INTERNATIONAL
REGISTERED ITEM
CLASS: BIIF PROFILE

ISO/IEC
BIIF PROFILE
BPJ2K01.10

Document: ISO/IEC JTC1/SC24 N 3111

- Information Technology -
- Computer Graphics and Image Processing -
- Registered Item -
- Class: BIIF Profile -

BIIF Profile for JPEG 2000
Version 01.10
(BPJ2K01.10)

(Note: This BIIF Profile for JPEG 2000 is a profile using the JPEG 2000 proforma, intended to be used in BIIF applications.)

15 April 2009

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

Foreword

The International Standard (IS) 12087-5:1998, Basic Image Interchange Format (BIIF), provides guidance for creating profiles of BIIF. At present, two profiles of BIIF have been established: 1) the model profile of BIIF as specified in ISO/IEC 12087-5; and 2) the North Atlantic Treaty Organization (NATO) Secondary Imagery Format Version 01.01 (NSIF01.01). The NSIF01.01 Profile of BIIF allows for the compression of image data using the provisions of ISO/IEC 15444, JPEG 2000 Part 1: Image Coding System: Core Coding System.

The following is submitted as a result of the NATO Standardization Agreements (STANAG) 4545, STANAG 4609 (Edition 3), and STANAG 7023 promulgated by the Chairman, Military Agency for Standardization (MAS) under the authority vested in him by the NATO Military Committee:

BIIF Profile: BIIF Profile for JPEG 2000 Version 01.10

This BIIF Profile for JPEG 2000 is a profile developed using the JPEG 2000 proforma, ISO/IEC 15444-1, intended to be used with BIIF, LVSD (Large Volume Streaming Data, see STANAG 4609) and NATO's air reconnaissance primary imagery data (see STANAG 7023) applications.

This standard is normative.

Introduction

The BIIF Profile for JPEG 2000 Version 01.10 (BPJ2K01.10) replaces Version 01.00 (BPJ2K01.00, 30 July 2004). BPJ2K01.10 includes additional JPEG 2000 encoding profiles and a restructuring of the document that places all profiles into the Appendices. BPJ2K01.10 is within the context of the BIIF Profile class of items in accordance with the principles and procedures specified in ISO/IEC 9973:2006, "Computer graphics, image processing and environmental data representation – Procedures for registration of items," and Annex C of ISO/IEC 12087-5:1998, "Profiling BIIF."

The BPJ2K01.10 Profile of JPEG 2000 Part 1 (ISO/IEC 15444-1) was cooperatively developed between the ISO and NATO communities.

The NSIF01.01 is a BIIF profile intended to promote interoperability for imagery exchange among military Command, Control, Communications, and Intelligence (C3I) systems. The BPJ2K01.10 includes profiles for the JPEG 2000 compression of digital imagery, incorporating the compressed digital imagery into NSIF files, and exchanging these files within the user community. STANAG 4545 and (U.S.) MIL-STD-2500C are specific user community documents describing the implementation of the NSIF01.01 BIIF Profile and now, the BPJ2K01.10 Profile.

BPJ2K01.10 adds new optional JPEG 2000 preferred encodings for use within NSIF01.01. These additional preferred encodings are not normative requirements for encode-capable implementers of NSIF01.01.

The new TPJE (Tactical Preferred JPEG 2000 Encoding) was created for certain airborne applications that utilize NSIF. The TPJE is not mandated for NSIF01.01 encoding systems, but future NSIF profiles might require its use.

The new LPJE (LVSD Preferred JPEG 2000 Encoding) is normative for certain systems described within STANAG 4609 NATO Digital Motion Imagery Standard. The LPJE was developed for a new class of sensors that NATO has designated as LVSD sensors. The LVSD (Large Volume Streaming Data) designation was created for sensors that capture very large array imagery at rates of one frame per second (1fps) and faster for extended periods of time over a common area of regard.

The new SPJE (STANAG 7023 Preferred JPEG 2000 Encoding) is the mandated JPEG 2000 compression for STANAG 7023 compliant systems. This includes NATO air reconnaissance primary imagery data systems. The SPJE is not mandated for NSIF01.01 encoding systems, but future NSIF profiles might require its use.

