1 – Parameters of the extended 1D_SR_ARB procedure	86

	Page
Figure H.2 – Extended procedure 1D_SR_ARB	87
Figure H.3 – Parameters of the 1D_SCALER procedure	87
Figure H.4 – Parameters of the 1D_STEPR procedure.....	88
Figure H.5 – Procedure 1D_STEPR.....	88
Figure H.6 – Parameters of the 1D_EXT_WS procedure.....	89
Figure H.7 – Parameters of the 1D_EXT_CON procedure	89
Figure H.8 – Parameters of the 1D_UPDATER_REV procedure	90
Figure H.9 – Parameters of the 1D_UPDATER_IRR procedure.....	90
Figure H.10 – Parameters of the extended 1D_SD_ARB procedure.....	91
Figure H.11 – Extended procedure 1D_SD_ARB.....	92
Figure H.12 – Parameters of the 1D_STEPPD procedure	92
Figure H.13 – Procedure 1D_STEPPD.....	93
Figure H.14 – Parameters of the 1D_UPDATED_REV procedure.....	93
Figure H.15 – Parameters of the 1D_UPDATED_IRR procedure	94
Figure H.16 – Parameters of the 1D_SCALED procedure	94
Figure H.17 – Lifting implementation for forward half-sample symmetric wavelet transformations.....	97
Figure I.1 – Precincts of one reduced resolution (modified Figure B.8 of Rec. ITU-T T.800 ISO/IEC 15444-1) ..	99
Figure I.2 – Codeblocks and precincts in sub-band b from four different tiles	100
Figure I.3 – The IDWT_SSO Procedure.....	103
Figure I.4 – The 2D_SR_SSO procedure	104
Figure I.5 – Parameters of the 1D_FILTR_SSO procedure.....	104
Figure I.6 – The FDWT_SSO procedure.....	106
Figure I.7 – The 2D_SD_SSO procedure	107
Figure I.8 – Parameters of the 1D_FILTD_SSO procedure	108
Figure I.9 – Position of SSO blocks	109
Figure I.10 – Tiling of the reference grid diagram	111
Figure J.1 – Inverse multiple component transformation processing	113
Figure J.2 – Procedure MCO_TRANSFORM.....	114
Figure J.3 – A single multiple component collection transformation (MCC_TRANS) stage	115
Figure J.4 – Procedure MCC_TRANS	116
Figure J.5 – A single component collection transformation (CC_TRANS) stage	117
Figure J.6 – Procedure CC_TRANS.....	117
Figure J.7 – SERM implementation of reversible decorrelation transformation	121
Figure J.8 – SERM implementation of forward reversible decorrelation transformation.....	122
Figure J.9 – Irreversible dependency transformation.....	123
Figure J.10 – Forward irreversible dependency transformation	124
Figure J.11 – Reversible dependency transformation.....	125
Figure J.12 – Forward reversible dependency transformation.....	126
Figure K.1 – Non-linear transformation application during decoding.....	128
Figure K.2 – Example gamma-type forward non-linear transformation.....	130
Figure L.1 – Rectangular mask on the reference grid.....	135
Figure L.2 – Elliptic mask on the reference grid	135
Figure M.1 – Example fragmented JPX file where all fragments are in the same file	144
Figure M.2 – Example fragmented JPX file where some fragments are stored in other files or resources	145

Figure M.3 – Example combination of two codestreams into a single compositing layer.....	146
Figure M.4 – Example of the box description figures	161
Figure M.5 – Example of the superbox description figures.....	161
Figure M.6 – Boxes defined within a JPX file.....	162
Figure M.7 – Organization of the contents of the Reader Requirements box.....	165
Figure M.8 – Organization of the contents of a Data Reference box.....	168
Figure M.9 – Organization of the contents of a Fragment Table box	168
Figure M.10 – Organization of the contents of a Fragment List box.....	169
Figure M.11 – Organization of the contents of a Fragment table box	170
Figure M.12 – Organization of the contents of an Image Header box.....	171
Figure M.13 – Organization of the contents of a Codestream Header box.....	173
Figure M.14 – Organization of the contents of a Compositing Layer Header box.....	174
Figure M.15 – Organization of the contents of a Colour Group box.....	175
Figure M.16 – Organization of the contents of a Colour Specification box	176
Figure M.17 – Organization of the contents of the METHDAT field for the Enumerated method.....	178
Figure M.18 – Organization of the contents of the METHDAT field for the Any ICC method.....	180
Figure M.19 – Organization of the contents of the METHDAT field for the Vendor Colour method	180
Figure M.19bis – Organization of the METHDAT field for the Parameterized method	181
Figure M.20 – Organization of the contents of the EP field for the CIELab (EnumCS = 14).....	181
Figure M.21 – Organization of the contents of the EP field for the CIEJab (EnumCS = 19).....	183
Figure M.22 – Organization of the contents of an Opacity box.....	185
Figure M.23 – Organization of the contents of a Codestream Registration box.....	187
Figure M.23bis – Layout of the Pixel Format Box	188
Figure M.24 – Organization of the contents of a Composition box.....	190
Figure M.25 – Organization of the contents of a Composition Options box	191
Figure M.26 – Organization of the contents of an Instruction Set box.....	191
Figure M.27 – Organization of the contents of an INST field within an Instruction Set box	192
Figure M.28 – Example of ROI specific metadata associated with one or more images.....	195
Figure M.29 – Example of Multiple XML documents associated with one or more images	195
Figure M.30 – Example of a Labelled XML document.....	195
Figure M.31 – Example of a labelled image.....	195
Figure M.32 – Organization of the contents of an Association box	196
Figure M.33 – Organization of the contents of a Number List box.....	196
Figure M.34 – Organization of the contents of a Label box	197
Figure M.35 – Organization of the contents of a Binary Filter box.....	197
Figure M.36 – Organization of the contents of the Desired Reproductions box.....	198
Figure M.37 – Organization of the contents of the Graphics Technology Standard Output box.....	198
Figure M.38 – Organization of the contents of the ROI Description box.....	199
Figure M.39 – Organization of the contents of a Digital Signature box.....	202
Figure M.40 – Organization of the contents of a MPEG-7 Binary box	204
Figure M.41 – Organization of the contents of a Compositing Layer Extensions box	205
Figure M.42 – Organization of the Compositing Layer Extensions Info box.....	207
Figure M.43 – Organization of the contents of a Multiple Codestream box.....	208
Figure M.44 – Organization of the contents of a Multiple Codestream Offsets box	208

Figure M.45 – Organization of the contents of a Grouping box	209
Figure M.46 – Organization of the contents of a decomposed XML box	210
Figure M.47 – Organization of the contents of a XML header box	210
Figure M.48 – Example Box layout for a Decomposed XML box	212
Figure N.1 – Organization of the contents of Image Creation box	215
Figure N.2 – Organization of the contents of Content Description box	216
Figure N.3 – Organization of the contents of History box	216
Figure N.4 – Organization of the contents of Intellectual Property Rights box	216
Figure N.5 – Organization of the contents of Image Identifier box	217
Figure N.6 – Schema of the Image Creation metadata	217
Figure N.7 – Schema of the General Creation Information metadata	218
Figure N.8 – Schema of the Camera Capture metadata	220
Figure N.9 – Schema of the Device Characterization metadata	221
Figure N.10 – Schema of the Spatial Frequency Response metadata	223
Figure N.11 – Schema of the Colour Filter Array Pattern metadata	224
Figure N.12 – Schema of the Opto-electronic Conversion Function metadata	225
Figure N.13 – Schema of the Camera Capture Settings metadata	227
Figure N.14 – Schema of the Scanner Capture metadata	230
Figure N.15 – Schema of the Scanner Settings metadata	231
Figure N.16 – Schema of the Software Creation metadata	231
Figure N.17 – Schema of the Captured Item metadata	232
Figure N.18 – Schema of the Reflection Print metadata	233
Figure N.19 – Schema of the Film metadata	234
Figure N.20 – Schema of the Content Description metadata	235
Figure N.21 – Schema of the Person Description metadata	236
Figure N.22 – Schema of the Thing Description metadata	237
Figure N.23 – Schema of the Organization Description metadata	238
Figure N.24 – Schema of the Event Description metadata	239
Figure N.25 – Schema of the Participant metadata	240
Figure N.26 – Schema of the Event Relationship metadata	241
Figure N.27 – Schema of the Audio metadata	241
Figure N.28 – Schema of the Property metadata	242
Figure N.29 – Schema of the Dictionary Definition metadata	243
Figure N.30 – Schema of the History metadata	243
Figure N.31 – Schema of the Processing Summary metadata	244
Figure N.32 – Schema of the Image Processing Hints metadata	246
Figure N.33 – Schema of the Previous metadata	247
Figure N.34 – Schema of the Image Reference metadata	248
Figure N.35 – Schema of the Intellectual Property Rights metadata	249
Figure N.36 – Schema of the IPR Names metadata	252
Figure N.37 – Schema of the IPR Description metadata	252
Figure N.38 – Schema of the IPR Dates metadata	253
Figure N.39 – Schema of the IPR Exploitation metadata	254
Figure N.40 – Schema of the IPR Management Systems metadata	255

Page

Figure N.41 – Schema of the IPR Identification metadata	256
Figure N.42 – Schema of the IPR Identifier metadata	256
Figure N.43 – Schema of the License Plate metadata	257
Figure N.44 – Schema of the IPR Contact Point metadata	257
Figure N.45 – Schema of the Image Identifier metadata	258
Figure N.46 – Schema of the non-negative double type	259
Figure N.47 – Schema of the rational type	259
Figure N.48 – Schema of the string including language attribute type	259
Figure N.49 – Schema of the degree type	260
Figure N.50 – Schema of the half degree type	260
Figure N.51 – Schema of the double size type	260
Figure N.52 – Schema of the integer size type	261
Figure N.53 – Schema of the DateTime type	262
Figure N.54 – Schema of the Address type	263
Figure N.55 – Schema of the Phone number type	264
Figure N.56 – Schema of the Email address type	265
Figure N.57 – Schema of the Web address type	266
Figure N.58 – Schema of the Person type	268
Figure N.59 – Schema of the Organization type	269
Figure N.60 – Schema of the Location type	270
Figure N.61 – Schema of the Coordinate location element	271
Figure N.62 – Schema of the Raw GPS Information element	273
Figure N.63 – Schema of the Raw GPS Information element (continued)	275
Figure N.64 – Schema of the Raw GPS Information element (concluded)	276
Figure N.65 – Schema of the Direction type	279
Figure N.66 – Schema of the Position type	280
Figure N.67 – Schema of the Point type	280
Figure N.68 – Schema of the Rect type	281
Figure N.69 – Schema of the Region type	282
Figure N.70 – Schema of the Product Details type	283
Figure N.71 – Schema of the Language attribute	283
Figure N.72 – Schema of the Timestamp attribute	283
Figure N.73 – Schema of the Comment element	284
Figure O.1 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = 123; I_0 = 2, d_0() = 31; I_S = 9, d_S() = 320300203$	347
Figure O.2 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = \underline{1}23; I_0 = 2, d_0() = 31; I_S = 9, d_S() = 320300203$	347
Figure O.3 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = 1\underline{2}3; I_0 = 2, d_0() = 31; I_S = 9, d_S() = 320300203$	348
Figure O.4 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = 12\underline{3}; I_0 = 2, d_0() = 31; I_S = 9, d_S() = 320300203$	348
Figure O.5 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = 123; I_0 = 2, d_0() = \underline{3}1; I_S = 9, d_S() = \underline{3}20300203$	348
Figure O.6 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = 123; I_0 = 2, d_0() = 3\underline{1}; I_S = 9, d_S() = 32\underline{0}300203$	348
Figure O.7 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = 123; I_0 = 2, d_0() = \underline{3}1; I_S = 9, d_S() = 32\underline{0}300203$	349
Figure O.8 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = 123; I_0 = 2, d_0() = 3\underline{1}; I_S = 9, d_S() = 320\underline{3}00203$	349
Figure O.9 – Sample wavelet decomposition: $N_L = 3; I_R = 3; d_R() = 123; I_0 = 2, d_0() = \underline{3}1; I_S = 9, d_S() = 3203\underline{0}0203$	349

