INFORMATION TECHNOLOGY

JPEG 2000 IMAGE CODING SYSTEM: COMFORMANCE TESTING

STUDY DOCUMENT JPEG 2000 PART IV CD, JULY 2001

THE ISO AND ITU WILL PROVIDE COVER PAGES.

NOTES to National Bodies:

Comments are explicitly solicited on the following issues:

1. What should the decoder minimum image size be for compliance classes 1 & 2 in table A-1? In stockholm we changed 160x120 for the low resolution to be 128x128, because it was a square and a power of two, rather than a specific application size like 160x120 or 176x144. It seems appropriate to likewise change the sizes in the other compliance classes to be square powers of 2.

2. What should the number of components be for compliance class 2 in table A-1?

3. Should compliance class and profile be "linked" together and only some combinations be allowed?

Currently Part 1, Amendment 1 defines 2 profiles, and Annex A defines 3 compliance classes (levels) which can operate with any profile. Thus it is possible to be complaint (P0,C0), (P0,C1), (P0,C2), (P1,C0), (P1,C1), or (P1,C2). While the profiles are nested, and the complaince class is nested, the combination is not! Thus there are images which a (P0,C1) decoder cannot decode but a (P1,C0) decoder can, but there are images that a (P0,C1) decoder decodes better than the (P1,C0) decoder.

A simple solution would be to leave the definitions as they are, but only allow compliance to be claimed at certain "interoperability points."

Point A: P0,C0

Point B: P0,C1

Point C: P1,C1

Point D: P1, C2

(The only other reasonable sequence would be to make Point B (P1,C0).) This allows there to be only one label "on the box." and the labels are nested, point C is better in every dimension than a point B.

4. Should file format compliance be required for some "interoperability" point?

Once again this

Help is need on the following items:

1. A decoder (or more than one) which can be set to "compliance class 0, 1 or 2" and stop decoding after the right number of code-blocks (Ncb), coded data (Lbody), number of bitplanes (M), etc.

2. Test codestream generation (Paris meeting, and email to gormish@rii.ricoh.com). Multiple sources of codestreams are good!

3. Analysis of the "equivalent" precision of the 5-3 filter when used to perform an inverse 9-7. (We agreed at stockholm to set the precision for the Cclass 0 9-7 wavelet to be such that it could be implemented with the 5-3 for at least the 3 levels of wavelet required by Cclass 0). Thus how shale we set the RMS error and peak error in Annex C?

TABLE OF CONTENTS

Foreword

Forward to be supplied by ISO and ITU.

Introduction

The scope of the ISO/IEC 15444 is a specification that describes an image compression system that allows great flexibility, not only for the compression of images but also for the access into the codestream. The codestream provides a number of mechanisms for locating and extraction portions of the compressed image data for the purpose of retransmission, storage, display, or editing. This access allows storage and retrieval of compressed image data appropriate for a given application without decoding.

This Recommendation | International Standard provides the framework, concepts, and methodology for testing and the criteria to be achieved to claim conformance to ISO/IEC 15444-1 standard. The objective of standardization in this field to promote interpretability between JPEG 2000 encoders and decoders and test these systems for compliance to this specification. Conformance testing is the testing of a candidate product for the existence of specific characteristics required by a standard in order to determine the extent to which that product is a conforming implementation. It involves testing the capabilities of an implementation against both the conformance requirements in the relevant standard and the statement of the implementation’s capability.

A framework of an abstract test suite (ATS) is standardized for relevant profiles and levels of ISO/IEC 15444-1. The standardization of ATS requires international definition and acceptance of a common test methodology, together with appropriate test methods and procedures. The purpose of this Recommendation | International Standard is to define this methodology, to provide a framework for specifying ATS, and to define the procedures to be followed during a conformance testing.

Test methods are also addressed in this Recommendation | International Standard however, any organization contemplating the use of test methods defined in this Recommendation | International Standard should carefully consider the constraints on their applicability. Conformance testing does not include robustness testing, acceptance testing, and performance testing.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY –

JPEG 2000 IMAGE CODING SYSTEM: PART IV COMPLIANCE TESTING

1 Scope

This Recommendation | International Standard specifies the framework, concepts, and methodology for testing and criteria to be achieved to claim conformance to ITU-T Recommendation T.800 | ISO/IEC 15444-1. It provides a framework for specifying abstract test suites (ATS) and for defining the procedures to be followed during conformance testing.

This Recommendation | International Standard

— specifies compliance testing procedures for encoding and decoding using JPEG 2000 Part 1 (ITU-T Recommendation T.800 | ISO/IEC 15444-1)

— specifies codestreams, decoded images, and error metrics to be used with the testing procedures

— specifies abstract test suites

— provides guidance for creating an encoder compliance test

— provides guidance for creating compliant subsystems of JPEG 2000

This Recommendation | International Standard does not include the following tests:

Acceptance testing: the process of determining whether an implementation satisfies acceptance criteria and enables the user to determine whether to accept the implementation. This includes the planning and execution of several kinds of tests (e.g., functionality, quality, and speed performance testing) that demonstrates that the implementation satisfies the users requirements.

Performance testing: measures the performance characteristics of an Implementation Under Test (IUT) such as its throughput, responsiveness, etc., under various conditions.

Robustness testing: the process of determining how well an IUT processes data which contains errors.

2 References

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. 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

- ITU-T Recommendation T.800 | ISO/IEC 15444-1, Information technology — JPEG 2000 Image Coding System.
- ITU-T Recommendation T.800 | ISO/IEC 15444-1 AMD1, Information technology — JPEG 2000 Image Coding System, Amendment 1.
- ITU-T Recommendation T.800 | ISO/IEC 15444-1 AMD2, Information technology — JPEG 2000 Image Coding System, Amendment 2.
- ITU-T Recommendation T.800 | ISO/IEC 15444-1 Corrigendum 1, Information technology — JPEG 2000 Image Coding System.
- ITU-T Recommendation T.800 | ISO/IEC 15444-1 Corrigendum 2, Information technology — JPEG 2000 Image Coding System.

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply. The definitions from ITU-T T.800 | IS 15444-1 section 3 also apply to this Recommendation | International Standard.

3.1 **Abstract Test Suite:** Generic compliance testing concepts and procedures for a given requirement.

3.2 **arithmetic coder:** An entropy coder that converts variable length strings to variable length codes (encoding) and visa versa (decoding).

3.3 **bit:** A contraction of the term “binary digit”; a unit of information represented by a zero or a one.

3.4 **bit-depth:** The number of bits that required to represent an original component of an image.

3.5 **bit-plane:** A two dimensional array of bits. In this Recommendation | International Standard a bit-plane refers to all the bits of the same magnitude in all coefficients or samples. This could refer to a bit-plane in a component, tile-component, code-block, region of interest, or other.

3.6 **bit stream:** The actual sequence of bits resulting from the coding of a sequence of symbols. It does not include the markers or marker segments in the main and tile-part headers or the EOC marker. It does include any packet headers and in stream markers and marker segments not found within the main or tile-part headers.

3.7 **box:** A portion of the file format defined by a length and unique box type. Boxes of some types may contain other boxes.

3.8 **buffer:** Computer storage space required to hold data as it is being processed.

3.9 **byte:** Eight bits.

3.10 **Cclass:** generic application groups that define guaranteed performance levels required to be compliant with a JPEG 2000 codestream

3.11 **channel:** One logical component of the image. A channel may be a direct representation of one component from the codestream, or may be generated by the application of a palette to a component from the codestream.

3.12 **cleanup pass:** A coding pass performed on a single bit-plane of a code-block of coefficients. The first pass and only coding pass for the first significant bit-plane is a cleanup pass; the third and the last pass of every remaining bit-plane is a cleanup pass.

3.13 **code-block:** A rectangular grouping of coefficients from the same subband of a tile-component.

3.14 **codestream:** A collection of one or more bit streams and the main header, tile-part headers, and the EOC required for their decoding and expansion into image data. This is the image data in a compressed form with all of the signalling needed to decode.

3.15 **coder:** An embodiment of either an encoding or decoding process.

3.16 **coding pass:** A complete pass through a code-block where the appropriate coefficient values and context are applied. There are three types of coding passes: significance propagation pass, magnitude refinement pass and cleanup pass. The result of each pass (after arithmetic coding, if selective arithmetic coding bypass is not used) is a stream of compressed image data.

3.17 **coefficient:** The values that are result of a transformation.

3.18 **colour channel:** A channel that functions as an input to a colour transformation system. For example, a red channel or a greyscale channel would be a colour channel.

3.19 **component:** A two-dimensional array of samples. A image typically consists of several components, for instance representing red, green, and blue.

3.20 **compressed image data:** Part or all of a codestream. Can also refer to a collection of bit streams in part or all of a codestream.

3.21 **conformance:** Fulfilment of the specified requirements, as defined in this specification, for a given profile and Cclass.

3.22 **conforming reader:** An application that reads and interprets a JP2 file correctly.

Editor's note: Need to review and see if compliant reader is the same thing and chose one.

3.23 **conformance test procedure:** The process of assessing compliance.

3.24 **context:** Function of coefficients previously decoded and used to condition the decoding of the present coefficient.

3.25 **context label:** The arbitrary index used to distinguish different context values. The labels are used as a convenience of notation rather than being normative.

3.26 **context vector:** The binary vector consisting of the significance states of the coefficients included in a context.

3.27 **decoder:** An embodiment of a decoding process, and optionally a colour transformation process.

3.28 **decoding process:** A process which takes as its input all or part of a codestream and outputs all or part of a reconstructed image.

3.29 **decomposition level:** A collection of wavelet subbands where each coefficient has the same spatial impact or span with respect to the source component samples. These include the HL, LH, and HH subbands of the same two dimensional subband decomposition. For the last decomposition level the LL subband is also included.

3.30 **discrete wavelet transformation (DWT)**: A transformation that iteratively transforms one signal into two or more filtered and decimated signals corresponding to different frequency bands. This transformation operates on spatially discrete samples.

3.31 **encoder:** An embodiment of an encoding process.

3.32 **encoding process:** A process, that takes as its input all or part of a source image data and outputs a codestream.

3.33 **Executable test suite:** Set of executable test cases that support the abstract test cases.

3.34 **file format:** A codestream and additional support data and information not explicitly required for the decoding of codestream. Examples of such support data include text fields providing titling, security and historical information, data to support placement of multiple codestreams within a given data file, and data to support exchange between platforms or conversion to other file formats.

3.35 **fully decode:** Applying ITU-T T.800 | IS 15444-1 to produce an image from a codestream where all coded data in the codestream has been used to produce the image.

3.36 **grid resolution:** The spatial resolution of the reference grid, specifying the distance between neighboring points on the reference grid.

Editor's note: Reference grid is used but grid resolution is not but will it be in the file format.

3.37 **guard bits:** Additional most significant bits that have been added to sample data.

3.38 **header:** Either a part of the codestream that contains only markers and marker segments (main header and tile part header) or the signalling part of a packet (packet header).

3.39 **HH subband:** The subband obtained by forward horizontal high-pass filtering and vertical high-pass filtering. This subband contributes to reconstruction with inverse vertical high-pass filtering and horizontal high-pass filtering.

3.40 **HL subband:** The subband obtained by forward horizontal high-pass filtering and vertical low-pass filtering. This subband contributes to reconstruction with inverse vertical low-pass filtering and horizontal high-pass filtering

3.41 **image:** The set of all components.

3.42 **image data:** The components and component samples making up an image. Image data can refer to either the source image data or the reconstructed image data.

3.43 **Implementation:** A realization of a specification.

3.44 **Implementation Conformance Statement (ICS):** Statement of specification options and the extent to which they have been implemented buy a implementation under test.

3.45 **Implementation Under Test (IUT):** An implementation which is being evaluated for compliance.