All compliant NSIF01.01 decoders are required to decode all compliant data within the limits of ISO/IEC 15444-1 Profile-1 and Cclass 2 of ISO/IEC 15444-4 and their BIIF compliance level (Section 7). All compliant NSIF01.01 encoders must also produce compressed data that is compliant and within the limits of this profile. It is preferred that encoders further constrain themselves to comply with the preferred encoding recommendations described in the Appendices of this document.

While the Appendices define the recommended encodings, NSIF will support all encoders and encoded data that are within the limits of Section 7 . Section 8 describes the interactions between NSIF and JPEG 2000 as well as the BIIF Tagged Record Extension (TRE) recommended for use in NSIF when compressing image data per the preferred JPEG 2000 encoding recommendations.

Appendix A gives recommendations and guidance for the following procedures: compression, parsing, decompression, repackaging and other common processing tasks. Appendix A will help users and developers in using the functionality that is achieved with the recommendations of the profile. Appendix B includes processing examples that are related to the recommendations of Appendix A . The ISO JPEG body (ISO/IEC JTC 1SC29/WG1) has defined two profiles of JPEG 2000, Profile-0 and Profile-1. Appendix C includes the JPEG 2000 Profile limitations for Profile-0 and Profile-1.

The NPJE encoding recommendations (Appendix D) were selected to achieve the greatest interoperability and functionality for large images. The compression efficiency, flexibility, and functionality of JPEG 2000 will meet the NSIF user requirements currently being met with several different compression algorithms. This includes all users from the Image Analyst, who needs the very best quality and resolution (lossless compression), all the way to the bandwidth constrained user, who only needs a low resolution lower quality image at high compression.

Appendix E introduces the Exploitation Preferred JPEG 2000 Encoding (EPJE) profile. The EPJE profile was introduced to address application performance issues when roaming and zooming through large JPEG 2000 encoded imagery at reduced resolutions. This profile is related to the NPJE profile and represents a re-ordering of the compressed imagery data with respect to NPJE encoded files.

Appendix F describes the Tactical Preferred JPEG 2000 Encoding (TPJE). The TPJE is recommended for use by airborne imagery providers when performing the initial first-stage compression of collected imagery. The recommended encoding parameters of TPJE are appropriate for hardware-based compression of airborne imagery. The TPJE is a superset of NPJE and EPJE.

The LPJE described in this document (Appendix G) allows for an expanded range of encoding parameters that exceed the constraints of NPJE, EPJE and other encodings in this profile. LPJE allows the selection of encoding parameters that exceed those in ISO/IEC 1544-1 Profile-1. LPJE is a superset of these encoding profiles. The system requirements needed to implement LPJE can be significant. In light of this, NSIF systems are not required to support LPJE. Only those systems that ingest or create JPEG 2000 compressed imagery for LVSD sensors need support the LPJE. See STANAG 4609 (Edition 3) for further details.

Appendix H describes the STANAG 7023 preferred JPEG 2000 encoding. SPJE as described in Appendix H is intended to be equivalent to the JPEG 2000 profile described within STANAG 7023. At this point in time, STANAG 7023 still retains its JPEG 2000 profile. The STANAG 7023 custodian intends to modify STANAG 7023 to point to the preferred encoding given in Appendix H thus removing the redundant profile and eliminating any potential conformance issues. Until the time that this accomplished, in the event of any discrepancy between STANAG 7023 and Appendix H , STANAG 7023 takes precedence. SPJE allows a wide range of compression options that exceed those in ISO/IEC 1544-1 Profile-1. In light of this, NSIF systems are not required to support SPJE. SPJE represents a subset of LPJE and a superset of other preferred encodings found within this document.

1 Scope

This profile is intended for the compression of literal imagery (e.g., panchromatic, color, detected SAR, multispectral, thermal IR, etc.) within the NSIF profile of BIIF. It is not expected to handle non-literal imagery types (e.g., I/Q data, M/P data, VPH data, Elevation data, Location- Grid data, etc.). It is expected that the multiple component transform framework from JPEG 2000 Part 2 (ISO/IEC 15444-2) will be included when the requirements for hyperspectral imagery are established.

This profile is expected to grow with new requirements and new applications. For example, it is expected that additional capabilities from JPEG 2000 Part 2 will be included in future versions of this profile. Added functionality and new recommendations will only be added to the profile as they are required.

1.1 General

The Basic Image Interchange Format (BIIF) provides a file format suitable for the interchange, storage, and retrieval of map and imagery information. The file format consists of a file header and associated image(s), symbol(s), text and/or associated data in a manner compatible between systems of different architectures and devices of differing capabilities and design.