Figure O.10 – Sample wavelet decomposition: $N_L = 3$; $I_R = 3$; $d_R() = 123$; $I_0 = 2$, $d_0() = \underline{31}$; $I_S = 9$, $d_S() = 320300\underline{203}$	349
Figure O.11 – Sample wavelet decomposition: $N_L = 3$; $I_R = 3$; $d_R() = 123$; $I_0 = 2$, $d_0() = \underline{31}$; $I_S = 9$, $d_S() = 320300\underline{203}$	350
Figure O.12 – Sample wavelet decomposition: $N_L = 3$; $I_R = 3$; $d_R() = 123$; $I_0 = 2$, $d_0() = \underline{31}$; $I_S = 9$, $d_S() = 320300\underline{203}$	350
Figure O.13 – Sample wavelet decomposition: $N_L = 3$; $I_R = 3$; $d_R() = 123$; $I_0 = 2$, $d_0() = \underline{31}$; $I_S = 9$, $d_S() = 320300\underline{203}$	350
Figure O.14 – Sample wavelet decomposition: $N_L = 3$; $I_R = 3$; $d_R() = 123$; $I_0 = 2$, $d_0() = \underline{31}$; $I_S = 9$, $d_S() = 320300\underline{203}$	351
Figure O.15 – FBI decomposition: $N_L = 5$; $I_R = 0$; $d_R() = 0$ (since $I_R = 0$, I_R and $d_R()$ get reset in Figure F.11 to $I_R = 5$ and $d_R() = 11111$); $I_0 = 4$, $d_0() = 2321$; $I_S = 17$, $d_S() = 1110111111111111$	351
Figure O.16 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = \underline{11111}$; $I_0 = 4$, $d_0() = 2321$; $I_S = 17$, $d_S() = 1110111111111111$	353
Figure O.17 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = \underline{11111}$; $I_0 = 4$, $d_0() = 2321$; $I_S = 17$, $d_S() = 1110111111111111$	354
Figure O.18 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = \underline{11111}$; $I_0 = 4$, $d_0() = 2321$; $I_S = 17$, $d_S() = 1110111111111111$	355
Figure O.19 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	356
Figure O.20 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	357
Figure O.21 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	358
Figure O.22 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	359
Figure O.23 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	360
Figure O.24 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	361
Figure O.25 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	362
Figure O.26 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	363
Figure O.27 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	364
Figure O.28 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	365
Figure O.29 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	366
Figure O.30 – FBI decomposition: $N_L = 5$; $I_R = 5$ and $d_R() = 11111$; $I_0 = 4$, $d_0() = \underline{2321}$; $I_S = 17$, $d_S() = \underline{1110111111111111}$	367
Figure O.31 – SPACL decomposition: $N_L = 4$; $I_0 = 2$, $d_0() = 21$; $I_R = 0$, $I_S = 0$	368
Figure O.32 – Component collection example	370
Figure O.33 – Original image components	370
Figure O.34 – Encoder multiple component transformation decisions	371
Figure O.35 – Decorrelation transformation array (MCC0 component collection 0 parameters)	372
Figure O.36 – Dependency transformation (MCC0 component collection 1 parameters)	372

	Page
Figure O.37 – Passing through intermediate components (MCC0 component collection 2 parameters)	372
Figure O.38 – Component collections in MCC0, transformation processing stage 0	373
Figure O.39 – Decorrelation transformation array (MCC1 component collection 0 parameters)	373
Figure O.40 – MCC1 component collection 1 (7 components passed through)	374
Figure O.41 – Component collections in MCC1, transformation processing stage 1	374
Figure O.42 – MCO marker segment for inverse multiple component transformation	374

Introduction

This Recommendation | International Standard defines a set of lossless (bit-preserving) and lossy compression methods for coding continuous-tone, bi-level, grey-scale, colour digital still images, or multi-component images.

This Recommendation | International Standard:

- specifies extended decoding processes for converting compressed image data to reconstructed image data;
- specifies an extended codestream syntax containing information for interpreting the compressed image data;
- specifies an extended file format;
- specifies a container to store image metadata;
- defines a standard set of image metadata;
- provides guidance on extended encoding processes for converting source image data to compressed image data;
- provides guidance on how to implement these processes in practice.

**INTERNATIONAL STANDARD
ITU-T RECOMMENDATION**

Information technology – JPEG 2000 image coding system: Extensions

1 Scope

This Recommendation | International Standard defines a set of lossless (bit-preserving) and lossy compression methods for coding continuous-tone, bi-level, grey-scale, colour digital still images, or multi-component images.

This Recommendation | International Standard:

- specifies extended decoding processes for converting compressed image data to reconstructed image data;
- specifies an extended codestream syntax containing information for interpreting the compressed image data;
- specifies an extended file format;
- specifies a container to store image metadata;
- defines a standard set of image metadata;
- provides guidance on extended encoding processes for converting source image data to compressed image data;
- provides guidance on how to implement these processes in practice.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- Recommendation ITU-T T.81 (1992) | ISO/IEC 10918-1:1994, *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines*.
- Recommendation ITU-T T.82 (1993) | ISO/IEC 11544:1993, *Information technology – Coded representation of picture and audio information – Progressive bi-level image compression*.
- Recommendation ITU-T T.84 (1996) | ISO/IEC 10918-3:1997, *Information technology – Digital compression and coding of continuous-tone still images: Extensions*, including Rec. ITU-T T.84 (1996)/Amd.1 (1999) | ISO/IEC 10918-3:1997/Amd.1:1999, *Information technology – Digital compression and coding of continuous-tone still images: Extensions – Amendment 1: Provisions to allow registration of new compression types and versions in the SPIFF header*.
- Recommendation ITU-T T.800 (2019) | ISO/IEC 15444-1:2019, *Information technology – JPEG 2000 image coding system: Core coding system*.
- Recommendation ITU-T T.805 | ISO/IEC 15444-6, *Information technology – JPEG 2000 image coding system: Compound image file format*.
- Recommendation ITU-T T.814 (2019) | ISO/IEC 15444-15:2019, *Information technology – JPEG 2000 image coding system: High-throughput JPEG 2000*.
- Recommendation ITU-T T.832 (2019) | ISO/IEC 29199-2:2020, *Information technology – JPEG XR image coding system – Image coding specification*.

2.2 Paired Recommendations | International Standards

- Recommendation ITU-T H.273 (in force), *Coding-independent code points for video signal type identification*.

ISO/IEC 23091-2: (in force), *Information technology – Coding-independent code points: Part 2: Video*.

2.3 Additional references

- Recommendation ITU-T T.42 (2003), *Continuous-tone colour representation method for facsimile*.
- Recommendation ITU-T T.45 (2000), *Run-length Colour Encoding*.
- IEC 61966-2-1 (in force), *Multimedia systems and equipment – Colour measurement and management: Part 2-1: Colour management – Default RGB colour space – sRGB*.
- IEC 61966-2-2:2003, *Multimedia systems and equipment – Colour measurement and management – Part 2-2: Colour management – Extended RGB colourspace – scRGB*.
- IETF RFC 1321 (1992), *The MD5 Message-Digest Algorithm*.
- IETF RFC 2630 (1999), *Cryptographic Message Syntax*.
- ISO 3166-1:2020, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*.
- ISO 3166-2:2020, *Codes for the representation of names of countries and their subdivisions – Part 2: Country subdivision code*.
- ISO 10126-2:1991, *Banking – Procedures for message encipherment (wholesale) – Part 2: DEA algorithm*.
- ISO 22028-2:2013, *Photography and graphic technology — Extended colour encodings for digital image storage, manipulation and interchange — Part 2: Reference output medium metric RGB colour image encoding (ROMM RGB)*.
- ISO 15076-1:2010, *Image technology colour management – Architecture, profile format and data structure – Part 1: Based on ICC.1:2010*.
- ISO/IEC 23001-1:2006, *Information technology – MPEG systems technologies – Part 1: Binary MPEG format for XML*.
- ISO/IEC 21122-1:2019, *Information technology – JPEG XS low-latency lightweight image coding system – Part 1: Core coding system*.
- ISO/IEC 60559:2020, *Information technology — Microprocessor Systems — Floating-Point arithmetic*.
- ANSI X9.30.2:1997, *Public Key Cryptography using Irreversible Algorithms – Part 2: The Secure Hash Algorithm (SHA-1)*.
https://infostore.saiglobal.com/en-us/standards/ANSI-X9-30-2-1997-1934_SAIG_ABA_ABA_5034/#
- Federal Information Processing Standard Publication (FIPS PUB) 186-4 (2013), *Digital Signature Standard (DSS)*.
<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>
- PIMA 7667:2001. *Photography - Electronics still picture imaging - Extended sRGB color encoding e-sRGB*.
https://www.imaging.org/ItemDetail?iProductCode=PIMA7667-2001&Category=DOWNLD_PDF&WebsiteKey=6d978a6f-475d-46cc-bcf2-7a9e3d5f8f82
- W3C Recommendation. *Extensible Markup Language (XML 1.0)*, fifth edition (26 November 2008).
<https://www.w3.org/TR/xml/>
- W3C Recommendation. *Namespaces in XML*, (14 January 1999).
<https://www.w3.org/TR/1999/REC-xml-names-19990114/>
- W3C Recommendation. *XML Schema Part 1: Structures*, second edition (28 October 2004).
<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>
- W3C Recommendation. *XML Schema Part 2: Datatypes*, second edition (28 October 2004).
<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply. The definitions defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, clause 3 also apply to this Recommendation | International Standard, except for the terms decomposition level, sub-band and resolution, which are redefined in this clause.

- 3.1 attribute:** XML construct that is a name-value pair extending or qualifying the meaning of an element.

- 3.2 cell:** Optional subdivision of a tile used for low-memory encoding and decoding.
- 3.3 component collection:** A subset of intermediate components used as inputs to a multiple component transformation stage, and a subset of intermediate components obtained as outputs from a multiple component transformation stage, where the subset's constituent components can occur in an arbitrary order, i.e., permuted with respect to their order of appearance in the set of input or output intermediate components.
- 3.4 component reconstruction array:** General term that refers to any of the following; decorrelation transformation array, dependency transformation array or offset array.
- 3.5 compositing:** Act of combining two compositing layers into a single, non-redundant set of image channels.
- 3.6 compositing layer:** Set of non-redundant channels drawn from one or more codestreams that shall be treated as a group.
- 3.7 deadzone:** Interval within which all sub-band coefficients are quantized to 0.
- 3.8 decomposition level:** Collection of sub-bands where each coefficient has the same spatial impact or span with respect to the original samples, including the LL, LH, HL, HH, LX, HX, XL and XH sub-band splits out of decomposition sublevels.
- 3.9 decomposition sub-level:** Collection of sub-bands that result from splits of a sub-band from a lower decomposition sub-level or splits of either LL, LX or XL sub-bands from a higher decomposition level.
- 3.10 decorrelation transformation array:** Array of coefficients that maps the input components of a component collection to the output components of the collection via a multiple component decorrelation transformation.
- 3.11 dependency transformation array:** Array of coefficients that maps the input components of a component collection to the output components of the collection via a multiple component dependency transformation.
- 3.12 element:** XML construct that consists of a start tag and an end tag with data enclosed within.
- 3.13 HX sub-band:** Sub-band obtained by forward horizontal high-pass analysis filtering and no vertical analysis filtering, and which contributes to reconstruction with inverse horizontal high-pass synthesis filtering and no vertical synthesis filtering.
- 3.14 intermediate component:** Single two-dimensional array of data involved in a stage of a multiple component transformation.
- 3.15 JPX baseline:** Subset of the JPX file format that uses codestream conforming to Rec. ITU-T T.800 | ISO/IEC 15444-1 and Rec. ITU-T T.801 | ISO/IEC 15444-2 only.
- 3.16 JPX baseline reader:** Application that correctly interprets all files that conform to the definition of a JPX baseline file.
- 3.17 JPX file:** Name of file in the file format described in this Recommendation | International Standard. Structurally, a JPX file is a contiguous sequence of boxes.
- 3.18 LX sub-band:** Sub-band obtained by forward horizontal low-pass analysis filtering and no vertical analysis filtering, and which contributes to reconstruction with inverse horizontal low-pass synthesis filtering and no vertical synthesis filtering.
- 3.19 metadata:** Additional data associated with the image data beyond the image data.
- 3.20 namespace:** Collection of names, identified by a URI, that allows XML documents of different sources to use the same element names within a single document to avoid element name conflicts.
- 3.21 offset array:** Array of coefficients containing offsets which are added to intermediate components during multiple component transformation of a component collection.
- 3.22 reconstructed image component:** Set of output intermediate components from the final transformation stage in the inverse multiple component transformation process.
- 3.23 rendered result:** Result generated by combining the compositing layers in the JPX file, either by composition or animation.
- 3.24 resolution:** Spatial relation of samples to a physical space.

NOTE – In this Recommendation | International Standard, the decomposition levels of the wavelet transformation create resolutions that differ by powers of two in either just horizontal, just vertical or both horizontal and vertical directions. The last (highest) decomposition level includes either an LL, LX or XL sub-band which is considered to be a lower resolution. Therefore, there is one more resolution level than decomposition levels.

3.25 sub-band: Group of transformation coefficients resulting from the sequence of low-pass and high-pass filtering operations, either just horizontally, just vertically or both horizontally and vertically.

3.26 spatially reconstructed component: Component which has been extracted from the codestream and passed through the decoding and inverse wavelet transformation process as specified by this Recommendation | International Standard, such that the set of spatially reconstructed components is the set of input components to the first transformation stage in the inverse multiple component transformation process.

3.27 transformation stage: Set of component collections and associated multiple component transformations.

3.28 visual masking: Mechanism where artefacts are masked by the image acting as a background signal.

3.29 XH sub-band: Sub-band obtained by no forward horizontal analysis filtering and vertical high-pass analysis filtering, and which contributes to reconstruction with vertical high-pass synthesis filtering and no inverse horizontal synthesis filtering.

3.30 XL sub-band: Sub-band obtained by no forward horizontal analysis filtering and vertical low-pass analysis filtering, and which contributes to reconstruction with vertical low-pass synthesis filtering and no inverse horizontal synthesis filtering.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply. The abbreviations defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, Clause 4 also apply to this Recommendation | International Standard.

ADS	Arbitrary Decomposition Styles
ATK	Arbitrary Transformation Kernels
CBD	Component Bit depth Definition
DCO	variable DC Offset
DFS	Downsample Factor Styles
DPI	Dots Per Inch
IPR	Intellectual Property Rights
MCC	Multiple Component transformation Collection
MCO	Multiple Component transformation Ordering
MCT	Multiple Component Transformation definition
NLT	Non-Linearity point Transformation
UUID	Universal Unique Identifier
VMS	Visual Masking

5 Conventions

In this Recommendation | International Standard, the word "shall" is used to express mandatory requirements for conformance to this Specification. When used to express a mandatory constraint on the values of syntax elements or the values of variables derived from these syntax elements, it is the responsibility of the encoder to ensure that the constraint is fulfilled. The word "may" is used to refer to behaviour that is allowed, but not necessarily required. The word "should" is used to refer to behaviour of an implementation that is encouraged to be followed under anticipated ordinary circumstances, but is not a mandatory requirement for conformance to this Specification.