3.46 **irreversible:** A transformation, progression, system, quantization, or other process that, due to systemic or quantization error, disallows lossless recovery. An irreversible process can only lead to lossy compression.

3.47 **irreversible filter:** A particular filter pair used in the wavelet transformation. This irreversible filter pair has 9 taps in the low-pass and 7 taps in the high-pass.

3.48 **JP2 file:** The name of a file in the file format described in this specification. Structurally, a JP2 file is a contiguous sequence of boxes.

3.49 **JPEG**: Joint Photographic Experts Group - The joint ISO/ITU committee responsible for developing standards for continuos-tone still picture coding. It also refers to the standards produced by this committee: ITU-T.81 | ISO/IEC 10918-1, ITU-T T.83 | ISO/IEC 10918-2, ITU-T T.84 | ISO/IEC 10918-2 and ITU-T T.87 | ISO/IEC 14495.

3.50 **LH subband**: The subband obtained by forward horizontal low-pass filtering and vertical high-pass filtering. This subband contributes to reconstruction with inverse vertical high-pass filtering and horizontal low-pass filtering.

3.51 **LL subband**: The subband obtained by forward horizontal low-pass filtering and vertical low-pass filtering. This subband contributes to reconstruction with inverse vertical low-pass filtering and horizontal low-pass filtering.

3.52 **layer:** A collection of compressed image data from coding passes of one, or more, code-blocks of a tile-component. Layers have an order for encoding and decoding that must be preserved.

3.53 **level:** A limitation on parameters within a given profile for a give set of applications.

3.54 **lossless:** A descriptive term for the effect of the overall encoding and decoding processes in which the output of the decoding process is identical to the input to the encoding process. Distortion free restoration can be assured. All of the coding processes or steps used for encoding and decoding are reversible.

3.55 **lossy:** A descriptive term for the effect of the overall encoding and decoding processes in which the output of the decoding process is not identical to the input to the encoding process. There is distortion (measured mathematically). At least one of the coding processes or steps used for encoding and decoding is irreversible.

3.56 **magnitude refinement pass:** A type of coding pass.

3.57 **main header:** A group of markers and marker segments at the beginning of the codestream that describe the image parameters and coding parameters that can apply to every tile and tile-component.

3.58 **marker:** A two-byte code in which the first byte is hexadecimal FF (0xFF) and the second byte is a value between 1 (0x01) and hexadecimal FE (0xFE).

3.59 **marker segment:** A marker and associated (not empty) set of parameters.

3.60 **mod:** mod(y,x) = z, where z is an integer such that, and such that y-z is a multiple of x.

3.61 **packet:** A part of the codestream comprising a packet header and the compressed image data from one layer of one precinct of one resolution level of one tile-component.

3.62 **packet header:** Portion of the packet that contains signalling necessary for decoding that packet.

3.63 **parser:** reads and identifies components of the codestream down to the code-block level

3.64 **partial decoding**: Producing an image from a subset of an entire codestream.

3.65 **precinct:** A one rectangular region of a transformed tile-component, within each resolution level, used for limiting the size of packets.

3.66 **precision:** Number of bits allocated to a particular sample, coefficient, or other binary numerical representation.

3.67 **progression:** The order of a codestream where the decoding of each successive bit contributes to a "better" reconstruction of the image. What metrics make the reconstruction "better" is a function of the application. Some examples of progression are increasing resolution or improved sample fidelity.

3.68 **profile:** A subset of technology, from ITU-T T.800 | IS 15444-1: JPEG 2000, that meets the needs of a given application with limits on parameters within a selected technology.

3.69 **quantization:** A method of reducing the precision of the individual coefficients to reduce the number of bits used to entropy code them. This is equivalent to division while compressing and multiplying while decompressing. Quantization can be achieved by an explicit operation with a given quantization value (scalar quantization) or by dropping (truncating) coding passes from the codestream.

3.70 **reconstructed image:** An image, that is the output of a decoder.

3.71 **reconstructed sample:** A sample reconstructed by the decoder. This always equals the original sample value in lossless coding but may differ from the original sample value in lossy coding.

3.72 **reference grid:** A regular rectangular array of points used as a reference for other rectangular arrays of data. Examples include components and tiles.

3.73 **region of interest (ROI**): A collections of coefficients that are considered of particular relevance by some user defined measure.

3.74 **reversible:** A transformation, progression, system, or other process that does not suffer systemic or quantization error and, therefore, allows lossless signal recovery.

3.75 **reversible filter:** A particular filter pair used in the wavelet transformation. This reversible filter pair has 5 taps in the low-pass and 3 taps in the high-pass.

3.76 **sample:** One element in the two-dimensional array that comprises a component.

3.77 **selective arithmetic coding bypass:** A coding style where some of the code-block passes are not coded by the arithmetic coder. Instead the bits to be coded are appended directly to the bit stream without coding.

3.78 **shift:** Multiplication or division of a number by powers of two.

3.79 **sign bit:** A bit that indicates whether a number is positive (zero value) or negative (one value).

3.80 **sign-magnitude notation:** A binary representation of an integer where the distance from the origin is expressed with a positive number and the direction from the origin (positive or negative) is expressed with a separate single sign bit.

3.81 **significance propagation pass:** A coding pass performed on a single bit-plane of a code-block of coefficients.

3.82 **significance state:** State of a coefficient at a particular bit-plane. If a coefficient, in sign-magnitude notation, has the first magnitude 1 bit at, or before, the given bit-plane it is considered "significant." If not, it is considered "insignificant."

3.83 **source image:** An image used as input to an encoder.

3.84 **subband:** A group of transform coefficients resulting from the same sequence of low-pass and high-pass filtering operations, both vertically and horizontally.

3.85 **subband decomposition:** A transformation of an image tile-component into subbands.

3.86 **Testing:** The process of evaluating compliance.

3.87 **tile:** A rectangular array of points on the reference grid, registered with and offset from the reference grid origin and defined by a width and height. The tiles which overlap are used to define tile-components.

3.88 **tile-component:** All the samples of a given component in a tile.

3.89 **tile index:** The index of the current tile ranging from zero to the number of tiles minus one.

3.90 **tile-part:** A portion of the codestream with compressed image data for some, or all, of a tile. The tile-part includes at least one, and up to all, of the packets that make up the coded tile.

3.91 **tile-part header:** A group of markers and marker segments at the beginning of each tile-part in the codestream that describe the tile-part coding parameters.

3.92 **transformation:** A mathematical mapping from one signal space to another.

3.93 **transform coefficient:** A value that is the result of a transformation.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply. The abbreviations defined in ITU-T T.800 | IS 15444-1 section 4.1 also apply to this Recommendation | International Standard.

Editor's note: Need to remove abbreviations not used in this Recommendation | International Standard**.**

ATS: Abstract Test Suite

CCITT: International Telegraph and Telephone Consultative Committee, now ITU-T

ETS: Executable Test Suite

ICC: International Colour Consortium

ICT: Irreversible Colour transformation

ICS: Implementation Conformance Statement

IEC: International Electrotechnical Commission

ISO: International Organization for Standardization

ITTF: Information Technology Task Force

ITU: International Telecommunication Union

ITU-T: International Telecommunication Union – Telecommunication Standardization Sector (formerly the CCITT)

IUT: Implementation Under Test

JPEG: Joint Photographic Experts Group - The joint ISO/ITU committee responsible for developing standards for continuous-tone still picture coding. It also refers to the standards produced by this committee: ITU-T T.81 | ISO/IEC 10918-1, ITU-T T.83 | ISO/IEC 10918-2, ITU-T T.84 | ISO/IEC 10918-3 and T.87 | ISO/IEC 14495.

1D-DWT: One-dimensional Discrete Wavelet Transformation

RCT: Reversible Colour Transformation

ROI: Region Of Interest

RMS: Root Mean Square

SNR: Signal to Noise Ratio

TCS: Test Codestream

5 thatSymbols

For the purposes of this Recommendation | International Standard, the following symbols apply. The symbols defined in ITU-T T.800 | IS 15444-1 section 4.2 also apply to this Recommendation | International Standard.

0x----: Denotes a hexadecimal number.

e_b: Exponent of the quantization value for a subband defined in QCD and QCC.

m_b: Mantissa of the quantization value for a subband defined in QCD and QCC.

M_b: Maximum number of bit-planes coded in a given code-block.

N_L: Number of decomposition levels as defined in COD and COC.

R_I: Dynamic range of a component sample as defined in SIZ.

COC: Coding style component marker

COD: Coding style default marker

COM: Comment marker

CRG: Component registration marker

EPH: End of packet header marker

EOC: End of codestream marker

PLM: Packet length, main header marker

PLT: Packet length, tile-part header marker

POC: Progression order change marker

PPM: Packed packet headers, main header marker

PPT: Packed packet headers, tile-part header marker

QCC: Quantization component marker

QCD: Quantization default marker

RGN: Region of interest marker

SIZ: Image and tile size marker

SOC: Start of codestream marker

SOP: Start of packet marker

SOD: Start of data marker

SOT: Start of tile-part marker

TLM: Tile-part lengths marker

6 General description

Perhaps the most distinctive feature of JPEG 2000 is its emphasis on and support for scalability. An existing codestream may be accessed at a reduced resolution, a reduced quality (higher compression), a reduced number of components and even over a reduced spatial region. Moreover, the Recommendation | International Standard supports a rich family of information progression sequences and the information may be reordered without introducing additional distortion, thereby enabling a single compressed codestream to serve the needs of a diverse range of applications.

JPEG 2000 encoders may employ only a small fraction of the tools and features supported by the Recommendation | International Standard. Likewise for some applications decoders will not support all the tools and features supported by the Recommendation | International Standard. It is impossible to provide test cases for all possible combinations of tools that an encoder or decoder may choose to implement. This Recommendation | International Standard provides “abstract” test procedures for these encoders and decoders. For these tests a developer may designate the implemented tools and determine a set of test cases that test only those tools. For the greatest level of interoperability, there are explicit decoder test procedures. These tests are run for a particular profile (defined in JPEG 2000 Part I Amendment 1), and a particular compliance class defined here-in. Passing the explicit tests allows a decoder to be labeled “Profile-x Cclass-y Compliant.”

Editor’s note: There is a proposal to collapse “Profile-x Cclass-y” into just one label, e.g. “Compliant A, B, or C” this would change a lot of the text, and reduce the number of defined complaince points. NB positions on this are of interest.

Even with the explicit decoder tests, it is expected that decompressors will often not decode all of the information that was originally incorporated into the codestream by the compressor. This is the only way to truly exploit the scalability of the JPEG 2000 Recommendation | International Standard. It is desirable to allow decompressors to ignore information that is not of interest to their target application. While this flexibility is one of the strengths of JPEG 2000, it also renders inappropriate some of the conventional compliance testing methodologies that have been applied to non-scalable or less scalable compression standards.

At one extreme, decompressor implementations might be allowed to decode any portion of the codestream that is of interest to them. At the other extreme, they might be required to correctly decode the entire codestream. The first of these extreme approaches offers content providers and consumers no guarantee concerning the quality of the resulting imagery. The other approach is also inappropriate because it offers the implementor no guarantee concerning the resources that may be required and in many cases the codestream may contain information that is of no interest to the application.

This document describes compliance for JPEG 2000 decoders in terms of a system of guarantees. These guarantees serve to encourage encoders not to produce codes-streams that will be exceedingly difficult or impossible for a decoder to process, to encourage decoders to provide quality images from any reasonable codestream, and to encourage use of the flexibility and scalability of JPEG 2000 codestreams.

Essentially, if a JPEG 2000 encoder produces a codestream with certain properties then a decoder of a certain Cclass will be capable of decoding an image with some defined level of quality. Of course, the compliance class of a decoder is based solely on passing certain tests. The tests in this document are designed to require a compliant decoder to be capable of decoding all codestream with a set of defined properties.