The BPJ2K01.10 profile defines allowed data values and ranges for JPEG 2000 header and subheader fields contained in an NSIF01.01, NITF02.1 file or LVSD codestream file. A BPJ2K01.10 file shall contain valid data (that is, data in accordance with the restrictions specified for the contents of each field in this profile definition). The BPJ2K01.10 profile meets BIIF ISO/IEC 12087-5:1998 application requirements.

1.2 Position Within the Item Register

BPJ2K01.10 is a profile for the application of ISO/IEC 15444-1, JPEG 2000 Part 1, registered under the BIIF Profile class of items in accordance with ISO/IEC 9973:2006. The BPJ2K01.10 tailors JPEG 2000 to promote a high degree of interoperability among two or more common communities of interest through the selection of a common set of functionality for digital mapping and imagery.

The item registration information is as follows:

Item Class:	BIIF Profile
Item Long Name:	BIIF Profile for JPEG 2000 Version 01.10
Item Short Name:	BPJ2K01.10
Sponsoring Authority:	The United Kingdom sponsors this Profile through their membership in the ISO committee.
Preparing Authority:	This document was prepared for the sponsoring authority by the NSIF (NATO

STANAG 4545) Custodian; National Geospatial-Intelligence Agency (NGA).

1.3 User Requirements and Scenario

NSIF01.01 is designed to promote interoperability for the exchange of digital electronic imagery among multi-national Command, Control, Communications, and Intelligence (C3I) Systems, and those systems needing to interoperate with C3I imagery systems. Adoption by NSIF of JPEG 2000 for compression of digital image data significantly enhances the ability of NSIF to meet its user requirements.

- The profile is comprehensive in the type and format of data permitted in the BIIF File.
- The profile may be implemented across a wide range of computer systems without reduction of available features.
- The profile allows extensibility to accommodate data types and functional requirements not foreseen.
- The profile provides a useful capability with limited formatting overhead.

2 References

Normative References:

The following documents contain provisions that, through reference in this text, constitute provisions of the BPJ2K01.10. Applicability is limited to only the specific instance of the reference document; other aspects of referenced documents are for information. At the time of publication the editions indicated were valid but all documents are subject to revision. Parties in agreement, based on this profile, are warned against automatically applying more recent editions of the documents listed in this section. The nature of references made by the profile to such documents is specific to a particular edition. Members of IEC and ISO maintain a register of currently valid International Standards and profiles.

Referenced Documents:	Title
ISO/IEC 12087-5: IS	Information Technology; Computer graphics and image processing; Image Processing and Interchange; Functional Specification - Part 5: Basic Image Interchange Format, 1 December 1998
NSIF Profile of BIIF	BIIF Profile: NATO Secondary Imagery Format (NSIF) Version 01.01
STANAG 4545	NATO Secondary Imagery Format (NSIF)
STANAG 4609	NATO Digital Motion Imagery Standard
STANAG 7023	NATO Primary Imagery Format
ISO/IEC 15444-1:2004	Information Technology — JPEG 2000 image coding system: Core coding system
ISO/IEC 15444-1:2004/ Amd 1:2006	Profiles for digital cinema applications.

ISO/IEC 15444-1:2004/ Cor 1:2007	Information technology — JPEG 2000 image coding system: Core coding system TECHNICAL CORRIGENDUM 1.
ISO/IEC 15444-2:2002	JPEG 2000 Image Coding System -- Part 2: Extensions.
ISO/IEC 15444-2:2004/ Cor 3:2005	JPEG 2000 Image Coding System -- Part 2: Extensions TECHNICAL CORRIGENDUM 3.
ISO/IEC 15444-2:2004/ Cor 4:2007	JPEG 2000 Image Coding System -- Part 2: Extensions TECHNICAL CORRIGENDUM 4.
ISO/IEC 15444-2:2004/ Amd 2:2006	JPEG 2000 Image Coding System -- Part 2: Extensions Extended capabilities marker segment.
ISO/IEC 15444-4:2004	JPEG 2000 Image Coding System -- Part 4: Conformance testing.
STDI-0002	The Copendium of Controlled Extensions (CE) for the National Imagery Transmission Format (NITF) Version 3.0

Non-Normative References:

The following documents are included for informational purposes only.