For historical reasons, in this Specification, the convention of listing all informative annexes after the last normative annex is not followed strictly to minimize cross referencing issues in external texts.

6 General description

The purpose of this clause is to give an overview of this Recommendation | International Standard. Terms defined in previous clauses in this Recommendation | International Standard will also be introduced. (Terms defined in clauses 3 and 4 in Rec. ITU-T T.800 | ISO/IEC 15444-1 continue to apply in this Recommendation | International Standard.)

This Recommendation | International Standard defines a set of lossless (bit-preserving) and lossy compression methods for coding continuous-tone, bi-level, grey-scale, colour digital still images, or multi-component images. This set of methods

extends the elements in the core coding system described in Rec. ITU-T T.800 | ISO/IEC 15444-1. Extensions which pertain to encoding and decoding are defined as procedures which may be used in combination with the encoding and decoding processes described in Rec. ITU-T T.800 | ISO/IEC 15444-1. Each encoding or decoding extension shall only be used in combination with particular coding processes and only in accordance with the requirements set forth herein. These extensions are backward compatible in the sense that decoders which implement these extensions will also support configuration subsets that are currently defined by Rec. ITU-T T.800 | ISO/IEC 15444-1. This Recommendation | International Standard also defines extensions to the compressed data format, i.e., interchange format and the abbreviated formats.

6.1 Extensions specified by this Recommendation | International Standard

The following extensions are specified in this Recommendation | International Standard.

6.1.1 Syntax

An extension of the code stream syntax is described in Annex A. This extension provides all the codestream signalling in this Recommendation | International Standard. Further, it anticipates signalling needed for future specifications that include this Recommendation | International Standard as a normative reference. In addition to the codestream syntax defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, the following capabilities are supported: variable DC offset, variable scalar quantization, trellis coded quantization, visual masking, arbitrary decomposition, arbitrary transformation kernels, single sample overlap, multiple component transformations, non-linear transformation, arbitrary regions of interest. These extended markers conform to the same rules as the syntax in Rec. ITU-T T.800 | ISO/IEC 15444-1.

6.1.2 Variable DC offset

An extension which provides for variable DC offset is described in Annex B. Variable DC offset may be used to generate a better data distribution for input to the ICT or RCT multi component transformation, defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, and/or the wavelet transformation. Images with very skewed sample distributions may benefit from a non-default DC offset.

6.1.3 Variable scalar quantization

An extension that provides for variable scalar quantization is described in Annex C. This extension allows smaller or larger deadzones to be used with the scalar quantizer. This technique may improve visual appearance of low level texture.

6.1.4 Trellis coded quantization

An extension of the quantization is described in Annex D. This extension provides for trellis coded quantization (TCQ). The TCQ algorithm applies spatial-varying scalar quantization to its input sequence by choosing one of four scalar quantizers for each sample. Quantizer indices from supersets of these quantizers along with quantizer transitions in the form of a trellis provide all information necessary to reconstruct TCQ encoded wavelet coefficients.

6.1.5 Visual masking

An extension which provides for visual masking is described in Annex E. Visual masking is a mechanism where artefacts are masked by the image acting as a background signal. The main goal is to improve the image quality, especially for displays. The first effect of this technique is to improve the image quality, where the improvement becomes greater as the image becomes more complex. The second main effect of this technique is that for a given fixed bit-rate, the image quality is more robust against variations in image complexity. This is accomplished at the encoder via an extended non-linearity interposed between the transformation stage and the quantization stage.

6.1.6 Arbitrary decomposition

An extension providing for arbitrary decomposition of the tile component is described in Annex F. This extension can control the bandpass extent of wavelet sub-bands and thus provide control over the decorrelation process in order to tune compression performance. This extension also allows for transcoding of other wavelet based compression algorithms into codestreams of this Recommendation | International Standard.

6.1.7 Arbitrary wavelet transformation

Extensions that provide for transformation of image tile components using user defined wavelet filters are described in Annexes G and H. Annex G describes whole sample filters while Annex H describes arbitrary filters.

6.1.8 Single sample overlap discrete wavelet transformations

Extensions providing for block based wavelet transformations are described in Annex I. These extensions consist of one method for tile based wavelet transformation without tiling artefacts and one method for cell-based wavelet transformation.

6.1.9 Multiple component transformations

An extension which provides for multiple component transformations is described in Annex J. This extension specifies two types of multiple component transformations:

- 1) A specification of a multiple component transformation which uses linear transformations of bands to reduce the correlation of each band. This is similar to the most common colour transformations.
- 2) A specification of a wavelet transformation along the component direction.

6.1.10 Non-linear transformation

Annex K specifies three non-linear point transformations that are used after decoding processes and inverse multiple component transformations to map reconstructed values back to their proper range. These transformations may be employed by encoders prior to multiple component transformation and encoding to increase compression efficiency. The first two transformations, gamma and look-up table (LUT) style non-linearities, can be used to perceptually flatten a scanner or sensor with a linear response, from 12 bits to 8 bits precision prior to compression. The third, which maps sign-magnitude numbers into a two's complement representation, can be used to represent floating point numbers.

6.1.11 Region of interest

An extension which provides for Region of interest coding using the scaling based method is described in Annex L. The scaling based method provides for having different scaling values for different regions of interest. The extension also specifies how to generate the masks in the wavelet domain that describe the set of wavelet coefficient belonging to each Region of interest.

6.1.12 File format

An extension of the file format is described in Annex M. This extension provides for the exchange of compressed image files between application environments. This extension is an optional file format, called JPX, that applications may choose to use to contain JPEG 2000 compressed image data. JPX is an extension to the JP2 file format defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex I. The format:

- 1) specifies a binary container for both image and metadata;
- 2) specifies a mechanism to indicate image properties, such as the tone-scale or colour space of the image;
- 3) specifies a mechanism by which readers may recognize the existence of intellectual property rights information in the file;
- 4) specifies a mechanism by which metadata (including vendor specific information) can be included in files specified by this Recommendation | International Standard;
- 5) specifies a mechanism by which multiple codestreams can be combined into a single work, by methods such as compositing and animation.

6.1.13 Metadata definitions

Metadata definitions are described in Annex N. Metadata is additional information that is associated with the primary data (the image). In the context of this Specification, it is "additional data linked with the image data beyond the pixels which define the image". Metadata, to be most valuable for the owner(s) and user(s) of an image, needs to be consistently maintained throughout the image lifecycle. In today's environment of image editing applications, rapid transmission via the Internet, and high quality photographic printers, the lifecycle of a digital image may be very long as well as complex.

6.1.14 Progression order extensions

Beyond the five progression orders specified in Rec. ITU-T T.800 | ISO/IEC 15444-1, this Recommendation | International Standard introduces a position-resolution level-component-layer progression order, that is specified in Annex I.

6.2 Relation between extensions

The relations, at decoder side, between the extensions listed above are given in Figure 6-1. Technologies described in Rec. ITU-T T.800 | ISO/IEC 15444-1 are indicated in the boxes.

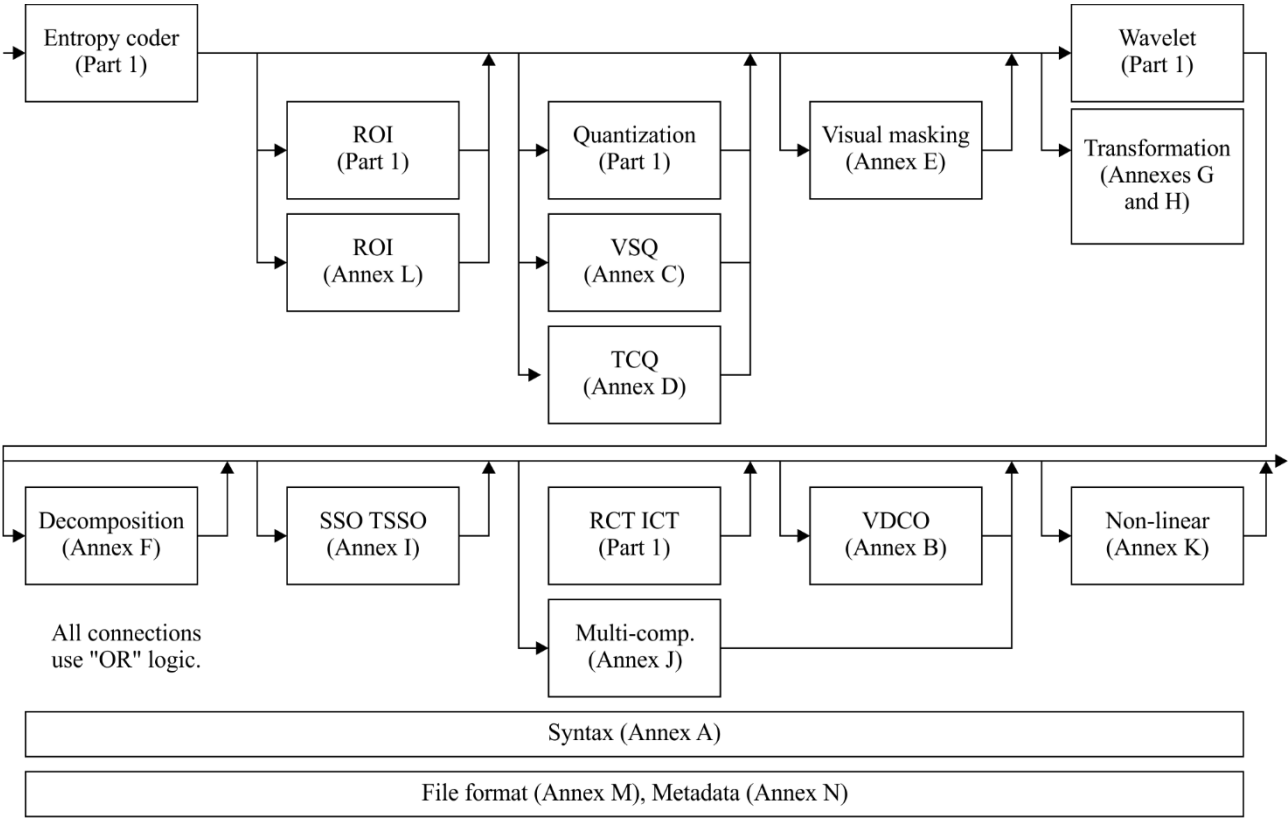


Figure 6-1 – Decoder block diagram

Annex A

Compressed data syntax, extension

(This annex forms an integral part of this Recommendation | International Standard.)

In this annex and all of its subclauses, the flow charts and tables are normative only in the sense that they define an output that alternative implementations shall duplicate.

This annex specifies the marker and marker syntax extensions to the Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A syntax. These markers provide all the codestream signalling in this Recommendation | International Standard. Further, it anticipates signalling needed for future specifications that include this Recommendation | International Standard as a normative reference.

All positive (unsigned) integer values of parameters are placed in the codestream as unsigned integers. All other (signed) integers are expressed in two's complement. Unless otherwise indicated, all values are in big endian order.

In the tables of this annex, the symbol "r" denotes bits that are reserved, and the symbol "x" denotes bits whose value can be either 0 or 1.

For codestreams conforming to this Recommendation | International Standard alone, the value of each bit denoted with an "r" shall be 0.

NOTE – The behaviour of implementations that conform to this Recommendation | International Standard is left unspecified when processing a codestream where the value of any bit denoted with an "r" is not 0.

A.1 Extended capabilities

The syntax in this annex supports the extensions in this Recommendation | International Standard. These marker segments conform to the same rules as the syntax in Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A. The addition of parameter values to some marker segments in Rec. ITU-T T.800 | ISO/IEC 15444-1 and the addition of new marker segments signals the information specific to the extensions in this Recommendation | International Standard. In every marker segment the first two bytes after the marker shall be an unsigned value that denotes the length in bytes of the marker segment parameters (including the two bytes of this length parameter but not the two bytes of the marker itself). When a marker segment that is not specified in this Recommendation | International Standard or in Rec. ITU-T T.800 | ISO/IEC 15444-1 is encountered in a codestream, the decoder shall use the length parameter to discard the marker segment. Table A.1 shows the marker segments affected by this Recommendation | International Standard.

Table A.1 – Syntax support for extensions

Extension	Extended Rec. ITU-T T.800 ISO/IEC 15444-1 marker segments	New marker segments
All extensions	SIZ	–
Variable DC offset	–	DCO
Variable scalar quantization	QCD, QCC, SOT	QPD, QPC
Trellis coded quantization	QCD, QCC, SOT	QPD, QPC
Visual masking	–	VMS
Single sample offset transform	SIZ, COD, COC	–
Arbitrary decomposition styles	COD, COC	DFS, ADS
Arbitrary transformation kernels	COD, COC	ATK
Multiple component transformation	COD	CBD, MCT, MCC, MCO
Non-linearity point transformation	–	NLT
Arbitrary shaped region of interest	RGN	–
Precinct length, tile-part header	–	RLT
Progression order extensions	COD, POC	–

A.2 Extensions to Rec. ITU-T T.800 | ISO/IEC 15444-1 marker segment parameters

This clause describes the extensions to marker segments defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A.

A.2.1 Image and tile size (SIZ), extended

The capability parameter Rsiz of Rec. ITU-T T.800 | ISO/IEC 15444-1 denotes capabilities that a decoder needs to properly decode a codestream. Table A.2 defines Rsiz values of this parameter specific to this Recommendation | International Standard.