6.1 Conformance requirements

Conformance requirements may be classified as

a) **mandatory requirements:** these shall be observed in all cases;

b) **conditional requirement:** these shall be observed if the conditions set out in the specification apply;

c) **optional requirements:** these may be selected to suit the implementation, provided that any requirements applicable to the optional are observed.

Editor’s note: MJG I hate having “optional requirements” it’s an oxymoron.

6.2 Implementation conformance statement

To evaluate the conformance of a particular implementation, there shall be a statement of the options that have been implemented. This will allow the implementation to be tested for conformance against the relevant requirements - and against those requirements only. Such a statement is called an Implementation Conformance Statement (ICS). This statement shall contain only options within the framework of requirements specified in the ISO/IEC 15444-1 Standard. Examples of these for decoders are in Annex F and encoders in Annex G.

6.3 Encoders and Codestreams

The codestream may be understood as providing certain guarantees to the decoder. In particular, JPEG 2000 Part I Annex A describes a codestream syntax and parameter limitations pertaining to that syntax that every decoder can expect. Two restricted profiles (profile 0 and profile 1) are also described in ITU-T T.800 | IS 15444-1 Amendment 1, which provide further guarantees concerning the parameter ranges and information placement that can be expected by a decoder. Since codestream limitations may also adversely affect scalability and inter-operability, the smallest possible number of limitations is imposed at this level.

Compliant implementations of the decoder are not required to decode the entire codestreams but are required to guarantee performance up to one of a collection of “Compliance Classes” or “Cclass.” These guarantees are directly connected with resources required by an implementation. They may be interpreted as a contract by the implementation to recover, decode and transform a well-defined minimal subset of the information contained in any codestream. This contract is described in a manner that scales with the compliance class. The contract may be exploited by content providers to optimize recovered image quality over a family of decoders according to their known Cclasses.

Encoders may also be required to conform to certain guarantees in particular application areas of interest. As an example, a medical image application may require the encoder to guarantee lossless performance up to a given image size.

An encoder’s mandatory requirements are defined by the ICS for the encoder or the codestream and are limited by the definition of the profile and Cclass.

6.4 Decoders

Decoder guarantees are made in connection only with a specific codestream profile. These guarantees are expressed in terms of several parameters including image dimensions, H (height) and W (width), and a number of components, C, for the compliance class. The parameters are not dependent on the codestream that is actually being decompressed. Decoder guarantees also do not impose restrictions on the codestreams that the decompressor must be able to process in a compliant manner. They refer to the claimed capabilities of the decompressor. Annex A defines the parameters and Annex B defines the classes for which compliance claims may be made and tested.

Decoders are expected to be able to recover all of the code-block contributions from the codestream that are relevant to their claimed Cclass, up to some limit that bounds the memory required for storing packet header information.

Decoders are expected to decode all of the compressed bits that are available for code-blocks belonging to their claimed Cclass in accordance with ITU-T T.800 | IS 15444-1 Annex D. These code-blocks represent various resolutions and spatial regions and components defined for each Cclass. Decoders also need only decode those compressed bits that correspond to the most significant M magnitude bit-planes for the relevant code-block.

Decoders are expected to implement both the reversible 5/3 transform and the irreversible 9/7 transform to a prescribed level of accuracy. They are also required to implement dequantization and colour transformation to a prescribed level of accuracy.

Decoder IUT’s mandatory requirements are defined by the profile and Cclass to which the IUT desires to achieve compliance. Extensions above these mandatory requirements are defined by the ICS of the IUT.

6.5 Compliance classes

Three Compliance Classes (CClass) are defined in Annex A. These Cclasses are defined to allow for an implementor to meet a specific market need without being encumbered with the requirements from other market requirements. The main difference between the classes are focused on computation complexity and memory requirements, since these parameters are the main differences between applications and markets.

6.6 Abstract test suites

The Abstract Test Suites (ATS) define the general tests for parts of the JPEG 2000 Part 1 standard. Each of the ATSs include the following parts and are defined in Annex B.

d) Test purpose: What is the requirement

e) Test Method: The procedures to be followed to for the given ATS.

f) Reference: The section of the ISO document is being tested by the given ATS.

g) Test Type: Mandatory, conditional, or optional requirements.

h) Encoder ICS: The parameters that are defined in the ICS that are tested by the given ATS.

i) Decoder ETS files: The test codestreams that the exercise the ATS.

6.7 Encoder compliance testing procedure

The procedures for testing encoders are defined in Annex D and are complimented by the information that is gathered from a completed ICS (Annex G). These procedures are informative since this is a code-stream and decoder compliance standard.

6.8 Decoder compliance testing procedure

The procedures for testing decoders are defined in Annex C and use the ETS are defined in Annex B. These procedures and ETS will allow an IUT to evaluate compliance to each profile and Cclass.

Annex A

Compliance Classes

(This Annex forms a informative part of this Recommendation | International Standard.)

This Annex describes the compliance classes for JPEG 2000 Part I. The classes and parameters are described to provide assistance in designing a complaint decoder and to define the codestreams of Annex B. Actual compliance is determined by the codestreams and testing procedures of Annex C.

A.1 Compliance Class Parameter Definitions

Because of resource limitations, implementations of JPEG 2000 will typically not be able to decode all possible codestream. This section defines various parameters which a specific implementation might be limited.

A.1.1 Profile: codestream guarantees

Profiles provide limits on the codestream syntax parameters. Two Profiles are defined in ITU-T T.800 | IS 15444-1 AMD-1, labeled Profile 0 and Profile 1. There is an implied profile that has no restrictions above the restrictions on implementations of ITU-T T.800 | IS 15444-1. It is difficult to adequitly test conformance to unrestricted codestream. All defined tests will be with regard to a specific profile and only use codestreams conforming to that profile.

A.1.2 H, W, C: Image size guarantees

Editor's note: Amendment 1 requires a 128x128 or smaller LL subband for tiled images but not for tiled images. Thus we require decoding of the reduced resolution version for single tile images, but only the upper left tile for tiled images. This limits the ability to test things by using different options in different tiles. Any good suggestions on this are welcome.

Decoders are expected to decode all of the compressed bits that are available for all code-blocks belonging to their claimed resolution, HxW, and number of image components, C, in accordance with Annex D of ITU-T T.800 | IS 15444-1, unless these code-blocks may be skipped because of other limitations. For images with a single tile, profile 0 and profile 1 require there to be a resolution level with a size smaller than 128x128. Compliance is based on the ability to decode single tile images at the largest resolution smaller than or equal to the decoder resolution HxW.

For tiled images, ideally a decoder is able to decode code-blocks with an application-defined offset. However, for testing purposes only code-blocks at the upper left portion of the image will be decoded. Code-blocks containing samples corresponding to reference grid points within the WxH region for components i<C shall be decoded,

$$x - XO_{siz} \le W$$

$$y - YO_{siz} \le H$$

A decoder claiming compliance at some Cclass with dimensions, WxH and number of components C, must also be capable of decompressing any image with width less than or equal to W, height less than or equal to H, and number of components less than or equal to C.

A.1.3 Ncb: Parsing guarantees

Decoders need not decode compressed bits which are not recovered from the codestream as a result of the "parser quit" condition being reached.

At any given point, x, in the codestream, the quantity $N_{cb}(x)$ is defined as the total number of code-blocks for which a non-empty packet has been encountered up to that point. A precinct's first non-empty packet has a first header byte that is larger than 0x8F. An upper bound for the parser state memory required to reach point x in the codestream, may be expressed as a multiple of $N_{cb}(x)$, where the multiplicative factor depends upon the particular decoder implementation.

The existence of PLM or PLT markers in the codestream reduces the burden for parsing. If these markers are present in a codestream, Ncb(x) shall only include packets that must be decoded in order to be complaint. The point at which decoders are permitted to "quit" (i.e., ignore the rest of the codestream) is defined in terms of $N_{cb}(x)$ as:

$N_{cb}(x) > N_{cb}quit$

A.1.4 Lbody: Coded data buffering guarantees

The parser state memory described above is required to parse packets regardless of whether their code-blocks are relevant to the dimensions and number of components for which compliance is being claimed. For those code-blocks that are relevant, the implementation is required to store the recovered packet body bytes. These are the code bytes that are processed by the block decoder (Annexes C and D, ITU-T T.800 | IS 15444-1).

At any given point, x, in the codestream, the quantity $L_{body}(x)$ is defined as the total number of packet body bytes that have been encountered so far in packets whose precincts are relevant to the dimensions and components for which compliance is being claimed. Although some implementations may be able to decode some of these packet body bytes incrementally, L_{body} represents an upper bound on the number of packet body bytes that must be stored by the decoder prior to decoding. If number of packet body parts exceeds the L_{body} then the IUT is allowed to quit with expectations of partial decoding of the bit-stream.

A.1.5 M: Decoded Backplane guarantees

The block decoder shall decode all of the packet body bytes recovered by the parser in accordance with the requirements described above. This obligation is limited to the most significant M bit-planes of each code-block. Specifically, the block decoder must correctly decode the first $3(M-P_b)-2$ coding passes, if available, of any relevant code-block, b, where P_b is the number of missing most significant bit-planes signalled in the relevant packet header, as described in Annex B of ISOITU-T T.800 | IS 15444-1. The decoder is free to decode any number of additional coding passes for any code-block.

A.1.6 P: 9-7 Precision guarantees

The irreversible processing path involves dequantization, the irreversible 9/7 inverse DWT, and potentially the irreversible inverse color transform (ICT). The precision is defined in the wavelet domain with respect to the nominal range of a coefficient. Thus, for each coefficient in each subband after each level of the wavelet transform, the resulting value should be close to the coefficient from a IEEE double precision floating point representation:

abs(xref - xtest) < Px for all coefficients in all subbands

To facilitate end-to-end testing for compliance, dequantization may be performed using mid-point rounding. That is, the value of r in equation G.6 of ITU-T T.800 | IS 15444-1 will be r=1/2 in the computation of the reference images. Implementations under test may provide the option of using different values for the reconstruction parameter, r; however, if the value r=1/2 is supported and employed for compliance testing this will increase the ease of passing.

The precision values for the wavelet transform are chosen to allow very high quality imagery at various bit depths, e.g. 8, 12, and 16 bits per sample. However, for Cclass 0, the accuracy of the 9-7 filter required is set such that it is possible to be complaint by decoding and inverse quantizing and performing the 5-3 inverse wavelet transform. This allows lower cost decoders to be used for the lowest compliance class only. For higher compliance classes using the 5-3 filter in place of the 9-7 filter will not be sufficient to pass the compliance tests.

A.1.7 B: 5-3 Precision guarantees

The decoder is expected to implement the reversible 5/3 inverse DWT exactly, for bit-depths of B bits/sample or less, as specified in the SIZ marker segment (see Annex A of ITU-T T.800 | IS 15444-1). If the codestream employs the color transform (RCT) and the implementation claims compliance at 3 or more components, it must be able to perform both the reversible 5/3 and the inverse RCT exactly for bit-depths of B bits/sample or less.

This document does not specify the manner in which reconstructed image sample values are to be represented by the implementation; however, that representation must be sufficient to hold all recovered bits for image bit-depths of B bits/sample or less.

A.1.8 T_L: Transform Levels guarantees

A decoder is expected to decode a number of levels of wavelet transform. Thus, for an image where it would normally be necessary to decoder an entire 640x480 image, if 16 levels of wavelet transform have been applied, the decoded image from a complaint decode may include only the top T_L levels. This relieves compliant decoders from having to decode inefficient codestreams where there are several completely empty transform levels.

A.1.9 L: Layer guarantees

A decoder is expected to decode a number of layers of the codestream. Thus, for an image where it would normally be necessary to decode an entire image, if 64,000 layers have been used for the image data, the decoded image from a compliant decoder may include only the top L levels. This relieves compliant decoders from the burden of decoding inefficient codestreams with excessive layers.