Table A.2 – Capability Rsiz parameter, extended

Value (bits) MSB LSB	Capability
1x00 xxxx xxxx xxxx	At least one of the extended capabilities specified in this Recommendation International Standard is present
1x00 xxx0 xxxx xxx1	Variable DC offset capability is required to decode this codestream ^{a) b)}
1x00 xxxx xxxx xx1x	Variable scalar quantization capability is required to decode this codestream ^{a)}
1x00 xxxx xxxx x1xx	Trellis coded quantization capability is useful to decode this codestream ^{c)}
1x00 xxxx xxxx 1xxx	Visual masking capability is useful to decode this codestream ^{c)}
1x00 xxxx xxx1 xxxx	Single sample overlap capability is required to decode this codestream ^{a)}
1x00 xxxx xx1x xxxx	Arbitrary decomposition style capability is required to decode this codestream ^{a)}
1x00 xxxx x1xx xxxx	Arbitrary transformation kernel capability is required to decode this codestream ^{a)}
1x00 xxxx 1xxx xxxx	Whole sample symmetric transformation kernel capability is required to decode this codestream ^{a)}
1x00 xxx1 xxxx xxxx	Multiple component transformation capability is required to decode this codestream ^{a)}
1x00 xx1x xxxx xxxx	Non-linear point transformation capability is useful to decode this codestream ^{c)}
1x00 x1xx xxxx xxxx	Arbitrary shaped region of interest capability is required to decode this codestream ^{a)}
1x00 1xxx xxxx xxxx	Precinct-dependent quantization is required to decode this codestream ^{a)}
	All other values reserved
^{a)} "Required to decode" implies that no useful data or image can be reconstructed without the use of this capability. ^{b)} Shall not be used with the multiple component transformation. ^{c)} "Useful to decode" implies that use of this capability would improve the quality of the reconstructed data or image; however, the data or image may be decoded without its use. ^{d)} The 2 MSBs of Rsiz are used as follows: 10 – Rec. ITU-T T.801 ISO/IEC 15444-2 capabilities 11 – Rec. ITU-T T.801 ISO/IEC 15444-2 capabilities extended via a CAP marker segment.	

A.2.2 Start of tile-part (SOT) extended

If Rsiz indicates that the precinct-dependent quantization capability is used, then the SOT marker segment from Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A, is extended to allow 1-65535 tile-parts. Table A.3 replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.5, and Table A.4 replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.6.

Table A.3 – Start of tile-part parameter values, extended

Parameter	Size (bits)	Values
SOT	16	0xFF90
Lsot	16	10
Isot	16	0 to 65534
Psot	32	0, or $14 \cdot (2^{32} - 1)$
TPsot	16	0 to 65535
TNsot	16	Table A.4

Table A.4 – Number of tile-parts, TNsot, parameter value, extended

Value	Number of tile-parts
0	Number of tile-parts of this tile in the codestream is not defined in this header
1-65 535	Number of tile-parts of this tile in the codestream

A.2.3 Coding style (COD, COC), extended

Geometric manipulation is enabled with two bits in the Scod parameter shown in Table A.5.

Block coder extensions are enabled with 1 bit in the Scod parameter, as shown in Table A.5. When this bit is set, the COD marker segment is extended to include a 16-bit parameter SXcod that follows SPcod as shown in Figure A.1bis. The meaning of SXcod is defined by Table A.11bis.

Available progression orders are extended as shown in Table A.7bis, which replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.16.

If the position-resolution level-component-layer progression order is used, bit-14 of Ccap² shall be set (see Table A.49).



Figure A.1bis – Coding style default syntax

If the Rsiz field of the SIZ marker segment indicates that the multiple component transformations are used, then Table A.8 replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.17.

If the Rsiz field of the SIZ marker segment indicates that the arbitrary transformation kernels are used, then Table A.10 replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.20.

If the Rsiz field of the SIZ marker segment indicates that the single sample overlap transformation capability is necessary, then an extra 8 bit field is added to Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.13 after the transformation field as shown in Table A.7. The SSO values are found in Table A.11.

If the Rsiz field of the SIZ marker segment indicates that the arbitrary decomposition styles are used then the maximum number of decomposition levels field definitions are found in Table A.9 rather than in Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.13. This is shown in Table A.7.

Table A.5 – Coding style parameter values for the Scod parameter

Values (bits) MSB LSB	Coding style
rrxx xxx0	Entropy coder, precincts with PP _x = 15 and PP _y = 15
rrxx xxx1	Entropy coder with precincts defined below
rrxx xx0x	No SOP marker segments used
rrxx xx1x	SOP marker segments may be used
rrxx x0xx	No EPH marker used
rrxx x1xx	EPH marker shall be used
rrxx 0xxx rrxx 1xxx	Offset in the horizontal dimension, z _x = 0 (CBAP) Offset in the horizontal dimension, z _x = 1
rrx0 xxxx rrx1 xxxx	Offset in the vertical dimension, z _y = 0 (CBAP) Offset in the vertical dimension, z _y = 1
rr0x xxxx	Block coder extensions not used
rr1x xxxx	Block coder extensions defined by SXcod
	All other values reserved

Table A.6 – Coding style parameter values of the SGcod parameter

Parameters (in order)	Size (bits)	Values	Meaning of SGcod values
Progression order	8	Table A.7bis	Progression order
Number of layers	16	1 to 65535	Number of layers
Multiple component transformation	8	Table A.8	Multiple component transformation usage

Table A.7 – Coding style parameter values of the SPcod and SPcoc parameters, extended

Parameters (in order)	Size (bits)	Values (bits) MSB LSB	Meaning of SPcod values
Maximum number of decomposition levels	8	Table A.9	Decomposition mapping and levels
Code-block width	8	Rec. ITU-T T.800 ISO/IEC 15444-1, Table A.18	Code-block width exponent offset value, <i>xcb</i>
Code-block height	8	Rec. ITU-T T.800 ISO/IEC 15444-1, Table A.18	Code-block height exponent offset value, <i>ycb</i>
Code-block style	8	Rec. ITU-T T.800 ISO/IEC 15444-1, Table A.19	Style of the code-block coding passes
Transform	8	Table A.10	Wavelet transformation used
SSO overlap	16	Table A.11	SSO overlap values
Precinct size	variable	Rec. ITU-T T.800 ISO/IEC 15444-1, Table A.21	If Scod or Scoc = xxxx xxx0, this parameter is not present, otherwise this indicates precinct width and height. The first parameter (8 bits) corresponds to the N_{LL} , N_{LX} , or N_{XL} sub-band. Each successive parameter corresponds to each successive resolution in order.

Table A.7bis – Progression order for the SGcod and Ppoc parameters

Values (bits) MSB LSB	Multiple component transformation type
0000 0000	Layer-resolution level-component-position progression
0000 0001	Resolution level-layer-component-position progression
0000 0010	Resolution level-position-component-layer progression
0000 0011	Position-component-resolution level-layer progression
0000 0100	Component-position-resolution level-layer progression
0000 0101	Position-resolution level-component-layer progression
	All other values reserved.

Table A.8 – Multiple component transformation for the SGcod parameters

Values (bits) MSB LSB	Multiple component transformation type
0000 0000	No multiple component transformation specified.
0000 0001	Component transformation used on components 0, 1, 2 for coding efficiency. Irreversible component transformation used with irreversible filters. Reversible component transformation used with reversible filters.
0000 0x10	Array-based multiple component transformation is used. May be combined with wavelet-based multiple component transformation.
0000 01x0	Wavelet-based multiple component transformation is used. May be combined with array-based multiple component transformation.
	All other values reserved.

Table A.9 – Decomposition for the SPcod and SPcoc parameters, extended

Values (bits) MSB LSB	Decomposition type
0000 0000 to 0010 0000	Number of levels of wavelet decomposition, dyadic decomposition, N_L (Zero implies no transform.).
1000 0001 to 1111 1111	If in the main header: Downsampling factor style index value (1 to 127). (See clause A.3.3.) If in tile-part header: Arbitrary decomposition style index value (1 to 127). (See clause A.3.4.)
	All other values reserved

Table A.10 – Transformation for the SPcod and SPcoc parameters, extended

Values (bits) MSB LSB	Transformation type
0000 0000	Rec. ITU-T T.800 ISO/IEC 15444-1 9-7 irreversible wavelet transform.
0000 0001	Rec. ITU-T T.800 ISO/IEC 15444-1 5-3 reversible wavelet transform.
0000 0010 to 1111 1111	Arbitrary transformation kernel definition index value (2 to 255). Definitions are found in the appropriate ATK marker segment (see clause A.3.5).

Table A.11 – SSO parameters, extended

Values (bits) MSB LSB	SSO size (see Annex I)
0xxx xxxx xxxx xxxx 1xxx xxxx xxxx xxxx	SSO not used SSO used
x0xx xxxx xxxx xxxx x1xx xxxx xxxx xxxx	TSSO not used for every tile in the image (shall be the same in every COD marker segment) TSSO used for every tile in the image (shall be the same in every COD marker segment)
xx0x xxxx xxxx xxxx xx1x xxxx xxxx xxxx	Vovlp = 0 (shall be the same in every COD marker segment) Vovlp = 1 (shall be the same in every COD marker segment)
xxx0 xxxx xxxx xxxx xxx1 xxxx xxxx xxxx	Hovlp = 0 (shall be the same in every COD marker segment) Hovlp = 1 (shall be the same in every COD marker segment)
xxxx 0xxx xxxx xxxx x1xx 1xxx xxxx xxxx	TBDWT is not used (shall be the same in every COD marker segment) TBDWT is used (shall be the same in every COD marker segment)
xxxx xxxx xxxx 0000 to xxxx xxxx xxxx 1111	Cell width exponent value, $XC = 2^{\text{value}}$
xxxx xxxx 0000 xxxx to xxxx xxxx 1111 xxxx	Cell height exponent value, $YC = 2^{\text{value}}$
	All other values reserved

Table A.11bis – SXcod parameter

Values (bits) MSB LSB	Coding style
TTTT TTTT TTTT TT00	No modifications to selective arithmetic coder bypass mode.
TTTT TTTT TTTT TT01	Selective arithmetic coder bypass modified to start from the 2nd significant bit-plane if bit 0 of SPcod/SPcoc set, as specified in Annex P.
TTTT TTTT TTTT TT10	Selective arithmetic coder bypass modified to start from the 3rd significant bit-plane if bit 0 of SPcod/SPcoc set, as specified in Annex P.
TTTT TTTT TTTT TT11	Selective arithmetic coder bypass modified to start from the 4th significant bit-plane if bit 0 of SPcod/SPcoc set, as specified in Annex P.
	All other values reserved.

A.2.3bis Progression order change (POC), extended

Table A.11ter replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.32.

Available progression orders are extended as shown in Table A.7bis, which replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.16.

If the position-resolution level-component-layer progression order is used, bit-14 of Ccap² shall be set (see Table A.49).

Table A.11ter – Progression order change, tile parameter values

Parameter	Size (bits)	Values
POC	16	0xFF5F
Lpoc	16	9 to 65535
RSpoc ⁱ	8	0 to 33
CSpoc ⁱ	8	0 to 255; if Csiz < 257
	16	0 to 16383; Csiz ≥ 257
LYEpoc ⁱ	16	1 to 65535
REpoc ⁱ	8	(RSpoc ⁱ + 1) – 33
CEpoc ⁱ	8	(CSpoc ⁱ + 1) to 255, 0; if Csiz < 257
	16	(CSpoc ⁱ + 1) to 16384, 0; Csiz ≥ 257 (0 is interpreted as 256)
Ppoc ⁱ	8	Table A.7bis

A.2.4 Quantization (QCD, QCC), extended

If Rsiz indicates that the variable scalar quantization (see Annex C) capability is used, then the deadzone adjustment is signalled in modified QCD and QCC marker segments from Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A. If Rsiz indicates that trellis coded quantization is used, then these values are also signalled via the extended QCD and QCC marker segments from Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A. This shall only be used with irreversible transformations.

Table A.12 replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.28, and Table A.13 replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.30.

Table A.12 – Quantization default values for the Sqcd, Sqcc, Sqpd, and Sqpc parameters, extended

Values (bits) MSB LSB	Quantization style	SPqxx size (bits)	SPqxx usage
xxx0 0000	No quantization	8	Rec. ITU-T T.800 ISO/IEC 15444-1, Annex A
xxx0 0001	Scalar derived (values signalled for N_{LL} sub-band only). Use Rec. ITU-T T.800 ISO/IEC 15444-1, Equation E-5.	16	Rec. ITU-T T.800 ISO/IEC 15444-1, Annex A
xxx0 0010	Scalar expounded (values signalled for each sub-band). There are as many step sizes signalled as there are sub-bands.	16	Rec. ITU-T T.800 ISO/IEC 15444-1, Annex A
xxx0 0011	Variable deadzone and scalar derived (values signalled for N_{LL} sub-band only). Use Rec. ITU-T T.800 ISO/IEC 15444-1, Equation E-5.	32	Table A.13
xxx0 0100	Variable deadzone derived and scalar expounded (values signalled for each sub-band). There are as many step sizes signalled as there are sub-bands.	16	Table A.14 then Table A.15
xxx0 0101	Variable deadzone and scalar expounded (values signalled for each sub-band). There are as many step sizes signalled as there are sub-bands.	32	Table A.13
xxx0 1001	Trellis coded quantization derived (values signalled for N_{LL} sub-band only). Use Rec. ITU-T T.800 ISO/IEC 15444-1, Equation E-5.	16	Rec. ITU-T T.800 ISO/IEC 15444-1, Annex A
xxx0 1010	Trellis coded quantization expounded (values signalled for each sub-band). There are as many step sizes signalled as there are sub-bands.	16	Rec. ITU-T T.800 ISO/IEC 15444-1, Annex A
000x xxxx to 111x xxxx	Number of guard bits 0-7.		
	All other values reserved.		

Table A.13 – Quantization values (irreversible transformation only), extended

Values (bits) MSB LSB	Deadzone adjustment and quantization step size values
0000 0000 0000 0000 xxxx xxxx xxxx xxxx to 1111 1111 1111 1111 xxxx xxxx xxxx xxxx	Variable deadzone, num_nz _b , value −32768 to 32767 (see Equation C-1)
xxxx xxxx xxxx xxxx xxxx x000 0000 0000 to xxxx xxxx xxxx xxxx xxxx x111 1111 1111	Mantissa, μ _b , of the quantization step size value 0 to 2047 (see Rec. ITU-T T.800 ISO/IEC 15444-1, Equation E-3)
xxxx xxxx xxxx xxxx 0000 0xxx xxxx xxxx to xxxx xxxx xxxx xxxx 1111 1xxx xxxx xxxx	Exponent, ε _b , of the quantization step size value 0 to 31 (see Rec. ITU-T T.800 ISO/IEC 15444-1, Equation E-3)

Table A.14 – SPqcd, SPqcc, SPqpd, and SPqpc parameters (irreversible transformation only), extended

Values (bits) MSB LSB	Deadzone adjustment values (one for each sub-band)
0000 0000 0000 0000 to 1111 1111 1111 1111	First two bytes of SPqcx are the deadzone adjustment, num_nzb, value −32768 to 32767 (see Equation C-1)

Table A.15 – SPqcd, SPqcc, SPqpd, and SPqpc parameters (irreversible transformation only), extended

Values (bits) MSB LSB	Quantization step size values
xxxx x000 0000 0000 to xxxx x111 1111 1111	After first two bytes of SPqcx are the mantissa, μ_b , of the quantization step size value 0 to 2047 (see Rec. ITU-T T.800 ISO/IEC 15444-1, Equation E-3)
0000 0xxx xxxx xxxx to 1111 1xxx xxxx xxxx	After first two bytes of SPqcx are the exponent, ϵ_b , of the quantization step size value 0 to 31 (see Rec. ITU-T T.800 ISO/IEC 15444-1, Equation E-3)

A.2.5 Region of interest marker (RGN), extended

If Rsiz indicates that an arbitrary region of interest is used (see Annex L), then the description of a coefficient shift and a mask are signalled in a modified RGN marker segment from Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A. Table A.16 replaces Rec. ITU-T T.800 | ISO/IEC 15444-1, Table A.25. If there is RGN marker segment in the main header with a Srgn = 0, there shall not be any RGN marker segment anywhere in the codestream with a non-zero Srgn value for the component given by the corresponding Crgn value. Likewise, if there is RGN marker segment in the main header with a non-zero Srgn value, there shall not be any RGN marker segment anywhere in the codestream with a Srgn = 0 for the component given by the corresponding Crgn value.

When used in both the main header and the first tile-part header, the RGN in the first tile part header overrides the main for that tile. Also, an RGN specifying a single component (Crgn $\neq$ 65 535) overrides on specifying all components (Crgn = 65 535). Thus, the order of precedence is the following:

Tile-part RGN (Crgn $\neq$ 65 535) > Tile-part RGN (Crgn = 65 535) > Main RGN (Crgn $\neq$ 65 535) > Main RGN (Crgn = 65 535)

where the "greater than" sign, >, means that the greater overrides the lesser marker segment.

Table A.16 – Region-of-interest parameter values for the Srgn parameter

Values	ROI style (Srgn)	SPrgn usage
0	Implicit ROI (maximum shift)	Rec. ITU-T T.800 ISO/IEC 15444-1, Table A.26
1	Arbitrary region of interest, rectangle	Table A.18
2	Arbitrary region of interest, ellipse	Table A.18
	All other values reserved	

Table A.17 – Component index parameter value for the Crgn parameter

Parameter	Size (bits)	Values	Components index parameter
Component	16	0 to 16383 16394 to 65354 65535	Specifies component to which these region of interest descriptions apply Reserved Region of interest descriptions apply to all components

Table A.18 – Region-of-interest values from SPrgn parameter (Srgn = 1 or Srgn = 2)

Parameter	Size (bits)	Values	Meaning of SPrgn parameter
Binary shift	8	0 to 255	Binary shifting of coefficients in the region of interest above the background.
XArgn (left)	32	0 to ($2^{32} - 1$)	Horizontal reference grid point from the origin of the first point. (In the case of the ellipse, Srgn = 2, this value shall not exceed the width of the image.)
YArgn (top)	32	0 to ($2^{32} - 1$)	Vertical reference grid point from the origin of the first point. (In the case of the ellipse, Srgn = 2, this value shall not exceed the height of the image.)
XBrn (right)	32	0 to ($2^{32} - 1$)	Horizontal reference grid point from the origin of the second point.
YBrn (bottom)	32	0 to ($2^{32} - 1$)	Vertical reference grid point from the origin of the second point.

A.3 Extended marker segments

Table A.19 lists the markers specified in this Recommendation | International Standard.

Table A.19 – List of markers and marker segments

	Symbol	Code	Main header ^{a)}	Tile-part header ^{a)}
Variable DC offset	DCO	0xFF70	optional	optional
Visual masking	VMS	0xFF71	optional	optional
Downsampling factor style	DFS	0xFF72	optional	optional
Arbitrary decomposition style	ADS	0xFF73	optional	optional
Arbitrary transformation kernels	ATK	0xFF79	optional	optional
Component bit depth	CBD	0xFF78	optional	optional
Multiple component transformation definition	MCT	0xFF74	optional	optional
Multiple component transformation collection	MCC	0xFF75	optional	optional
Multiple component transformation ordering	MCO	0xFF77	optional	optional
Non-linearity point transformation	NLT	0xFF76	optional	optional
Quantization default, precinct	QPD	0xFF5A	optional	optional
Quantization component, precinct	QPC	0xFF5B	optional	optional
Precinct length, tile-part header	RLT	0xFF95	no	optional
^{a)} "optional" means it may be used in the header if this extension is used. "no" means it shall never be present in the header.				

A.3.1 Variable DC offset (DCO)

Function: Describes the variable DC offset for every component.

Usage: Present only if the variable DC offset capability bit in the Rsiz parameter (see clause A.2.1) is a one value. Main and first tile-part header of a given tile. Optional in both the main and tile-part headers. No more than one shall appear in any header. If present in the main header, it describes the variable DC offset for every component in every tile. If present in the first tile-part header of a given tile, it describes the variable DC offset for every component in that tile only. When used in both the main header and the first tile-part header, the DCO in the first tile part header overrides the main for that tile. Thus, the order of precedence is the following:

Tile-part DCO > Main DCO

where the "greater than" sign, >, means that the greater overrides the lesser marker segment.

Shall not be used with the multiple component transformation.

Length: Variable depending on the number of components. The syntax is depicted in Figure A.1.

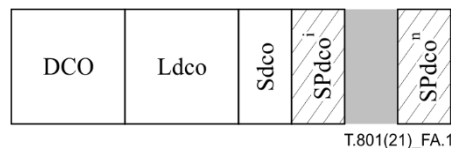


Figure A.1 – Variable DC offset syntax

DCO: Marker code. Table A.20 shows the size and parameter values for coding style component marker segment.

Ldco: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Ldco = \begin{cases} 3 + Csz & Sdco = 0 \\ 3 + 2 \cdot Csz & Sdco = 1 \\ 3 + 4 \cdot Csz & Sdco = 2 \\ 3 + 8 \cdot Csz & Sdco = 3 \end{cases} \quad (A-1)$$

where Csz is from Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A.

NOTE – If Ldco were to be larger than 65 535, then the DCO marker segment cannot be used. Instead, multiple component transformation functionality could be used.

Sdco: Variable DC offset type definition.

SPdcoⁱ: Variable DC offset for the *i*th component. There is one SPdco parameter for every component in the image.

Table A.20 – Variable DC offset parameter values

Parameter	Size (bits)	Values
DCO	16	0xFF70
Ldco	16	5 to 32770
Sdco	8	Table A.21
SPdco ⁱ	variable	Table A.21

Table A.21 – Variable DC offset parameter values for the Sdco parameter

Values (bits) MSB LSB	Offset type definition
0000 0000	Offsets are 8 bit unsigned integers
0000 0001	Offsets are 16 bit signed integers
0000 0010	Offsets are 32-bit binary floating point (ISO/IEC/IEEE 60559)
0000 0011	Offsets are 64-bit binary floating point (ISO/IEC/IEEE 60559)
	All other values reserved

IEEE numbers as used in the Rec. ITU-T T.801 | ISO/IEC 15444-2 codestream have to be written in big-endian order, using the following bit-assignments for encoding the floating point numbers:

SEEE EEEE EMMM MMMM MMMM MMMM MMMM MMMM

for single precision IEEE numbers, where S = sign bit, E = exponent bits, M = mantissa bits

SEEE EEEE EEEE EMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM

for double precision IEEE numbers, where S = sign bit, E = exponent bits, M = mantissa bits

NOTE 2 – This encoding requires typically an endian-swap on little-endian machines, and is typically the native encoding for big-endian machines.

A.3.2 Visual masking (VMS)

Function: Describes the visual masking for all tile-components in the image or tile.

Usage: Present only if the visual masking capability bit in the Rsiz parameter (see clause A.2.1) has the value one. Optionally used in the main and/or the first tile-part header of a given tile. No more than one VMS marker segment for a component shall appear in any header. When used in both the main header and the first tile-part header, the VMS marker segment in the first tile part header overrides the one in the main header for that tile. A VMS marker segment specifying a single component (Cvms ≠ 65 535) overrides on specifying all components (Cvms = 65 535). Thus, the order of precedence is the following:

Tile-part VMS (Cvms ≠ 65 535) > Tile-part VMS (Cvms = 65 535) > Main VMS (Cvms ≠ 65 535) > Main VMS (Cvms = 65 535)

where the "greater than" sign, >, means that the greater overrides the lesser marker segment.

Length: Fixed. The syntax is depicted in Figure A.2.

VMS	Lvms	Cvms	Svms	Wvms	Rvms	Avms	Bvms
-----	------	------	------	------	------	------	------

T.801(21)_FA.2

Figure A.2 – Visual masking syntax

- VMS:** Marker code. Table A.22 shows the size and parameter values for coding style, default marker segment.
- Lvms:** Length of marker segment in bytes (not including the marker). Fixed at 7 bytes.
- Cvms:** The index of the component to which this marker segment applies. Could be all components.
- Svms:** Minimal resolution level and respect block boundaries flag.
- Wvms:** Window width variable, *win_width* (see E.6).
- Rvms:** Bits retained variable, *bits_retained* (see E.6).
- Avms:** Value of the numerator of the α parameter, $\alpha = Avms/128$ (see E.6).
- Bvms:** Value of the numerator of the β parameter, $\beta = Bvms/128$ (see E.6).

Table A.22 – Visual masking parameter values

Parameter	Size (bits)	Values
VMS	16	0xFF71
Lvms	16	9
Cvms	16	Table A.23
Svms	8	Table A.24
Wvms	8	0 to 8
Rvms	8	0 to 255
Avms	8	0 to 255
Bvms	8	0 to 255

Table A.23 – Component parameter value for the Cvms parameter

Values	Component index parameter
0 to 16383	Specifies component to which these region of interest descriptions apply
16394 to 65354	Reserved
65535	Region of interest descriptions apply to all components

Table A.24 – Visual masking for the Svms parameters

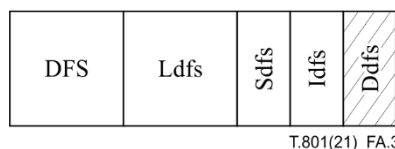
Values (bits) MSB LSB	Visual masking parameters
x000 0000 to x001 0000	Minimum resolution level value, <i>minlevel</i> (0 to 32) (see E.6)
0xxx xxxx 1xxx xxxx	Variable <i>respect_block_boundaries</i> = 0 (see E.6) Variable <i>respect_block_boundaries</i> = 1 (see E.6)
	All other values reserved

A.3.3 Downsampling factor styles (DFS)

Function: Describes the arbitrary decomposition pattern for the lowest resolution sub-band for all tiles of a given component.

Usage: Present only if the custom decomposition style bit in the Rsiz parameter (see clause A.2.1) is a one value. Main header. Assigned to a component by an index in the main header COD or COC markers.

Length: Variable. The syntax is depicted in Figure A.3.

**Figure A.3 – Downsampling factor styles syntax**

- DFS:** Marker code. Table A.26 shows the size and values of the symbol and parameters for coding style, default marker segment.
- Ldfs:** Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:
- $$Ldfs = 4 + \left\lceil \frac{Idfs}{4} \right\rceil \quad (A-2)$$
- Sdfs:** The index of this DFS marker segment. This marker segment is associated with a component via the parameter in the COD or COC marker segments found in the main header.
- Idfs:** Number of elements in the string defining the number of decomposition sub-levels.
- Ddfs:** String defining the number of decomposition sub-levels. The two bit elements are packed into bytes in big endian order. The final byte is padded to a byte boundary.

Table A.25 – Downsampling factor styles parameter values

Parameter	Size (bits)	Values
DFS	16	0xFF72
Ldfs	16	5 to 65535
Sdfs	16	0 to 15
Idfs	8	0 to 255
Ddfs	variable	String of elements

A.3.4 Arbitrary decomposition styles (ADS)

Function: Describes the arbitrary decomposition pattern for a tile-component or all tile-components within a single tile.