A.1.10 Progressions

A decoder is expected to decode all possible basic progressions as specified in the COD tags. If a POC marker segment is used a complaince class 0 decoder shall decode packets associated with the first progression specified in the first POC marker segment for that tile, additional packets in the tile may be skipped. For all other compliance classes packets may be skipped only due to other limitations (Ncb, or Lbody) and there is no explicit limitation on the number of progresssion order changes which may occur.

A.2 Compliance Class Definitions

Table A-1 — Definitions of Compliance Classes (Cclass)

Parameter	Cclass 0	Cclass 1	Cclass 2
Profile	0	0	1
Size (WxH)	128x128	1024x1024 or 2048x2048 or 1280x1280	8192x8192 or 16384x16384 or 10240x10240 or 20480x20480?
Components	1	4	4?
Ncb	(HW/1024 + 32)C = 48	(HW/256 + 128)C	(HW/256 + 128)C
Lbody	1/2(HWC) = 8192 bytes	3/2HWC	2HWC
M	11	15	30?
P	low enough to allow 5-3 decoding of 9-7 data	2^-11	2^-15
B	8	12	24
T_L	3	6	12
L	15	255	65535
Progressions	All “basic” progressions in COD, Only need decode first progresion per tile	Decode all progressions? up to 30 progressions?	Decode all progressions
Tile Parts	Decode only first tile part per tile	Decode all tile parts up to NcbQuit or Lbody	Decode all tile parts up to NcbQuit or Lbody
Precincts	Decode 1st precinct per subband	Decode all precincts up to Ncb or Lbody limits	Decode all precincts up to Ncb or Lbody limits

Editor’s note: Please comment on Cclass 1 & 2 image size and number of components.

Editor’s note: Lbody for Class 0 was changed because for 128x128 image and 1 component required a 2K buffer and it was hard to test with such a small value, plus a slightly larger buffer seems reasonable. Limitations on progressions and tile parts and precincts were add because they seem to be implied by amendment 1, but exact definition is debatible.

The minimum compliance point, Cclass 0, guarantees sufficient resources for application providers to ensure truly lossless performance to a bit-depth of at least 8 bits per sample. This does not mean that lossless performance will be achieved, even if the codestream contains a lossless representation of the image. It may not be achieved if the codestream contains a large amount of irrelevant information (e.g., extra image components or resolutions that are not targeted by the particular decompressor implementation under consideration), so that the parser “quit” condition occurs before all relevant information has been recovered.

Again, lossless decompression may not be achieved, even if the codestream contains a lossless representation of 8-bit imagery, if the compressor employed an unnecessarily large number of guard bits or unnecessarily large ranging parameters, ε_b, for some subbands, or if ROI information (Annex H, ITU-T T.800 | IS 15444-1) was included in the codestream. The compressor is at liberty to make such choices and their potential impact on decoders at any compliance class are defined.

Annex B

Compliance Tests

(This Annex forms a normative and integral part of this Recommendation | International Standard.)

This Annex specifies the abstract test suites, executable test suites, and the location to find the testing data for the ETS that will be used in the compliance test procedures in Annex C.

B.1 Abstract Test Suite (Informative)

The Abstract Test Suites (ATS) define the general tests for parts of the JPEG 2000 Part 1 Standard.

B.1.1 Compressed image data order

a) Test purpose: Test the ability of an implementation to encode and decode codestreams with optional markers, marker values, and markers in different locations in the codestream.

b) Test Method: Using lossless codestreams, encode, or decode several different codestreams with variations of markers, marker values, and marker locations. This includes the following markers and how to vary their parameters. These parameters are limited to the Profile and Cclass that the implementation is being test for compliance.

— Location of markers in codestream - optional markers in different positions of the codestream.

— Priority of markers - markers in the codestream that override previous markers.

— The proper use of the pointer markers- place markers that identify quit conditions

— Image offsets (XOsiz, YOsiz) - several values including odd and even values

— Tile dimensions - several size tiles including odd, even, very small tiles

— Tile Offsets (XTOsiz, YTOsiz) - several values including odd and even values

— Component Subsampling (XRsiz, YRsiz) - several values including odd and even

— Code-blocks dimensions - all values

— Packet headers - include SOP and EPH, 5 progression orders, progression order changes, number of layers, different locations of pack headers (main header, tilepart header, codestream)

— Precincts - having several values of precinct sizes (including different for each sub-band and one that are smaller than the code-block sizes.

— Tileparts - placing the tileparts in different locations in the codestream

c) Reference: ITU-T Recommendation T.800 | ISO/IEC 15444-1 Annex A: Codestream syntax and Annex B: Image and compressed image data ordering.

d) Test Type: Mandatory for the defined ICS and profile achieved by the implementation.

e) Encoder ICS: Defines the optional markers, location, and level of use of each optional marker.

f) Decoder ETS files: As defined by profiles.

Editor's note: Do we want to break out the parameters that are the main drivers of Cclass (image size, tile size, bit depth, components,...)

Editor's note: Can we actual define what in the ICS and the ETS that points to this abstract test.

B.1.2 Arithmetic entropy encoding

a) Test purpose: Evaluate the accuracy of the implementation of the arithmetic entropy encoder within the JPEG 2000 ITU.

b) Test Method: Losslessly compressed images using all different combinations of above options. Could be done with single image by changing the options in each tile. Encode, or decode several different codestreams with different arithmetic entropy coding parameters as shown below:

— AC bypass

— Context reset on coding pass boundaries

— Termination on coding pass

— Vertical striped causal

— Predictable termination

— Segment

— Segmentation symbols

c) Reference: ITU-T Recommendation T.800 | ISO/IEC 15444-1 Annex C: Arithmetic entropy coding.

d) Test Type: Mandatory for the defined ICS and profile achieved by the implementation.

e) Encoder ICS:

f) Decoder ETS files:

B.1.3 Coefficient bit modeling

a) Test purpose: To test the accuracy of the coefficient bit modeling of the entropy encoding of the JPEG 2000 IUT.

b) Test Method: Encode, or decode several different codestreams with all possible neighboring location contexts.

c) Reference: ITU-T Recommendation T.800 | ISO/IEC 15444-1 Annex D Coefficient bit modeling

d) Test Type: Mandatory for the defined ICS and profile achieved by the implementation.

e) Encoder ICS:

f) Decoder ETS files:

B.1.4 Quantization

a) Test purpose: To test the accuracy of the quantization implementation of the JPEG 2000 ITU.

b) Test Method: Encode, or decode several different codestreams with zero levels of decomposition in the wavelet transform so that the quantization is the only parameter being tested. The accuracy should be tested with RMS Error and max error.

— Exponent and Mantissa

— Guardbits

— Dequantization offset value

— Derived and Explicit quantization

c) Reference: ITU-T Recommendation T.800 | ISO/IEC 15444-1 Annex F: Discrete wavelet transformation of tile-components.

d) Test Type: Mandatory for the defined ICS and profile achieved by the implementation.

e) Encoder ICS:

f) Decoder ETS files:

B.1.5 Discrete wavelet transform

a) Test purpose: To test the accuracy of the implementation of the two discrete wavelet transform within different tile component applications.

b) Test Method: Encode, or decode several different codestreams with the two filters varying the tile size, number of levels and test the accuracy of each of the filters. The accuracy of the 9-7 filter shall be defined by RMS error and max error while the 5-3 filter shall have no difference.

— precision of 9-7

— irreversibility of 5-3

— number of levels

— different offset conditions and tile sizes, very small subbands down to 1x1, empty subbands

— Saturation conditions?

c) Reference: ITU-T Recommendation T.800 | ISO/IEC 15444-1 Annex F: Discrete wavelet transformation of tile-components.

d) Test Type: Mandatory for the defined ICS and profile achieved by the implementation.

e) Encoder ICS:

f) Decoder ETS files:

B.1.6 DC level shift and multiple component transform

a) Test purpose: To test the ability of the implementation to achieve the DC level shift and the accuracy of the two multiple component transform.

b) Test Method: Encode, or decode several different codestreams with Test using codestreams with different combinations of all the above possibilities.

— Component depth

— Component samples that are signed and unsigned

— Precision of ICT

c) Reference: ITU-T Recommendation T.800 | ISO/IEC 15444-1 ANnex G: DC level shifting and multiple component transformations.

d) Test Type: Mandatory for the defined ICS and profile achieved by the implementation.

e) Encoder ICS:

f) Decoder ETS files:

B.1.7 Region of interest

a) Test purpose: To test the accuracy of the implementation of the ROI within the JPEG 2000 ITU.

b) Test Method: Encode, or decode several different codestreams with different size, number, shift value, different in each component and test for accuracy of reconstruction.

— Different ROI in each component

— RGN marker in main tilepart header, test for proper treatment of priorities

— Shift value?

— Partial decoding of an image with ROI

c) Reference: ITU-T Recommendation T.800 | ISO/IEC 15444-1 Annex

d) Test Type: Mandatory for the defined ICS and profile achieved by the implementation.

e) Encoder ICS:

f) Decoder ETS files:

B.1.8 codestreamFile format

a) Test purpose: The ability of the implementation to accurately represent the JPEG 2000 compressed data within the JP2 file format.

b) Test Method: Encode, or decode several different codestreams with different parameters of the file format.

c) Reference: ITU-T Recommendation T.800 | ISO/IEC 15444-1 Annex I JP2 file format syntax

d) Test Type: Optional.

e) Encoder ICS:

B.2 Executable test suites (ETS)

The executable test suites are the embodiment of the ATS. Commonly several ATS are embodied in one ETS. The following ETS are defined for each Profile and Cclass.

Editor's note: See the excel spreadsheet for a greatly expanded version of the codestream contents, this spreadsheet might be included here after it is finalized, although it has a much higher level of detail. This should be complete by the end of July ISO meeting. Mike G. action.

Different items are tested for profile 0 and profile 1, a list of the items tested by the codestreams organized by category appears in Table B-1.

Table B-1 — Items Tested by Each Codestream

Feature	Tested in Profile 0	Additional Items Tested in Profile 1
Tiles	1, many	1x1, NxN with N odd, 1024x1024
Tile Parts	1 per tile, more than one per tile (interleaved & not), empty tile part, number of tile parts not indicated	out of order tiles, interleaved & not interleaved tile parts
Image content	Natural, Synthetic (Random, graphics, special values)	
Image Offsets	0	odd values, very large
Tile Offsets	0	odd values, almost tile size
Components & Transform	1, 3 without, 3 with ICT (9x7), 3 with RCT (5x3), more than 255*	
Subsampling	1,2, 4 horizontal and vertical	3
Component depth	1,4,8, 12*, (signed and unsigned)	16
Progression	5 basic, POC	
Layers	1,8, 16, many*	100s?
MQ-coder	predictive termination, segmentation symbols, termination every coding pass, none, combinations	selective bypass, reset context probabilities, vertically causal
Guard bits	0, 2, 3	
MQ-coder	All probabilities & contexts used	
RGN	none, main, tile, both, different Srgn	
Codeblock Size	32x32, 64x64	4x4, 64x32, 4x64
Precincts	Max, Explicit (big enough for 1 per subband)	Small enough for many per subband, empty precincts, smaller than default codeblock, different sizes in different subbands

Table B-1 — Items Tested by Each Codestream

Feature	Tested in Profile 0	Additional Items Tested in Profile 1
COC overrides COD	no COC, override codeblock style, override codeblock size, # of levels, transform, precinct size	
Tile overrides main header	----	tile COC overrides main COD, tile COC over main COC, tile QCC over main QCC, tile COD over main COC, tile QCD over main COC
Informational Markers	None, COM in main, COM max length, PLT, TLM, CRG, PLM, SOP, EPH, SOP not everywhere, COM in tile	
PPM/PPT	None	PPM, PPT
Levels of wavelet	0,3, 5*, 6*, 8*	
Wavelet	5-3 lossless, precision of 9-7, empty subbands, 1x1	
Quantization	Implicit, Scalar Implicit, Scalar Derived	

Editor's note: See the excel spreadsheet for a greatly expanded version of the codestream contents. It is possible the tags from the codestreams will be dumped into a table, but only after the codestreams are agreed upon.