Usage: Present only if the custom decomposition style capability bit in the Rsiz parameter (see clause A.2.1) is a one value. Shall not be used to describe the decomposition described in Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex F. Main and first tile-part header of a given tile. There may be up to 127 such marker segments with unique index values. If an index value is found in a tile-part header, then it is used instead of an ADS marker segment in the main header with the same index value. These are assigned to a particular tile-component via the parameter in the COD or COC marker segments found only in a specific tile-part header.

Length: Variable. The syntax is depicted in Figure A.4.

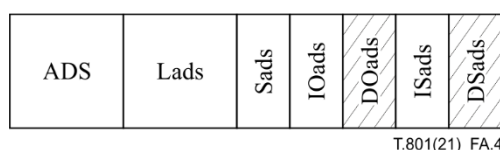


Figure A.4 – Arbitrary decomposition styles syntax

- ADS:** Marker code. Table A.26 shows the size and values of the symbol and parameters for coding style, default marker segment.
- Lads:** Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:
- $$Lads = 5 + \left\lceil \frac{IOads + ISads}{4} \right\rceil \quad (A-3)$$
- Sads:** The index of this ADS marker segment. This marker segment is associated with a component via the parameter in the COD or COC marker segments found in that tile-part header.
- IOads:** Number of elements in the string defining the number of decomposition sub-levels.
- DOads:** String defining the number of decomposition sub-levels. The two bit elements are packed into bytes in big endian order. The final byte is padded to a byte boundary.
- ISads:** Number of elements in the string defining the arbitrary decomposition structure.
- DSads:** String defining the arbitrary decomposition structure. The two bit elements are packed into bytes in big endian order. The final byte is padded to a byte boundary.

Table A.26 – Arbitrary decomposition styles parameter values

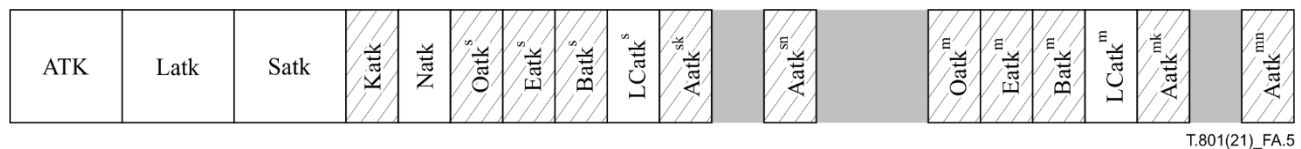
Parameter	Size (bits)	Values
ADS	16	0xFF73
Lads	16	3 to 65535
Sads	8	1 to 127
IOads	8	0 to 255
DOads	variable	String of elements
ISads	8	0 to 255
DSads	variable	String of elements

A.3.5 Arbitrary transformation kernels (ATK)

Function: Describes a transformation kernel and an index that allows assignment to tile-components.

Usage: Present only if the arbitrary transformation kernel capability bit in the Rsiz parameter (see clause A.2.1) is a one value. Main and first tile-part header of a given tile. May be up to 254 marker segments in any header. A marker segment in the tile-part header with the same index as one in the main header overrides the main header marker segment. Table A.27 describes the arbitrary transformation parameter values.

Length: Variable. The syntax is depicted in Figure A.5.

**Figure A.5 – Arbitrary transformation default syntax**

ATK: Marker code. Figure A.5 shows the size and values of the symbol and parameters for arbitrary transformation marker segment.

Latk: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$\text{Latk} = \begin{cases} 5 + 2\text{Natk} + \text{sizeof}(\text{Coeff\_Typ}) \left(1 + \sum_{s=0}^{\text{Natk}-1} \text{LCatk}^s \right) & \text{WT\_Typ} = \text{IRR}, \text{Filt\_Cat} = \text{ARB} \\ 5 + \text{Natk} + \text{sizeof}(\text{Coeff\_Typ}) \left(1 + \sum_{s=0}^{\text{Natk}-1} \text{LCatk}^s \right) & \text{WT\_Typ} = \text{IRR}, \text{Filt\_Cat} = \text{WS} \\ 5 + 3\text{Natk} + \text{sizeof}(\text{Coeff\_Typ}) \left(\text{Natk} + \sum_{s=0}^{\text{Natk}-1} \text{LCatk}^s \right) & \text{WT\_Typ} = \text{REV}, \text{Filt\_Cat} = \text{ARB} \\ 5 + 2\text{Natk} + \text{sizeof}(\text{Coeff\_Typ}) \left(\text{Natk} + \sum_{s=0}^{\text{Natk}-1} \text{LCatk}^s \right) & \text{WT\_Typ} = \text{REV}, \text{Filt\_Cat} = \text{WS} \end{cases} \quad (\text{A-4})$$

where $\text{sizeof}(\text{Coeff\_Typ})$ is the size (in bytes) of the Satk parameter that takes values of the type Coeff_Typ .

Satk: Index of the ATK marker segment; the type, Coeff_Typ , of the scaling factor and lifting step parameters; the wavelet filter category, Filt_Cat ; wavelet transformation type, WT_Typ , the initial odd or even subsequence, m_{init} .

Katk: The scaling factor, K . Present for irreversible transformation only, $\text{WT\_Typ} = \text{IRR}$.

Natk: Number of lifting steps, N_{LS} .

Oatk^s: Offset for lifting step s , off_s . The index, s , ranges from $s = 0$ to $\text{Natk} - 1$. Present only if $\text{Filt\_Cat} = \text{ARB}$.

Eatk^s: The base two scaling exponent for lifting step s , ε_s . Present only with reversible transformation, $\text{WT\_Typ} = \text{REV}$. The index, s , ranges from $s = 0$ to $\text{Natk} - 1$.

- Batk^s**: Additive residue for lifting step, s . Present for reversible transformations ($WT\_Typ = REV$) only. The index, s , ranges from $s = 0$ to $Natk - 1$.
- LCatk^s**: Number of lifting coefficients signalled for lifting step s . Provides the range, k , for $Aatk^{sk}$. The index, s , ranges from $s = 0$ to $Natk - 1$.
- Aatk^{sk}**: The k th lifting coefficient for the lifting step s , $\alpha_{s,k}$. The index, s , ranges from $s = 0$ to $Natk - 1$. The index, k , ranges from $k = 0$ to $LCatk - 1$.

Table A.27 – Arbitrary transformation parameter values

Parameter	Size (bits)	Values
ATK	16	0xFF79
Latk	16	9 to 65535
Satk	16	Table A.28
Katk	0 8 16 32 64 128	$WT\_Typ = REV$ $WT\_Typ = IRR, Coeff\_Typ = 0$ $WT\_Typ = IRR, Coeff\_Typ = 1$ $WT\_Typ = IRR, Coeff\_Typ = 2$ $WT\_Typ = IRR, Coeff\_Typ = 3$ $WT\_Typ = IRR, Coeff\_Typ = 4$
Natk	8	0 to 255
Oatk ^s	0 8	$Filt\_Cat = WS$ -128 to 127; $Filt\_Cat = ARB$
Eatk ^s	0 8	$WT\_Typ = IRR$ 0 to 255; $WT\_Typ = REV$
Batk ^s	0 8 16 32 64 128	$WT\_Typ = IRR$ $Coeff\_Typ = 0$ $Coeff\_Typ = 1$ $Coeff\_Typ = 2$ $Coeff\_Typ = 3$ $Coeff\_Typ = 4$
LCatk ^s	8	0 to 255
Aatk ^{sk}	8 16 32 64 128	$Coeff\_Typ = 0$ $Coeff\_Typ = 1$ $Coeff\_Typ = 2$ $Coeff\_Typ = 3$ $Coeff\_Typ = 4$

Table A.28 – Arbitrary transformation values for the Satk parameter

Values (bits) MSB LSB	Index
xxxx xxxx 0000 000x	Not available NOTE – The indices "0000 0000" and "0000 0001" are not available having been assigned to the 9-7 irreversible wavelet filter and the 5-3 reversible wavelet filter respectively in Rec. ITU-T T.800 ISO/IEC 15444-1, Annex A.
xxxx xxxx 0000 0010 to xxxx xxxx 1111 1111	Index of this marker segment (2 to 255)
xxxx x000 xxxx xxxx xxxx x001 xxxx xxxx xxxx x010 xxxx xxxx xxxx x011 xxxx xxxx xxxx x100 xxxx xxxx	Parameters 8-bit signed integer, <i>Coeff_Typ</i> = 0 Parameters 16-bit signed integer, <i>Coeff_Typ</i> = 1 Parameters 32-bit binary floating point (ISO/IEC/IEEE 60559), <i>Coeff_Typ</i> = 2 Parameters 64-bit binary floating point (ISO/IEC/IEEE 60559), <i>Coeff_Typ</i> = 3 Parameters 128-bit binary floating point (ISO/IEC/IEEE 60559), <i>Coeff_Typ</i> = 4
xxxx 0xxx xxxx xxxx r1xx 1xxx xxxx xxxx	Arbitrary filters, <i>Filt_Cat</i> = <i>ARB</i> WS filters, <i>Filt_Cat</i> = <i>WS</i>
rxx0 xxxx xxxx xxxx rxx1 xxxx xxxx xxxx	Irreversible filter, <i>WT_Typ</i> = <i>IRR</i> Reversible filter, <i>WT_Typ</i> = <i>REV</i>
rx0x xxxx xxxx xxxx rx1x xxxx xxxx xxxx	Modify even-indexed subsequence in first reconstruction step, $m_{init} = 0$ Modify odd-indexed subsequence in first reconstruction step, $m_{init} = 1$
r0xx 0xxx xxxx xxxx r1xx xxxx xxxx xxxx	Boundary extension method used in lifting steps is constant, <i>Exten</i> = <i>CON</i> Boundary extension method used in lifting steps is whole-sample symmetric, <i>Exten</i> = <i>WS</i>
	All other values reserved

A.3.6 Component bit depth definition (CBD)

Function: Defines the bit depth of reconstructed image components coming out of any multiple component transformation process.

Usage: Present only if the multiple component transformation capability bit in the Rsiz parameter (see clause A.2.1) is a one value. Main header. The CBD marker segment is required if the multiple component transformation processes are used. At most there can be one CBD in the main header.

The presence of a CBD marker segment in a codestream alters the procedures used to determine the precision of output image components and the interpretation of the SIZ marker. See Annex J for further details.

Length: Variable depending on the number of reconstructed image component bit depths signalled. The syntax is depicted in Figure A.6.

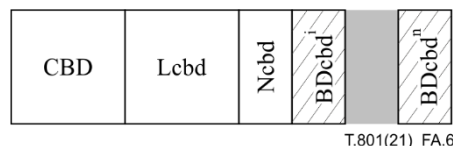


Figure A.6 – Component bit depth definition syntax

CBD: Marker code. Table A.29 shows the size and parameter values for component bit depth definition syntax.

Lcbd: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lmct = \begin{cases} 5 & Ncbd = 1xxxxxxxxxxxxx \\ 4 + Ncbd & Ncbd = 1xxxxxxxxxxxxx \end{cases} \quad (A-5)$$

Ncbd: Number of component bit depths included in marker segment. Table A.30 shows the value for the Ncbd parameter.

BDcbdⁱ: Bit depth and sign of the reconstructed image components in the order in which they are created as determined by the MCC and MCO marker segments. Either one value is signalled for all components (see Table A.30) or an individual bit depth is given for each component.

Table A.29 – Component bit depth definition parameter values

Parameter	Size (bits)	Values
CBD	16	0xFF78
Lcbd	16	5 to 16388
Ncbd	16	Table A.30
BDcbd ⁱ	8	Table A.31

Table A.30 – Component bit depth definition values for the Ncbd parameter

Values (bits) MSB LSB	Number of reconstructed image component bit depths in marker
x000 0000 0000 0001 to x100 0000 0000 0000	Number of reconstructed image components (1 to 16384)
0xxx xxxx xxxx xxxx 1xxx xxxx xxxx xxxx	Bit depths included, one per reconstructed image component One bit depth included, applies to <i>all</i> reconstructed image components
	All other values reserved

Table A.31 – Component bit depth definition values for the BDcbdⁱ parameter

Values (bits) MSB LSB	Reconstructed image component bit depths
x000 0000 to x010 0101	Component sample bit depth = value + 1. From 1 bit deep through 38 bits deep respectively.
0xxx xxxx	Component sample values are unsigned values
1xxx xxxx	Component sample values are signed values
	All other values reserved

A.3.7 Multiple component transformation definition (MCT)

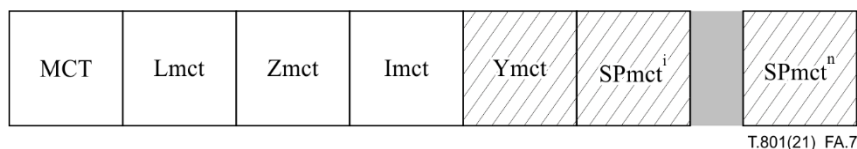
Function: Defines one multiple component transformation array per marker segment. The type and index of the array defined in this marker distinguishes it from other MCT marker segments in a given header. This array can be assigned to a collection of components within the MCC marker segment.

Usage: Present only if the multiple component transformation capability bit in the Rsiz parameter (see clause A.2.1) is a one value. Main and first tile-part header of a given tile. An MCT marker segment in a tile-part header overrides a main header MCT segment for that tile if and only if the ten low-order bits of the Imct fields of both marker segments are identical.