B.2.1 Profile 0 Bitstream Descriptions

Table B-2 describes the items tested by each codestream:

Table B-2 — Codestream Descriptions for Profile 0

Feature	Codestream1	Codestream2	Codestream2.j2k
Tiles	1	1	1, many
Tile Parts	1	1	1 per tile, more than one per tile (inter-leaved & not), empty tile part, number of tile parts not indicated
Image content	Natural	Natural, Synthetic (Random, graphics, special values)?	Natural, Synthetic (Random, graphics, special values)
Image size	128x128		
Image Offsets	0	0	0
Tile Offsets	0	0	0
Components & Trans-form	1	1	1, 3 without, 3 with ICT (9x7), 3 with RCT (5x3)
Subsampling	1,1	2 horizontal	1,2, 4 horizontal and vertical
Component depth	8 unsigned	8 signed	1,4,8, 12, (signed and unsigned)
Progression	0	layer	5 basic, POC
Layers	1	6	1,8, 16
MQ-coder	-	predictive termination, segmentation symbols	predictive termination, segmentation symbols, reset context probabilitys, none, combinations
Guard bits	2	not 2?	0, 2, 3
MQ-coder	? probabilities & contexts used	? probs & contexts	All probabilities & contexts used
RGN	none	none	none, main, tile, both, different Srgn
Codeblock Size	32x32	64x64	32x32, 64x64
Precincts	Max	Explicit max	Max, Explicit (big enough for 1 per sub-band)
COC overrides COD	no COC	COC overrides codeblock style, over-ride codeblock size, # of levels, trans-form, precinct size	no COC, override codeblock style, over-ride codeblock size, # of levels, trans-form, precinct size
Tile overrides main header	----	----	----
Informational Markers	none	CRG, COM in tile, SOP & EPH every-where	None, COM in main, COM max length, PLT, TLM, CRG, PLM, SOP, EPH, SOP not everywhere, COM in tile
PPM/PPT	None	None	None
Levels of wavelet	3	3	0,3, 5, 6, 8
Wavelet	5-3 lossless	5-3 lossless	5-3 lossless, precision of 9-7, empty sub-bands, 1x1
Quantization	Scalar Implicit	Scalar Implicit	Implicit, Scalar Implicit, Scalar Derived

B.2.2 Profile 1 Bitstream Descriptions

Table B-3 describes the items tested by each codestream for profile 1.:

Table B-3 — Codestream Descriptions for Profile 1

Feature	Additional Items Tested in Profile 1	Additional Items Tested in Profile 1	Additional Codestreams for Profile 1
Tiles	1x1, NxN with N odd, 1024x1024	1x1, NxN with N odd, 1024x1024	1x1, NxN with N odd, 1024x1024
Tile Parts	out of order tiles, inter-leaved & not interleaved tile parts	out of order tiles, interleaved & not inter-leaved tile parts	out of order tiles, interleaved & not inter-leaved tile parts
Image content			
Image size	odd values, very large	odd values, very large	odd values, very large
Image Offsets	odd values, almost tile size	odd values, almost tile size	odd values, almost tile size
Tile Offsets			
Components & Trans-form	3	3	3
Subsampling	16	16	16
Component depth			
Progression	100s?	100s?	100s?
Layers	selective bypass, termination every coding pass, vertically causal	selective bypass, termination every cod-ing pass, vertically causal	selective bypass, termination every cod-ing pass, vertically causal
MQ-coder			
Guard bits			
MQ-coder			
RGN	4x4, 64x32, 4x1024	4x4, 64x32, 4x1024	4x4, 64x32, 4x1024
Codeblock Size	Small enough for many per subband, empty precincts, smaller than default code-block, different sizes in differ-ent subbands	Small enough for many per subband, empty precincts, smaller than default codeblock, different sizes in different subbands	Small enough for many per subband, empty precincts, smaller than default codeblock, different sizes in different subbands
Precincts			

Table B-3 — Codestream Descriptions for Profile 1

Feature	Additional Items Tested in Profile 1	Additional Items Tested in Profile 1	Additional Codestreams for Profile 1
COC overrides COD	tile COC overrides main COD, tile COC over main COC, tile QCC over main QCC, tile COD over main COC, tile QCD over main COC	tile COC overrides main COD, tile COC over main COC, tile QCC over main QCC, tile COD over main COC, tile QCD over main COC	tile COC overrides main COD, tile COC over main COC, tile QCC over main QCC, tile COD over main COC, tile QCD over main COC
Tile overrides main header			
Informational Markers	PPM, PPT	PPM, PPT	PPM, PPT
PPM/PPT			
Levels of wavelet			
Wavelet			
Quantization			
Created with	1x1, NxN with N odd, 1024x1024	1x1, NxN with N odd, 1024x1024	1x1, NxN with N odd, 1024x1024
JJ2000 decode	out of order tiles, interleaved & not interleaved tile parts	out of order tiles, interleaved & not interleaved tile parts	out of order tiles, interleaved & not interleaved tile parts
Jasper decode			
VM8.6 decode	odd values, very large	odd values, very large	odd values, very large
Other decode	odd values, almost tile size	odd values, almost tile size	odd values, almost tile size

B.2.3 Fileformat Test Descriptions

Items not currently “captured” by table

Codestreams that use byte stuffing (packet headers, MQ-coder)

Normalization test 1xN, Nx1, 1x1 for wavelet

Precinct & tile boundries - equalivalent at low resolution but different at full resolution

POC with tile parts

Ability to decode

Annex C

Decoder Compliance Testing Procedures

(This Annex forms a normative and integral part of this Recommendation | International Standard.)

C.1 General

Every compliant decoder must be able to fully decode or partially decode a codestream within the terms of a given profile and Cclass (as defined in Annex B) which the IUT is being evaluated for compliance.

C.2 Decoder compliance test procedure

To achieve the testing of multiple different aspects of the JPEG 2000 compression system, multiple test codestreams were developed (as defined in Annex B) for each profile and Cclass. The process includes testing for each of these codestreams, where failure to meet the requirements on one of these streams is a failure in compliance. It is expected that the majority of the systems will be at least Profile 1 Cclass 1 but each systems is allowed to be tested to any given Profile and Cclass.

C.2.1 Multiple test codestreams

A decoder is defined as compliant, to a given Cclass and profile, if the IUT passes all of the ETS up to and including the ETS codestreams for the given Cclass and profile. For example, a profile 1 Cclass 1 compliant system successfully decodes (within the accuracy defined in this Annex) Profile 0 Cclass 0, Profile 0 Cclass 1, Profile 1 Cclass 0, and Profile 1 Cclass 1 but it does not have to successfully decode Profile 1 Cclass 2. The process to test to a given profile and Cclass is shown in Figure C-1. Only when each of the TCS for each of the Profiles and Cclasses up to and including the IUT desired Profile and Cclass can a decoder claim compliance to the JPEG 2000 Part 1 Standard.

C.2.2 Compliance test procedure for a given codestream

Figure C-2 shows the flow of the decoder compliance test and the steps for this compliance test are listed below. A decoder is found to be compliant for a given TCS if the decoded test data matches the original test image within the required tolerance. A system is considered compliant if it can decode a given codestream to the required level of quality or better. The steps for this test are listed below.

a) For each new codestream, decode with the IUT.

b) If the decoder legally quits, skip to step d.

c) If the decoder quits without meeting the quit conditions, then the decoder fails this compliance test.

d) Compare the original image to the decoded (or partially decoded) image as defined in Tables C-1 through Tables C-5 for the Cclass desired.

e) If the difference between the original and decoded image is equal to or less than the limit (as defined in Tables C-1 through C-5) between the original image and the fully decoded or partially decoded image, the decoder is found to be compliant.

f) If the difference between the original and decoder image is greater than the limit the decoder is fails this compliance test.

g) Repeat for each test image required for the decoder under testing Cclass

l.

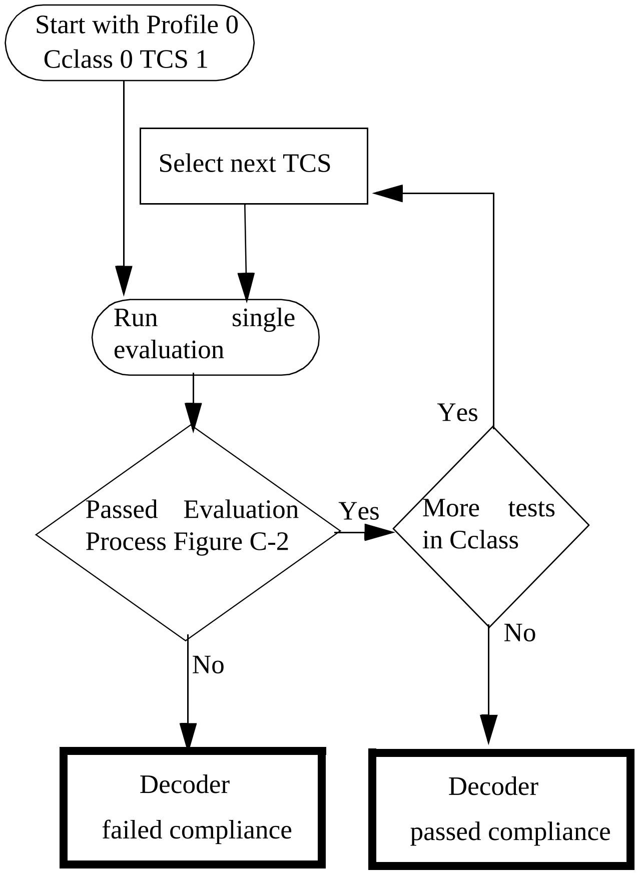

Figure C-1 — Decoder compliance test flow chart

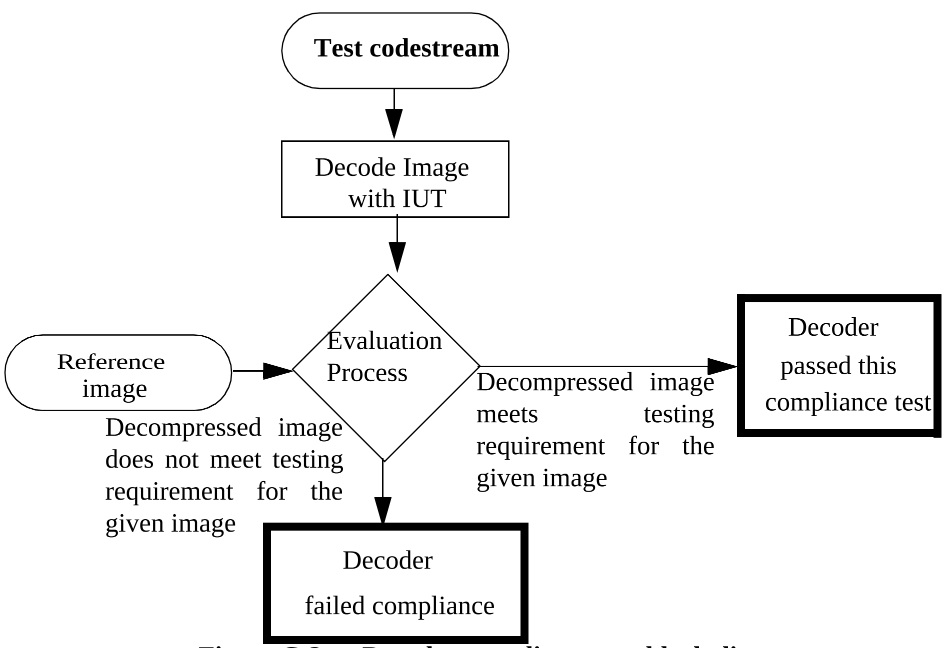

Figure C-2 — Decoder compliance test block diagram

C.3 Compliance requirements

For each of the test codestreams there is a testing requirement for each profile and Cclass. Each TCS has a Max Error and RMS error associated with it. For fully decoded lossless encoded codestreams the RMS error of zero and max error of zero. When the IUT meets the quit condition a partially decoded image is evaluated with a higher max error and RMS error, as specified for each TBS in Tables C-1 through C-5.

Editor's note: Need to define these for each test codestreams and the given profile requirements.

Editor's note: We probably need to have a MSE and Peak error per component, especially when we are testing subsampled components

Editor's note: If an implementation can decode the full image, but the compliance class only requires a 128x128 portion, shall we allow extra tools to resize the image before comparison? Or are complaint implementations required to support decoding at the Cclass point (I don't think they should be). Likewise are implementations required to decode single components to pass Cclass 0 when a 3 component image is provided? Is it okay to run multiple times to extract different components? (If this isn't allowed a complaint system could be created by saving the file, and they running N times).

Editor's note: What shall we do about clipping? (The standard doesn't specify what to do with an output of 257 for an 8 bit image, but says that clipping is often performed, do we require it?)

Table C-1 — TCS compliance requirements

codestream	Profile 0 Cclass 0	Profile 0 Cclass 1	Profile 1 Cclass 0	Profile 1 Cclass 1	Profile 1 Cclass 2
TCS 1 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 2 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 3 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 4 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 5 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 6 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 7 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 8 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 9 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0
TCS 10 Max Error RMS Error	0 0	0 0	0 0	0 0	0 0

Annex D

Encoder Compliance Test Procedure

(This Annex forms a normative and 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 are defining a procedure for testing compliance of an encoder. The encoder can produce any subset of the JPEG 2000 Part 1 but what ever it does produce must be compliant.

D.1 General

All encoding modes or capabilities are not a requirement for compliance to this Recommendation | International Standard. While these aspects are not a requirement they are a desired feature for many applications. The abstract test suites, as described in Annex B is used to support the concepts in this section. It is impossible to define test for every encoder variation. Therefore, it is left to the user to define the test that are required. This section gives guidance based on the abstract test procedures, the conformance statement, and the procedures below.

D.2 Compliance requirement and acceptance

It is not a requirement for an encoder to produce any specific codestream or every codestream; however, any codestream that is produced must be compliant. Compliance for a properly encoded codestream is dependent on the acceptance of the syntax checker and being within the requirements of the reference decoder output. This means that the same requirements for a given codestream to be fully decoded in Annex C for similar test suites. The reference decoder is supplied in Annex E.

D.3 Encoding compliance test procedure

In this Annex and all of its subclauses, the flow charts and tables are normative only in the sense that they are defining an output. Lossy encoding is not a requirement to be a compliant encoder; however, if a lossy codestream is produced, it must be compliant. Figure D-1 shows the flow for the lossy encoder compliance test and steps for this compliance tests are shown below. To facilitate this test, the encoder under compliancies testing must supply the corresponding decoded image.

a) Select test images that represents the type of imagery that the encoder is designed to compress. (The images that are provided with these compliance tests are acceptable but not required.)

b) Encode with the encoder under testing at some user-defined bit rate (lossless to lossy).

c) Check the resulting codestream and markers with the provided codestream/marker checker program.

d) If codestream or markers are found not to be compliant, the encoder has failed compliance.

e) If the codestream and markers are found to be compliant, send the codestream to the reference decoder and the user defined decoder. Each of these must fully decode the image.

f) Compare the reference reconstructed image (decoded image) to the user IUT.

g) If the difference is greater than the limit of the test requirements, the decoder has failed this compliance test.

h) An encoder is found to be compliant if the resulting test data matches the decoder reference reconstructed image for each sample within the requirements.

i) Repeat steps a) through i) for all parameters for which the encoder is designed. These parameters (tile size, cells, decomposition levels, color conversions), which are in the image characteristic column of Table J-1, should be varied to the extent in which the encoder will be used.

j) Repeat steps a) through j) for several test images, sampling the breadth of imagery types (small image size, large image size, odd image sizes, number of components, components depths, component sampling) the encoder is designed to compress.

The reference code is limited in some parameters such as the number of components and component bit depth. Also, the codestream can be embedded in other file formats. It is the responsibility of the tester to strip away any proceeding and trailing file information before sending the encoded codestream to the marker checker or reference decoder.

Editor’s note: Need to cross reference of Table J-1.

Editor's note: .REDO figure to be either block diagram or flow chart (no parallel paths in flow chart!)

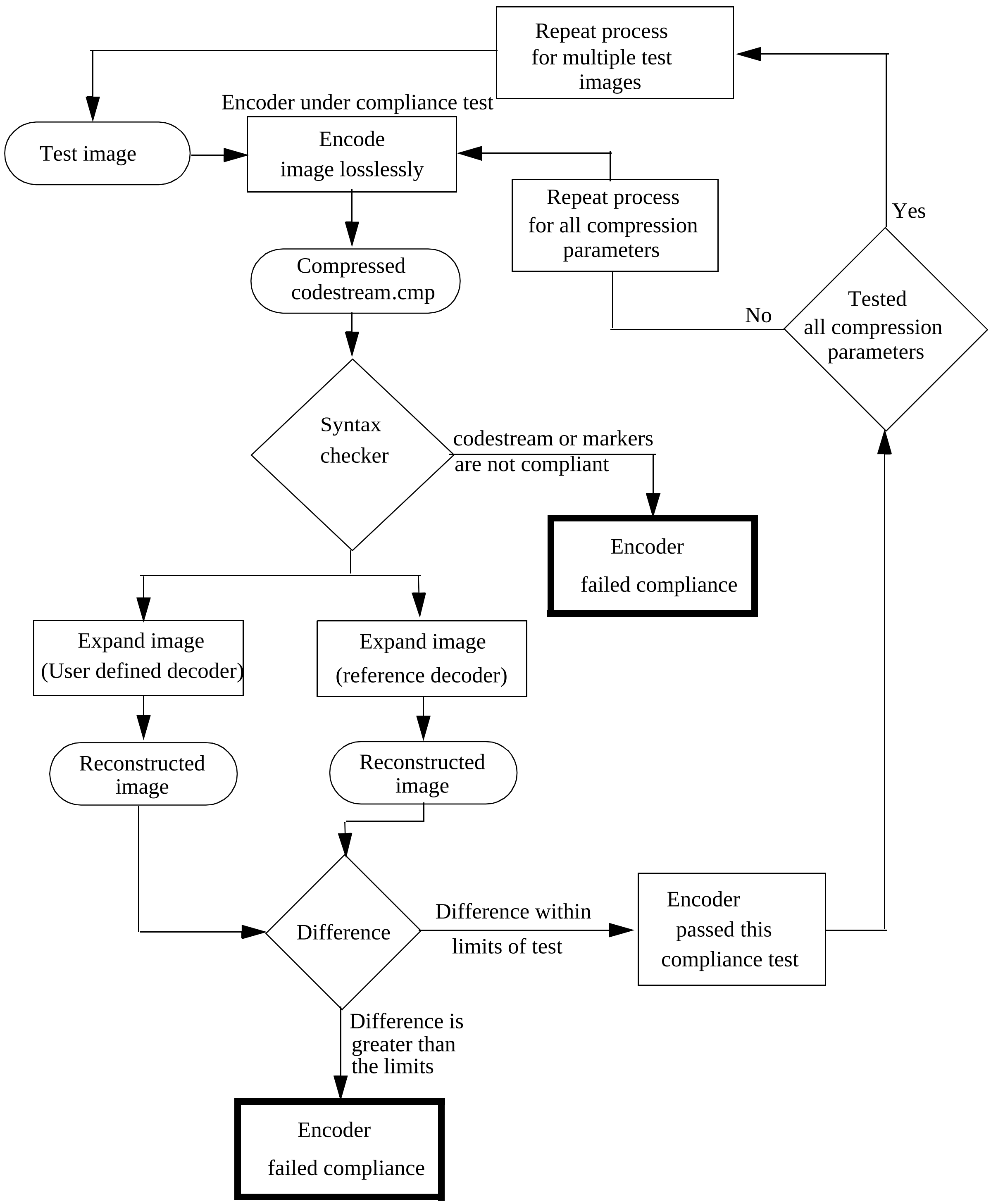

Figure D-1 — Lossy encoder compliance test block diagram

Annex E

Compliance Testing Tools

(This Annex forms an normative part of this Recommendation | International Standard.)

E.1 General

This section describes the compliance testing tools that will be used in this Recommendation | International Standard. The syntax checker and reference decoder are used for the encoder compliance testing and the quality evaluation tool is used for both the encoder and decoder compliance testing. Each of these tools are provided to the user by the ISO with no guaranties to the user. All of these tools are available on the CD with this Recommendation | International Standard or the latest version should be available on the JPEG web site (www.jpeg.org).

E.2 Syntax Checker

This is the software that is used to evaluate codestreams produced by a encoder IUT for compliance. The JPEG 2000 standard allows for flexible development of the JPEG 2000 codestream and syntax markers. The markers, code-blocks, and other parts of the codestream are allowed to be arranged into the codestream in a variety of the orders. For each of these images the syntax checker will evaluate the compliance to the rules and restrictions of JPEG 2000 Part 1 bit ordering.

E.3 Reference Decoder

The reference decoder is used for the evaluation of compliance of an IUT encoder. The reference decoder has been developed by the ISO WG 1 committee for the purpose of guidance for implanter and data providers. The reference decoder should be able to decode most encoder developed codestreams.

E.4 Quality Evaluation Tool

This tool is used to compare two images and produce two values that are used to evaluate if the IUT meets the requirements. The Max error and RMS error for each of the TCS at each profile and Cclass are defined in Tables C-1 through C-5. The QET will produce RMS error and Max error measurements between two test images. The two test images for an encoder IUT will be the reference decoded image and the original image. The two test images for a decoder will be

RMS error is defined as follows:

RMS Error =

Max Error is define as follows:

Max Error =

Editor's note: The editor does not know how to do equation editor in Frame.

Annex F

Decoder Implementation Conformance Statement

(This Annex forms an informative part of this Recommendation | International Standard.)

F.1 General

This Annex allows the decoder IUT to describe it's capabilities. The first section is the set Profiles and Cclasses. Each system will define to what level it should be test to. Once it has defined a Profile and Cclass, the IUT will be evaluated with all ETS up to and including the Profile and Cclass being test.

F.2 Decoder implementation conformance statement

The following tables are used to define limitations of an implementation. Table F-2 describes the location and number of occurrences for the given markers. Only areas that are white or gray can be filled out. Table F-3 indicate the actual values that are supported in each marker

Table F-1 — ICS for profiles and Cclass

Cclass	Profile 0	Profile 1	File Format
Cclass 0			
Cclass 1			
Cclass 2			

F.3 Extended support

While this Recommendation | International Standard defines compliance testing to two profiles and three Cclassess, it is expected that many IUT will support extra features beyond the Profile or Cclass that is being tested. Table F-2 and Table F-3 define the parameters that may be extended beyond the current defined Profile or Cclass. For example, a system may want to support more components that are required for Profile 1 Cclass 1 but do not wish to extend their system to the next Cclass. In this case, the IUT would define indicate in Table F-1 that they were Profile 1 Cclass 1 compliant and would include information in Table F-2 and Table F-3 that they are capable of more components than are required for Profile 1 Cclass 1. It is expected that special test may be required to test these capabilities since it is unknown where an implementation may have extended support. Some extended support is included in the current ETS but it is recommended that these tests are produced as needed.