A series of MCT marker segments (defined as having the same Imct value in the same header and a Ymct > 0) shall all appear in the same header in order of (consecutive) Zmct parameter values.

To apply the transformation array included in an MCT marker segment, an MCC marker segment shall exist that associates the MCT marker segment with a component collection. This association is made through the array definition index of the MCT marker segment and the Tmccⁱ fields of MCC marker segments. If no such MCC marker segment exists, then the transformation array included in the MCT marker segment shall not be used in the decoding process.

Length: Variable depending on the size of the array. The syntax is depicted in Figure A.7.

**Figure A.7 – Multiple component transformation definition syntax**

- MCT:** Marker code. Table A.32 shows the size and parameter values for multiple component transformation definition marker segment.
- Lmct:** Length of marker segment in bytes (not including the marker).
- Zmct:** Index of this marker segment in a series of MCT marker segments. All the marker segments in the series have the same Imct parameter value present in this header. The data in each subsequent MCT marker segment shall be appended, in order, to make on stream of SPmctⁱ parameter values. The Ymct parameter values are present only in the first marker segment in the series (Zmct = 0).
- Imct:** Multiple component transformation index value, array type, and parameter size. An MCT marker segment, or series, with a given Imct value in the tile-part header overrides a main header MCT marker segment, or series, if and only if the ten low-order bits (index, and transformation type) of the Imct values of both markers are identical.
- Ymct:** Index of the last number of MCT marker segment in the series. For every series of MCT marker segments (i.e., MCT marker segments in this header with the same Imct parameter value), there shall be MCT marker segment with Zmct parameter values of 0 to Ymct. The last MCT marker segment will have Zmct = Ymct. This value is present only in the first marker segment in the series (Zmct = 0).
- SPmctⁱ:** Parameters for the multiple component transformation definition. One parameter value for each element in the array. See J.2 to determine the number of array elements and their order in the marker segment. The number of elements in a row and the number of rows (elements in a column) are determined by the type of array and the number of the input and output components to which it is assigned.

Table A.32 – Multiple component transformation definition parameter values

Parameter	Size (bits)	Values
MCT	16	0xFF74
Lmct	16	6 to 65535
Zmct	16	0 to 65535
Imct	16	Table A.33
Ymct	0 16	If Zmct > 0 0 to 65535
SPmct ⁱ	variable	Variable, Array of types as indicated in Table A.33

Table A.33 – Multiple component transformation definition values for the Imct parameter

Values (bits) MSB LSB	Index of the array definition, type and parameter type
rrrr xxxx 0000 0001 to rrrr xxxx 1111 1111	Index of the array definition, 1 to 255
rrrr xx00 xxxx xxxx	Dependency transformation array type
rrrr xx01 xxxx xxxx	Decorrelation transformation array type
rrrr xx10 xxxx xxxx	Offset array type
rrrr 00xx xxxx xxxx	Array elements are 16 bit signed integers
rrrr 01xx xxxx xxxx	Array elements are 32 bit signed integers
rrrr 10xx xxxx xxxx	Array elements are 32-bit binary floating point (ISO/IEC/IEEE 60559)
rrrr 11xx xxxx xxxx	Array elements are 64-bit binary floating point (ISO/IEC/IEEE 60559)
	All other values reserved

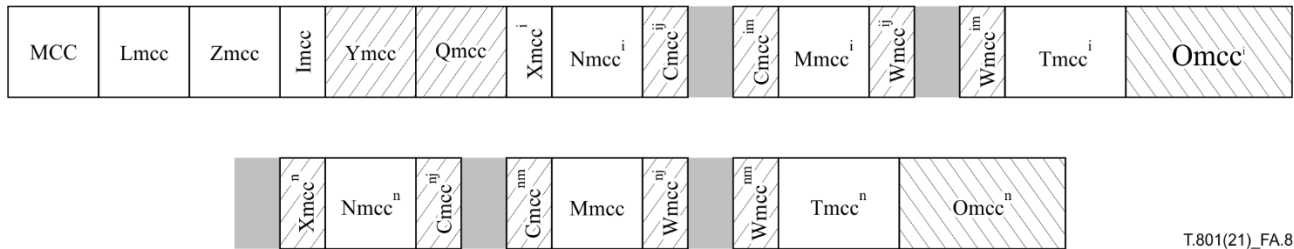
A.3.8 Multiple component transformation collection (MCC)

Function: Describes the collection of input intermediate components, the collection of output intermediate components, and the associated wavelets or arrays for a multiple component transformation. This marker segment can appear in the main header and can be referred to or overridden by an MCC marker in a tile-part header.

Usage: Present only if the multiple component transformation capability bit in the Rsiz parameter (see clause A.2.1) is a one value. Main and first tile-part header of a given tile. There may be up to 255 MCC marker segments, or series of marker segments, in the main header. There may be up to 255 MCC marker segments, or series of marker segments, in any tile-part header. An MCC marker segment in a tile-part header with the same index ($Imcc$) as one in the main header overrides the main header MCC segment for that tile.

A series of MCC marker segments (defined as having the same $Imcc$ value in the same header and a $Ymcc > 0$) shall all appear in the same header in order of (consecutive) $Zmcc$ parameter values.

Length: Variable depending on the number of component collections. The syntax is depicted in Figure A.8.



T.801(21)_FA.8

Figure A.8 – Multiple component collection syntax

- MCC:** Marker code. Table A.34 shows the size and parameter values for multiple component collection marker segment.
- Lmcc:** Length of marker segment in bytes (not including the marker).
- Zmcc:** Index of this marker segment in a series of MCC marker segments. All the marker segments in the series have the same $Imcc$ parameter value present in this header. The data in each subsequent MCC marker segment shall be appended, in order, to make one stream of the other parameters. The $Ymcc$ and $Qmcc$ parameter appears only in the first marker segment ($Zmcc = 0$).
- Imcc:** Index of this marker segment. An MCC marker segment, or series, with a given $Imcc$ value in the tile-part header overrides a main header MCC marker segment, or series, with the same $Imcc$ value.
- Ymcc:** Index of the last number of MCC marker segment in the series. For every series of MCC marker segments (i.e., MCC marker segments in this header with the same $Imcc$ parameter value), there shall be MCC marker segment with $Zmcc$ parameter values of 0 to $Ymcc$. The last MCC marker segment will have $Zmcc = Ymcc$. This value is present only in the first marker segment in the series ($Zmcc = 0$).
- Qmcc:** The number of collections in the MCC marker segment. This value is present only in the first marker segment in the series ($Zmcc = 0$).
- Xmccⁱ:** Indicates type of multiple component transformation used for the i th component collection (wavelet or array-based decorrelation or array-based dependency). Defines the interpretation applied to $Tmcc$.
- Nmccⁱ:** Indicates the number of input components for the i th component collection and defines the number of bits (8 or 16) used to represent the component indices in i th collection.
- Cmcc^{ij}:** Input intermediate component indices included the i th component collection. The number of indices in the i th component collection is $Nmcc^i$. Each index denotes an input intermediate component. The order of the indices defines the ordering applied to the input intermediate components prior to application of the inverse transform.
- Mmccⁱ:** Indicates the number of output intermediate components for the i th component collection and defines the number of bits (8 or 16) used to represent the component indices in i th collection. If anything, other than an array-based irreversible decorrelation transform is used, $Mmcc^i$ shall equal $Nmcc^i$.
- Wmcc^{ij}:** Intermediate component indices included the i th output component collection. The number of indices in the i th component collection is $Mmcc^i$. All output intermediate component indices in a given MCC marker segment shall appear only once across all collections in that MCC marker.
- Tmccⁱ:** For array-based component collection transforms, $Tmcc^i$ assigns arrays defined in an MCT marker segment to the i th component collection. An MCT marker segment with the right type and index in the first tile-part header of a tile is used before an MCT marker segment with the right type and index in the main header. $Tmcc^i$ also indicates the reversibility of array-based component transforms.
- For wavelet-based component collection transforms, $Tmcc^i$ assigns a wavelet kernel defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, Annex A or an ATK marker segment and the number of wavelet

decomposition levels for the i th component collection (only the dyadic decomposition of Rec. ITU-T T.800 | ISO/IEC 15444-1 is supported). An ATK marker segment with the proper index in the first tile-part header of a tile is used before an ATK marker segment with the proper index in the main header. $Tmcc^i$ also contains the index of an MCT marker segment that contains component additive offsets.

Omccⁱ: Present in the MCC marker segment only for those component collections that use a wavelet-based transform. $Omcc^i$ indicates the reference grid offset to apply in the component dimension for the i th component collection (see J.2.2).

Table A.34 – Multiple component collection parameter values

Parameter	Size (bits)	Values
MCC	16	0xFF75
Lmcc	16	5 to 65535
Zmcc	16	0 to 65535
Imcc	8	0 to 255
Ymcc	0 16	If $Zmcc > 0$ 0 to 65535
Qmcc	0 16	If $Zmcc > 0$ 0 to 16383
$Xmcc^i$	8	Table A.35
$Nmcc^i$	16	Table A.36
$Cmcc^{ij}$	8 16	0 to 255 0 to 16383
$Mmcc^i$	16	Table A.37
$Wmcc^{ij}$	8 16	0 to 255 0 to 16383
$Tmcc^i$	24	Table A.38 or Table A.39
$Omcc^i$	32	0 to 4294967295

Table A.35 – Multiple component collection values for the $Xmcc^i$ parameter

Values (bits) MSB LSB	Coding style	$Tmcc^i$ parameter
rrrr rr00	Component collection transform is array-based dependency transform	Table A.38
rrrr rr01	Component collection transform is array-based decorrelation transform	Table A.38
rrrr rr11	Component collection transform is wavelet-based transform	Table A.39
	All other values reserved	

Table A.36 – Multiple component collection values for the $Nmcc^i$ parameter

Values (bits) MSB LSB	Coding style
0xxx xxxx xxxx xxxx to 1xxx xxxx xxxx xxxx	Input component collection indices ($Cmcc^i$) are 8 bit integers Input component collection indices ($Cmcc^i$) are 16 bit integers
x000 0000 0000 0001 to x100 0000 0000 0000	Number of input components in i th component collection (1 to 16384)

Table A.37 – Multiple component collection values for the Mmccⁱ parameter

Values (bits) MSB LSB	Coding style
0xxx xxxx xxxx xxxx to 1xxx xxxx xxxx xxxx	Output component collection indices (Wmcc ⁱ) are 8 bit integers Output component collection indices (Wmcc ⁱ) are 16 bit integers
x000 0000 0000 0001 to x100 0000 0000 0000	Number of output components in ith component collection (1 to 16384)

Table A.38 – Multiple component collection values for the Tmccⁱ parameter (array-based)

Values (bits) MSB LSB	Coding style
rrrr rrrx xxxx xxxx 0000 0000 rrrr rrrx xxxx xxxx 0000 0001 to rrrr rrrx xxxx xxxx 1111 1111	No assigned transform array (NULL array) Decorrelation or dependency transform array index
rrrr rrrx 0000 0000 xxxx xxxx rrrr rrrx 0000 0001 xxxx xxxx to rrrr rrrx 1111 1111 xxxx xxxx	No assigned offset array (NULL array) Offset array index
rrrr rrr0 xxxx xxxx xxxx xxxx rrrr rrr1 xxxx xxxx xxxx xxxx	Decorrelation or dependency transform is irreversible Decorrelation or dependency transform is reversible
	All other values reserved

Table A.39 – Multiple component collection values for the Tmccⁱ parameter (wavelet-based)

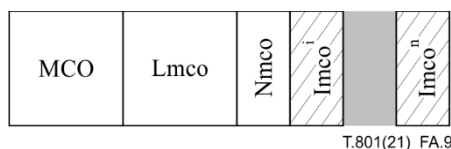
Values (bits) MSB LSB	Coding style
xxxx xxxx xxxx xxxx 0000 0000 xxxx xxxx xxxx xxxx 0000 0001 xxxx xxxx xxxx xxxx 0000 0010 to xxxx xxxx xxxx xxxx 1111 1111	Rec. ITU-T T.800 ISO/IEC 15444-1 9-7 irreversible filter Rec. ITU-T T.800 ISO/IEC 15444-1 5-3 reversible filter Index of the ATK marker segment containing the wavelet kernel for component collection
xxxx xxxx 0000 0000 xxxx xxxx xxxx xxxx 1111 1111 xxxx xxxx	Index of MCT marker segment containing additive offsets for component collection
xx00 0000 xxxx xxxx xxxx xxxx xx10 0000 xxxx xxxx xxxx xxxx	Number of dyadic wavelet decomposition levels used in the component collection (0 to 32)
	All other values reserved

A.3.9 Multiple component transformation ordering (MCO)

Function: Describes the order in which multiple component transformations are applied during inverse multiple component transformation processing.

Usage: Present only if the multiple component transformation capability bit in the Rsiz parameter (see clause A.2.1) is a one value. At most one MCO marker segment in main and first tile-part header of a given tile. If used in the main header, this marker segment defines the default ordering of multiple component transformation stages for all tiles. If used in the first tile-part header, then the component transformation order established by the MCO marker segment overrides any default ordering defined by a main header MCO marker segment.

Length: Variable depending on the number of multiple component transformation stages used. The syntax is depicted in Figure A.9.

**Figure A.9 – Multiple component transformation ordering syntax**

MCO: Marker code. Table A.40 shows the size and parameter values for the multiple component transformation ordering marker segment.

Lmco: Length of marker segment in bytes (not including the marker). The length is given by the following expression:

$$Lmco = 3 + Nmco \quad (A-6)$$

Nmco: Number of multiple component transformation stages specified for inverse transform processing. If $Nmco = 0$, then no multiple component transformation processing is used for the current tile and no $Imco^i$ parameters shall appear. Otherwise, $Nmco$ specifies the number of MCC marker segment identifiers that will follow.