Table F-2 — Extended support for decoder markers

Marker	Marker Value	Image Header	Tile Header	Codestream	Number of Occur-rences
SOC	0xFF4F	Req.			
SOT	0xFF90		Req.		
SOD	0xFF93		Req.		
EOC	0xFFD9			Req.	
SIZ	0xFF51	Req.			

Table F-2 — Extended support for decoder markers

Marker	Marker Value	Image Header	Tile Header	Codestream	Number of Occur-rences
COD	0xFF52				
COC	0xFF53				
RGN	0xFF5E				
QCD	0xFF5C	Req.			
QCC	0xFF5D				
POC	0xFF5F				
TLM	0xFF55				
PLM	0xFF57				
PLT	0xFF58				
PPM	0xFF60				
PPT	0xFF61				
SOP	0xFF91				
EPH	0xFF92				
CRG	0xFF63				
COM	0xFF64				

Editor's note: Should this table be changed to the same as the ones that are used for the ETS. There is no reason to included values or markers that are not limited in anyway in profiles or Cclass.

Table F-3 — Decoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Profile __ Cclass __ limitation	Decoders extended capability
SOC	16	0xFF4F		
SOT	16	0xFF90		
Lsot	16	10		
Isot	16	0 - 65 534		
Psot	32	"0 is unknown, 12 - (2^32 - 1)"		
TPsot	8	0 - 254		

Table F-3 — Decoder Supported Marker Values

Parameter Marker	Mark er Size	Possible Value	Profile __ Cclass __ limitation	Decoders extended capability
TNsot	8	“0 - Unknown 1-255 Number of tile parts”		
SOD	16	0xFF93		
EOC	16	0xFFD9		
SIZ	16	0xFF51		
Lsiz	16	41 - 49 190		
Rsiz	16	0000 0000 0000 0000		
Xsiz	32	0 - (2^32 -1)		
Ysiz	32	0 - (2^32 -1)		
X0siz	32	0 - (2^32 -2)		
Y0siz	32	0 - (2^32 -2)		
XTsiz	32	0 - (2^32 -1)		
YTsiz	32	0 - (2^32 -1)		
XT0siz	32	0 - (2^32 -2)		
YT0siz	32	0 - (2^32 -2)		
Csiz	16	1 - 16 384		255
Ssiz	8	x000 0000 - x0100101		
		0xxx xxxx		
		1xxx xxxx		
XRsiz	8	1 -255		
YRsiz	8	1 - 255		
COD	16	0xFF52		
Lcod	16	12 - 45		
Scod	8	"xxxx x000 - xxxx x111”		
SGcod	32	“32 bytes bro-ken into 8,16,8 “		

Table F-3 — Decoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Profile __ Cclass __ limitation	Decoders extended capability
Progression orders	8	“0000 0001 - 0000 0100"		
Number of Layers	16	1 - 65535		
Multiple Component Transform	8	“0000 0000 or 0000 0001"		
SPcod		Variable size		
Number of Levels	8	0 - 32		
Code-block Width		“xxxx 0000 - xxxx 1000"		
Code-block Height		“xxxx 0000 - xxxx 1000"		
Code-block Style		“xx00 0000 - xx11 1111"		
Transformation		“0000 0000 or 0000 0001"		
Precinct size	Variable			
COC	16	0xFF53		
Cloc	16			
Ccoc	8 or 16	“0 - 255 if Csiz is <2570 - 16 383 Csiz > 257"		
Scoc	8	“0000 0000 - 0000 0001"		
SPcod		Variable size		
Number of Levels	8	0 - 32		
Code-block Width		“xxxx 0000 - xxxx 1000"		
Code-block Height		“xxxx 0000 - xxxx 1000"		

Table F-3 — Decoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Profile __ Cclass __ limitation	Decoders extended capability
Code-block Style		"xx00 0000 - xx11 1111"		
Transformation		"0000 0000 or0000 0001"		
Precinct size	Variable			
RGN	16	0xFF5E		
Lrgn	16	5 or 6		
Crgn	"8 or 16"	"0 - 255 if Csiz is <257 0 - 16 383 Csiz > 257		
Srgn	8	0000 0000		
SPrgn	8	0 - 255		
QCD	16	0xFF5C		
Lqcd	16	4 - 197		
Sqcd	8	"0000 0000 - 1111 1111"		
SPqcd	Variable			
QCC	16			
Lqcc	16	5 - 199		
Cqcc	8 or 16	"0 - 255 if Csiz is <257 0 - 16 383 Csiz > 257"		
Sqcc	8	"0000 0000 - 1111 1111"		
SPqcc	Variable			
POC	16	0xFF5F		
Lpoc	16	9 - 65 534		
RSpoc	8	0 - 33		

Table F-3 — Decoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Profile __ Cclass __ limitation	Decoders extended capability
CSpoc	“8			
16"	“0 - 255 if Csiz is <257			
0 - 16 383 Csiz > 257"				
LYEpoc	16	0 - 65 534		
REpoc	8	RSpoc - 33		
CEpoc	“8			
16"	“CSpoc - 255 if Csiz is <257			
CSpoc - 16 383 Csiz > 257"				
Ppoc	8			
TLM	16	0xFF55		
Ltlm	16	6 - 65 535		
Zltm	8	0 - 255		
Sltm	8	“x000 xxxx -		
x111 xxxx”				
Ttlm	"0 if ST=0			
8 if ST=1				
16 if ST=2”	“tiles in order			
0 - 254				
0 - 65 534"				

Table F-3 — Decoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Profile __ Cclass __ limitation	Decoders extended capability
Ptlm	“16 if SP=0			
32 if SP=1"	"13 - 65 535			
13 - (2^32 - 1)"				
PLM	16	0xFF57		
Lplm	16	4 - 65 535		
Zplm	8	0 - 255		
Nplm	8	0 - 255		
Iplm	Vari-able			
PLT	16	0xFF58		
Lplt	16	4 - 65 535		
Zplt	8	0 - 255		
Iplt	Vari-able			
PPM	16	0xFF60		
Lppm	16	7 - 65 535		
Zppm	8	0 -255		
Nppm	32	0 - (2^32 - 1)		
Ippm	Vari-able	Packet Head-ers		
PPT	16	0xFF61		
Lppt	16	4 - 65 535		
Zppt	8	0 - 255		

Table F-3 — Decoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Profile __ Cclass __ limitation	Decoders extended capability
Ippt	Variable	Packet headers		
SOP	16	0xFF91		
Lsop	16	4		
Nsop	16	0 - 65 535		
EPH	16	0xFF92		
CRG	16	0xFF63		
Lcrg	16	6 - 65 534		
Xcrg	16	0 - 65 535		
Ycrg	16	0 - 65 535		
COM	16	0xFF64		
Lcom	16	5 - 65 535		
Rcom	16	0 or 1		
Ccom	8	0 - 255		

Annex G

Encoder Implementation Conformance Statement

(This Annex forms an informative part of this Recommendation | International Standard.)

G.1 General

It is impossible to define compliance tests for every variation of a compliant encoder, but since the only requirement for an encoder to be compliant is that has to produce compliant codestreams it can produce only one type of codestream or many types of codestreams. For example, a digital camera may only produce one type of JPEG 2000 encoded data (at different quality levels) while a photographic editing software may produce multiple versions of JPEG 2000 encoded data depending on the users requirements. This Annex allows the IUT to define its capabilities in a Implementation Conformance Statement (ICS) to be tested as defined in Annex D. This Annex was developed to support the evaluation of an encoder IUT.

G.2 Encoder Description

Table G-1 describes the markers used by an encoder. The number of occurrences states the number of times an encoder can include option tags. This table defines most of the usage of different encoder capabilities. Table G-2 defines the values that are supported for the encoder. For example, Xsiz and Ysiz as defined in the ICS may show that the encoder under test will only encoder square images that are between the size of 256-by-256 to 2048-by-2048 in powers of two. Therefor, in the compliance testing these values (256-by-256, 512-by-512, 1024-by-1024, and 2048-by-2048) should be tested in the compliance testing. Values outside these should not be tested since the encoder does not claim to do anything outside these values. If the encoder's ICS showed that the encoder could encode an image any size between 8-by-8 to 2048-by-2048 then the image size should be tested at several different sizes that are between each of these (including both ends). For example a good test set would include test images that are 8-by-8, 8-by-2037, 15-by-1025, 513-by-759, and 2048-by-2048. Note that many of the sizes are not powers of two, may have odd sizes, not square images, and pushes the limits of both the maximum and minimum.

Table G-1 — Encoder Implementation Marker Usage

Marker	Marker Value	Image Header	Tile Header	Codestream	Number of Occur-rences
SOC	0xFF4F	Req.			
SOT	0xFF90		Req.		
SOD	0xFF93		Req.		
EOC	0xFFD9			Req.	
SIZ	0xFF51	Req.			
COD	0xFF52				
COC	0xFF53				
RGN	0xFF5E				
QCD	0xFF5C	Req.			
QCC	0xFF5D				
POC	0xFF5F				

Table G-1 — Encoder Implementation Marker Usage

Marker	Marker Value	Image Header	Tile Header	Codestream	Number of Occur-rences
TLM	0xFF55				
PLM	0xFF57				
PLT	0xFF58				
PPM	0xFF60				
PPT	0xFF61				
SOP	0xFF91				
EPH	0xFF92				
CRG	0xFF63				
COM	0xFF64				

Table G-2 — Encoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Encoder Minimum	Encoder Maximum	Limitations (i.e., powers of two, only even)
SOC	16	0xFF4F			
SOT	16	0xFF90			
Lsot	16	10			
Isot	16	0 - 65 534			
Psot	32	“0 is unknown, 12 - (2^32 - 1)”			
TPsot	8	0 - 254			
TNsot	8	“0 - Unknown 1-255 Number of tile parts”			
SOD	16	0xFF93			
EOC	16	0xFFD9			
SIZ	16	0xFF51			
Lsiz	16	41 - 49 190			
Rsiz	16	0000 0000 0000 0000			

Table G-2 — Encoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Encoder Minimum	Encoder Maximum	Limitations (i.e., powers of two, only even)
Xsiz	32	0 - (2^32 -1)			
Ysiz	32	0 - (2^32 -1)			
X0siz	32	0 - (2^32 -2)			
Y0siz	32	0 - (2^32 -2)			
XTsiz	32	0 - (2^32 -1)			
YTsiz	32	0 - (2^32 -1)			
XT0siz	32	0 - (2^32 -2)			
YT0siz	32	0 - (2^32 -2)			
Csiz	16	1 - 16 384			
Ssiz	8	x000 0000 - x0100101			
		0xxx xxxx			
		1xxx xxxx			
XRsiz	8	1 -255			
YRsiz	8	1 - 255			
COD	16	0xFF52			
Lcod	16	12 - 45			
Scod	8	"xxxx x000 - xxxx x111”			
SGcod	32	“32 bytes broken into 8,16,8 “			
Progression orders	8	“0000 0001 - 0000 0100"			
Number of Layers	16	1 - 65535			
Multiple Component Transform	8	“0000 0000 or 0000 0001"			
SPcod		Variable size			

Table G-2 — Encoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Encoder Minimum	Encoder Maximum	Limitations (i.e., powers of two, only even)
Number of Levels	8	0 - 32			
Code-block Width		“xxxx 0000 - xxxx 1000"			
Code-block Height		“xxxx 0000 - xxxx 1000"			
Code-block Style		“xx00 0000 - xx11 1111"			
Transforma-tion		“0000 0000 or 0000 0001"			
Precinct size	Vari-able				
COC	16	0xFF53			
Cloc	16				
Ccoc	8 or 16	“0 - 255 if Csiz is <2570 - 16 383 Csiz > 257"			
Scoc	8	“0000 0000 - 0000 0001"			
SPcod		Variable size			
Number of Levels	8	0 - 32			
Code-block Width		“xxxx 0000 - xxxx 1000"			
Code-block Height		“xxxx 0000 - xxxx 1000"			
Code-block Style		“xx00 0000 - xx11 1111"			
Transforma-tion		“0000 0000 or0000 0001"			
Precinct size	Vari-able				
RGN	16	0xFF5E			
Lrgn	16	5 or 6			