$Imco^i$: Index of the MCC marker segment containing the component collection information for the i^{th} inverse multiple component transformation stage (see clause A.3.8).

Table A.40 – Multiple component intermediate collection parameter values

Parameter	Size (bits)	Values
MCO	16	0xFF77
Lmco	16	3 to 258
Nmco	8	0 to 255
$Imco^i$	8	0 to 255

A.3.10 Non-linearity point transformation (NLT)

Function: Describes either a gamma or LUT non-linearity to be applied to a single component or all components.

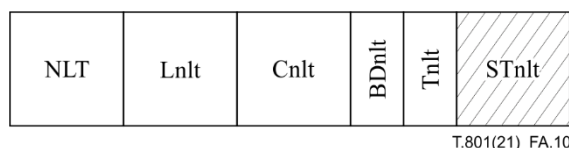
Usage: Present only if the non-linearity point transformation capability bit in the Rsiz parameter (see clause A.2.1) is a one value. Main and first tile-part header of a given tile. There may be no more than one marker segment per component plus one default in any header.

When used in the main header, the defined non-linearity can be established as a default for all components or established as a default for a single component. When used in a tile-part header, it can be used to establish a default for all components in the tile or to set the non-linearity transformation for a single component in that tile. Thus, the order of precedence is the following:

Tile-part NLT > Tile-part NLT default > Main NLT > Main NLT default

where the "greater than" sign, >, means that the greater overrides the lesser marker segment.

Length: Variable depending on the value of Tnlt. The syntax is depicted in Figure A.10.

**Figure A.10 – Non-linearity point transformation syntax**

NLT: Marker code. Table A.41 shows the size and values of the symbol and parameters for non-linearity point transformation marker segment.

Lnlt: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lnl = 6 + \begin{cases} 0 & Tnlt = 0 \text{ or } Tnlt = 3 \\ 15 & Tnlt = 1 \\ 11 + (N_{points} \cdot \Psi_{Tval}) & Tnlt = 2 \end{cases}, \quad (A-7)$$

$$\Psi_{Tval} = \begin{cases} 1 & PTval \in [1,8] \\ 2 & PTval \in [9,16] \\ 4 & PTval \in [17,32] \end{cases}$$

NOTE – Specifying a non linearity transform with Tnlt = 0 allows a transformation to be explicitly disabled.

- Cnlt:** The index of the component to which this marker segment relates. The components are indexed 0, 1, 2, etc. If this value is 65 535, then this marker segment applies to all components. Table A.42 shows the value for the Cnlt parameter.
- BDnlt:** Bit depth and sign of the decoded image component, Z_i , after processing of the i th reconstructed image component by the non-linearity. If Cnlt = 65 535, then this value applies to all components. Table A.43 shows the values for the BDnlt parameter.
- Tnlt:** Non-linearity type. Table A.44 shows the value for the Tnlt parameter.
- STnlt:** Parameter values associated with the non-linearity as controlled by the Tnlt field.

Table A.41 – Non-linearity transformation parameter values

Parameter	Size (bits)	Values
NLT	16	0xFF76
Lnl	16	12 to 65535
Cnlt	16	Table A.42
BDnlt	8	Table A.43
Tnlt	8	Table A.44
STnlt	variable	Table A.44

Table A.42 – Non-linearity transformation parameter values for the Cnlt parameter

Values	Components index parameter
0-16 383	Defines component to which these non-linearity transformation descriptions in this marker segment apply
65 535	Non-linearity transformation descriptions in this marker segment apply to all components
	All other values reserved

Table A.43 – Decoded image component bit depth parameter values for the BDnlt parameter

Values (bits) MSB LSB	Decoded image component bit depth
x000 0000 to x010 0101	Component sample bit depth = value + 1. From 1 bit deep through 38 bits deep respectively.
0xxx xxxx	Component sample values are unsigned values
1xxx xxxx	Component sample values are signed values
	All other values reserved

Table A.44 – Non-linearity transformation parameter values of the Tnlt parameter

Values (bits) MSB LSB	Meaning of Tnlt values	STnlt usage
0000 0000	No non-linearity transformation applied	–
0000 0001	Gamma-style non-linearity transformation	Table A.45
0000 0010	LUT-style non-linearity transformation	Table A.46
0000 0011	Binary Complement to Sign Magnitude Conversion	–
	All other values reserved	–

Table A.45 – Non-linearity transformation parameter values of the STnlt parameter (Tnlt = 1)

Parameter (in order)	Size (bits)	Values	Meaning of STnlt values
E	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65535\}}{65535}$	Non-linearity exponent (8-bit integer + 16-bit fraction)
S	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65535\}}{65535}$	Non-linearity toe slope (8-bit integer + 16-bit fraction)
T	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65535\}}{65535}$	Non-linearity threshold (8-bit integer + 16-bit fraction)
A	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65535\}}{65535}$	Non-linearity continuity parameter A (8-bit integer + 16-bit fraction)
B	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65535\}}{65535}$	Non-linearity continuity parameter B (8-bit integer + 16-bit fraction)
NOTE – E, S, and A parameters shall not be zero as the non linearity transformation defined by Equation K-2 is otherwise not well-defined.			

Table A.46 – Non-linearity transformation parameter values of the STnlt parameter (Tnlt = 2)

Parameters (in order)	Size (bits)	Values	Meaning of STnlt values
Npoints	16	1 to 8191	(Number of points – 1) in the LUT-style non-linearity definition (all other values reserved)
Dmin	32	0 to $(2^{32}-1)$	Dmin = parameter value / $(2^{32}-1)$
Dmax	32	1 to $(2^{32}-1)$	Dmax = parameter value / $(2^{32}-1)$
PTval	8	0000 0001 to 0010 0000	Precision of Tvalue parameter in bits (1-32). This also implies how many bytes are used to express the Tvalue (all other values reserved)
Tvalue	8, PTval ≤ 8; 16, 9 ≤ PTval ≤ 16 32, PTval > 16	variable	Run of table values for the LUT-style non-linearity. The (Npoints + 1) parameters are unsigned integers. The actual value of Tvalue is Tvalue = parameter value / $(2^{PTval} - 1)$

A.3.11 Quantization default, precinct (QPD)

Function: Describes the quantization default used for compressing all components of a particular resolution level and precinct. The parameter values can be overridden for an individual component, resolution level, and precinct by a QPC marker segment which, if present, shall appear in a tile-part header prior to any packets for that component, resolution level, and precinct.

Usage: Main and any tile-part header. Several QPD marker segments may appear in any tile-part header, but only one for each resolution level and precinct. If a QPD is used in a tile-part header it overrides the quantization characteristics defined by either QCD or QCC marker segments for all components of the resolution level and precinct indexed by the QPD within the scope of the particular tile. Thus, the quantization characteristics of a particular resolution level, precinct pair is determined by the presence of QCD, QCC, QPD or QPC markers in the following order of precedence:

Any tile-part QPC > Any tile-part QPD > First tile-part QCC > First tile-part QCD >
Main QPC > Main QPD > Main QCC > Main QCD

When QPD marker segments are used, they shall appear in tile-part headers before any packets are found for the indexed resolution level and precinct.

Length: Variable depending on the number of quantized sub-bands within the resolution level indexed. The syntax is depicted in Figure A.11.

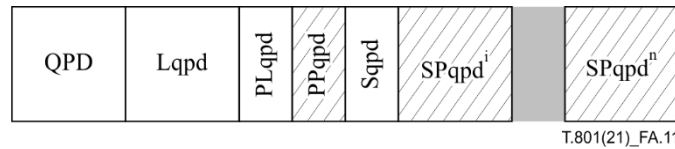


Figure A.11 – Quantization default, precinct syntax

QPD: Marker code. Table A.47 shows the size and values of the symbol and parameters for quantization default, precinct marker segment.

Lqpd: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lqpd = \begin{cases} 5 + \text{num\_subbands\_lev} & \text{no\_quant AND PLqpd} < 128 \\ 7 & \text{quant\_derived AND PLqpd} < 128 \\ 5 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND PLqpd} < 128 \\ 6 + \text{num\_subbands\_lev} & \text{no\_quant AND PLqpd} \geq 128 \\ 8 & \text{quant\_derived AND PLqpd} \geq 128 \\ 6 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND PLqpd} \geq 128 \end{cases} \quad (\text{A-8})$$

where *num_subbands_lev* can be derived from clause F.2.4 for each resolution level and whether this marker segment has *no_quant*, *quant_derived*, and *quant_expounded* is signalled in the *Sqpd* parameter.

NOTE – The *Lqpd* can be used to determine how many quantization step sizes are present in the marker segment. However, there is not necessarily a correspondence with the number of sub-bands present because the sub-bands can be truncated with no requirement to correct this marker segment.

PLqpd: The resolution level index for the quantization values signalled. Equation A-9 shows how this marker segment is constructed based on the resolution level index, *lev*, as well as the precinct index, *prec*.

$$PLqpd = \begin{cases} lev & prec < 256 \\ 128 + lev & prec \geq 256 \end{cases} \quad (\text{A-9})$$

The resolution level index, *lev*, can range from 0 to N_L , where N_L is the number of decomposition levels defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, clause A.6.1.

PPqpd: The precinct index for the quantization values signalled. The size of this marker segment parameter will be one byte when the *PLqpd* parameter is less than 128, but two bytes when *PLqpd* is greater than or equal to 128. This parameter will then just hold the precinct index, *prec*. The precinct index, *prec*, can range from 0 to *numprecincts* – 1, where *numprecincts* is the number of precincts at resolution level *lev* and is also defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, clause B.6.

Sqpd: Quantization style for all components at the resolution level, *lev*, and precinct, *prec*.

SPqpdⁱ: Quantization step size value for the *i*th sub-band at the resolution level, *lev*, in the order defined for *lev* in F.2.4. The number of parameters is at least as large as the number of sub-bands in the tile-component with the greatest number of sub-bands at resolution level, *lev*.

Table A.47 – Quantization default, precinct parameter values

Parameter	Size (bits)	Values
QPD	16	0xFF5A
Lqpd	16	6 to 101
PLqpd	16	0 to 32 or 128 to 160
PPqpd	8 16	0 to 255, PLqpd < 128 0 to 65535, PLqpd ≥ 128
Sqpd	8	Table A.12
SPqpd ⁱ	variable	Table A.12

A.3.12 Quantization precinct component (QPC)

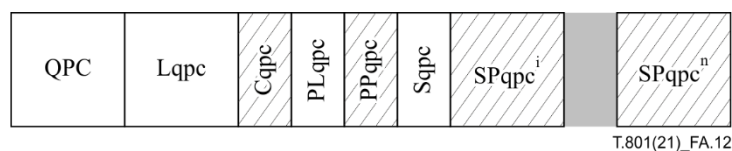
Function: Describes the quantization used for compressing a particular component, resolution level, and precinct.

Usage: Main and any tile-part header. Several QPC marker segments may appear in any tile-part header, but only one for each component, resolution level, and precinct. If a QPC is used in a tile-part header it overrides the quantization characteristics defined by QCD, QCC, or QPD marker segments for the triplet indexed by the QPC within the scope of the particular tile. Thus, the quantization characteristics of a particular component, resolution level, and precinct is determined by the presence of QCD, QCC, QPD or QPC markers in the following order of precedence:

Any tile-part QPC > Any tile-part QPD > First tile-part QCC > First tile-part QCD >
Main QPC > Main QPD > Main QCC > Main QCD

When QPC marker segments are used, they shall appear in tile-part headers before any packets are found for the indexed component, resolution level, and precinct.

Length: Variable depending on the number of quantized sub-bands within the resolution level indexed. The syntax is depicted in Figure A.12.

**Figure A.12 – Quantization precinct component syntax**

QPC: Marker code. Table A.48 shows the size and values of the symbol and parameters for quantization component marker segment.

Lqpc: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lqpc = \begin{cases} 6 + \text{num\_subbands\_lev} & \text{no\_quant AND Csiz} < 257 \text{ AND Plqpc} < 128 \\ 8 & \text{quant\_derived AND Csiz} < 257 \text{ AND Plqpc} < 128 \\ 6 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND Csiz} < 257 \text{ AND Plqpc} < 128 \\ 7 + \text{num\_subbands\_lev} & \text{no\_quant AND Csiz} \geq 257 \text{ AND Plqpc} < 128 \\ 9 & \text{quant\_derived AND Csiz} \geq 257 \text{ AND Plqpc} < 128 \\ 7 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND Csiz} \geq 257 \text{ AND Plqpc} < 128 \\ 7 + \text{num\_subbands\_lev} & \text{no\_quant AND Csiz} < 257 \text{ AND Plqpc} \geq 128 \\ 9 & \text{quant\_derived AND Csiz} < 257 \text{ AND Plqpc} \geq 128 \\ 7 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND Csiz} < 257 \text{ AND Plqpc} \geq 128 \\ 8 + \text{num\_subbands\_lev} & \text{no\_quant AND Csiz} \geq 257 \text{ AND Plqpc} \geq 128 \\ 10 & \text{quant\_derived AND Csiz} \geq 257 \text{ AND Plqpc} \geq 128 \\ 8 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND Csiz} \geq 257 \text{ AND Plqpc} \geq 128 \end{cases} \quad (\text{A-10})$$

where *num_subbands_lev* can be derived from clause F.2.4 for each resolution level and whether this marker segment has *no_quant*, *quant_derived*, and *quant_expounded* is signalled in the *Sqpc* parameter.