Table G-2 — Encoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Encoder Minimum	Encoder Maximum	Limitations (i.e., powers of two, only even)
Crgn	“8 or 16"	“0 - 255 if Csiz is <257 0 - 16 383 Csiz > 257			
Srgn	8	0000 0000			
SPrgn	8	0 - 255			
QCD	16	0xFF5C			
Lqcd	16	4 - 197			
Sqcd	8	“0000 0000 - 1111 1111"			
SPqcd	Vari-able				
QCC	16				
Lqcc	16	5 - 199			
Cqcc	8 or 16	“0 - 255 if Csiz is <257 0 - 16 383 Csiz > 257"			
Sqcc	8	“0000 0000 - 1111 1111"			
SPqcc	Vari-able				
POC	16	0xFF5F			
Lpoc	16	9 - 65 534			
RSpoc	8	0 - 33			
CSpoc	“8				
16"	“0 - 255 if Csiz is <257				
0 - 16 383 Csiz > 257"					
LYEpoc	16	0 - 65 534			

Table G-2 — Encoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Encoder Minimum	Encoder Maximum	Limitations (i.e., powers of two, only even)
REpoc	8	RSpoc - 33			
CEpoc	“8				
16"	“CSpoc - 255 if Csiz is <257				
CSpoc - 16 383 Csiz > 257"					
Ppoc	8				
TLM	16	0xFF55			
Ltlm	16	6 - 65 535			
Zltm	8	0 - 255			
Sltm	8	“x000 xxxx -			
x111 xxxx”					
Ttlm	"0 if ST=0				
8 if ST=1					
16 if ST=2”	“tiles in order				
0 - 254					
0 - 65 534"					
Ptlm	“16 if SP=0				
32 if SP=1"	"13 - 65 535				
13 - (2^32 - 1)"					

Table G-2 — Encoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Encoder Minimum	Encoder Maximum	Limitations (i.e., powers of two, only even)
PLM	16	0xFF57			
Lplm	16	4 - 65 535			
Zplm	8	0 - 255			
Nplm	8	0 - 255			
Iplm	Variable				
PLT	16	0xFF58			
Lplt	16	4 - 65 535			
Zplt	8	0 - 255			
Iplt	Variable				
PPM	16	0xFF60			
Lppm	16	7 - 65 535			
Zppm	8	0 -255			
Nppm	32	0 - (2^32 - 1)			
Ippm	Variable	Packet Headers			
PPT	16	0xFF61			
Lppt	16	4 - 65 535			
Zppt	8	0 - 255			
Ippt	Variable	Packet headers			
SOP	16	0xFF91			
Lsop	16	4			
Nsop	16	0 - 65 535			

Table G-2 — Encoder Supported Marker Values

Parameter Marker	Marker Size	Possible Value	Encoder Minimum	Encoder Maximum	Limitations (i.e., powers of two, only even)
EPH	16	0xFF92			
CRG	16	0xFF63			
Lcrg	16	6 - 65 534			
Xcrg	16	0 - 65 535			
Ycrg	16	0 - 65 535			
COM	16	0xFF64			
Lcom	16	5 - 65 535			
Rcom	16	0 or 1			
Ccom	8	0 - 255			

Annex H

JP2 File Format Reader Compliance Testing Procedure

(This Annex forms a normative and integral part of this Recommendation | International Standard.)

H.1 General

Every compliant JP2 file format reader must be able to fully decode or partially decode the codestream contained within the file within the terms as defined in Annexes C and D. In addition, the reader must also be able to properly interpret the colorspace of the decoded image data as specified within the file format. Minimal compliance for the interpretation of colorspace is defined as the possible upsampling of the decoded image data such that all decoded components are at the same resolution, and the transformation of the decoded image data to the sRGB colorspace for display on a typical computer monitor. While it is well understood that many applications will not convert images directly to the sRGB colorspace, the use of sRGB as a comparison point provides a simple and accurate way to compare output of an application under test with reference output.

While this compliance test is not normative in general for all implementations of the JPEG 2000 standard (as implementation of the JP2 file format is optional for conforming decoders), it is normative for all applications that do support the JP2 file format as defined in Annex I of ISO/IEC 15444-1.

This normative compliance test includes the required test files. The compliance test is separated into two parts, first, decoding of the codestream contained within the JP2 file, and second, the interpretation of the colorspace of the decoded image data.

H.2 Compliance requirement and acceptance

A compliant file format reader must first pass the compliance tests for decoding as defined in Annex C and Annex D. It must also be able to properly interpret the colorspace of each test file. Unlike the compliance tests defined in Annex C and Annex D, this test does not differentiate between full and partial decode of the image. Any JP2 file format reader that can properly extract the codestream from a JP2 file and decode the codestream to sufficiently pass the tests in Annex C and Annex D passes the first part of the file format compliance test.

However, for the purposes of this test, all test images should be fully decoded in order minimize the number of reference images that must be tested against. It is assumed that if a file format reader passes the tests in Annex C and Annex D, then it is capable of properly decoding any codestream contained within a JP2 file.

H.3 Reading a JP2 file compliance test procedure

Figure H-1 shows the flow of the lossless decoder compliance test. A decoder is found to be compliant if the resulting test data, for the tests specified for a particular process, exactly match the original test image.

a) For File1.jp2, extract the codestream and fully decode it with the decoder under test.

b) If the decoded components are not all at the same resolution, all components should be upsampled to the same resolution. While in general applications are not required to upsample the decoded data, and the particular method of interpolation is outside of the scope of the Recommendation | International Standard, this test requires the interpolation of the data in order to bring all images for comparison to a single state.

c) Convert the decoded image data from the source colorspace (as indicated in the Color Specification boxes within the JP2 file) to the sRGB colorspace.

d) Compare the reconstructed image (converted image) to the original data.

e) If there is a difference beyond the specified tolerance, the reader has failed this compliance test. Stop testing, identify the failure point, and correct it. After the reader is corrected, repeat steps a) through f).

Editor's note: Tolerance must be defined

f) If there is no difference between the original image and the reconstructed image, the decoder is found to be compliant, for this test.

g) Repeat steps a) through d) for all streams as defined in Table (UNKNOWN) for the level of compliance that is desired.

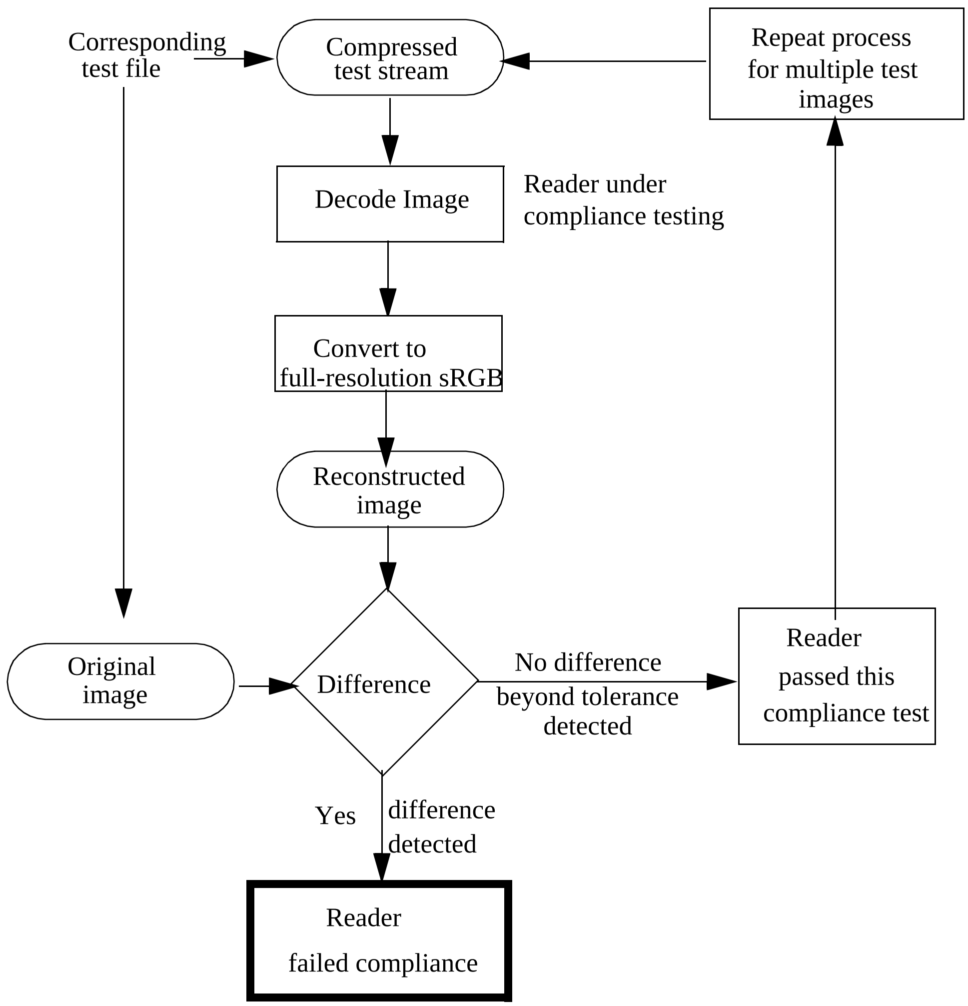

Figure H-1 — JP2 file format reader compliance test block diagram

H.4 JP2 file format test codestreams and images

Editor's note: These files should be added to Annex B

The following test images from Annex B will be used for the compliance test procedures as defined in the Annex H.2 and

Editor's note: Here are the descriptions of the files to be put into Annex B

File 1: sRGB, 3 component 8-bit, all decoded components at the same resolution

File 2: sRGB-YCC, 3 component, 8 bit, all components at the same resolution

File 3: sRGB-YCC, 3 component, 8-bit, Cb and Cr components at 1/2 the resolution of the Y component in both the horizontal and vertical directions.

File 4: greyscale, 1 component, 8-bit, all components at the same resolution

File 5: Restricted ICC, 3-component, 8-bit, all components at the same resolution

File 6: greyscale, 1 component, 12-bit, all components at the same resolution

Editor's note: The tolerance will be significantly higher for applications that only extract 8 bits of data out of this image

File 7: e-sRGB data specified using Restricted ICC, 3-component, 16-bit signed, all components at the same resolution.

File 8: ERIMM-RGB data specified using Restricted ICC, 3-component, 12-bit unsigned, all components at the same resolution

Editor's note: The tolerance will be significantly higher for applications that only extract 8 bits of data out of this image

Annex I

Subsystem Testing

(This Annex is informative only and is not an integral part of this Recommendation | International Standard.)

Editor's note: This section will probably have to be dropped unless someone is interested in doing the work rapidly. This could be very useful to developers!

I.1 General

Codestreams and intermediate decoder results are provided to assist in development of compliant decoders. For example, this section will allow a developer to separate the wavelet transform and evaluate the accuracy of their encoder or decoder implementation of the wavelet transform.

Editor's note: Should there be equivalent information for encoders?

I.2 Transform

I.2.1 9-7 wavelet filter

Provide samples of coefficients and image samples.

I.2.2 5-3 wavelet filter

Provide samples of coefficients and image samples.

I.3 Quantization

Editor's note: We need to define points of testing in the quantization. What can we supply to developers to test their implementation of the quantization for both encoder and decoder.

I.4 Context Model and Arithmetic coder

Provide samples of coefficients and context used, arithmetic coder A & C registers, and bytes.

Repeat for different terminations?