Related Documents:	Title
JPEG 2000: Image Compression Fundamentals, Standards, and Practice	Taubman & Marcellin, JPEG 2000: Image Compression Fundamentals, Standards, and Practice, Kluwer Academic, 2001. ISBN 0-7923-7519-X
ISO/IEC 9973:2006	Computer Graphics, Image processing and environmental data representation -- Procedures for registration of items
MIL-STD-2500C	National Imagery Transmission Format Version 2.1 for the National Imagery Transmission Format Standard
MISB RP 0301.3	MISB Profile for Aerial Surveillance and Photogrammetry Applications (ASPA), Version 1.3
AAF Specification V1. 1	Advanced Authoring Format (AAF) Object Specification v1.1
SMPTE 377M-2004	Material Exchange Format (MXF) — File Format Specification (Standard)
J2K_Guide (available from NITF Technical Board, http://www.gwg.nga.mil/ntb)	Bandwidth Compression (BWC) Guide for JPEG 2000 Visually Lossless and Numerically Lossless Compression of Imagery Data Working Draft 1.2, January 22, 2004
DCI System Specification	Digital Cinema System Specification Version 1.1, April, 2007
ISO/IEC 15444-9:2005	Information technology — JPEG 2000 image coding system: Interactivity tools, APIs and protocols.
ISO/IEC 15444-9:2005/ Amd 1:2006	APIs, metadata, and editing.
ISO/IEC 15444-9:2005/ Cor 1:2007	Information technology — JPEG 2000 image coding system: Interactivity tools, APIs and protocols. TECHNICAL CORRIGENDUM 1.

Application for copies of ISO documents may be addressed to the respective national ISO representative.

Copies of NATO Standardization Agreements may be obtained from HQ NATO, Military Agency for Standardization, 1110 Brussels, Belgium, or from the www.nato.int website, if releasable to the general public. Some Standardization Agreements may only be released to NATO member nations.

Copies of the U.S. MIL-STDs are available from Standardization document order Desk, 700 Robbins Avenue, Building 4D Philadelphia, PA 19111-5094.

3 Definitions

For the purposes of the BPJ2K01.10 profile, the definitions shown in ISO/IEC 15444-1, ISO/IEC 12087-5 BIIF, and NSIF01.01 apply.

4 Abbreviations

ABPP	Actual Bits Per Pixel
AAF	Advanced Authoring Format
ASPA	Aerial Surveillance and Photogrammetry Applications
BCS	Basic Character Set
BCS-A	Basic Character Set Alphanumeric
BCS-N	Basic Character Set Numeric
BIIF	Basic Image Interchange Format
BPJ2K	BIIF Profile for JPEG 2000
bpp	bits per pixel
bpppb	bits per pixel per band
BWC	Bandwidth Compression
C3I (C^3I)	Command, Control, Communications, and Intelligence
CIE	Commission Internationale de'lEclairage (International Commission on Illumination)
CCS	Common Coordinate System
COC	Coding style Component marker segment
COD	Coding style Default marker segment
COM	Comment Marker segment
COMRAT	Compression Rate
CONOPS	Concept of Operations
C-P-R-L	Component-Position-Resolution-Layer
CRG	Component Registration marker segment
DCT	Discrete Cosine Transform
DRA	Dynamic Range Adjustment
EOC	End of Codestream marker
EPH	End of Packet Header marker
EPJE	Exploitation Preferred JPEG 2000 Encoding
H	High pass filter
IC	Image Compression
ICC	International Color Consortium
ICT	Irreversible Component Transform
IEC	International Electrotechnical Commission
ILOC	Image Location
IREP	Image Representation
IS	International Standard
ISBN	International Standard Book Number
ISO	International Organization for Standardization
ITU	International Telecommunication Union

I/Q	In phase/Quadrature data
J2K	JPEG 2000
JP2	JPEG 2000 file format
JPC	JPEG 2000 codestream
JPEG	Joint Photograph Experts Group
JPX	JPEG 2000 eXtended file format
JTC1	Joint Technical Committee 1
KLV	Key Length Value
L	Low pass filter
LPJE	LVSD Preferred JPEG 2000 Encoding
L-R-C-P	Layer-Resolution-Component-Position
LSB	Least Significant Bit
LUT	Look-up Table
LVSD	Large Volume Streaming Data
MAS	Military Agency for Standardization
MIL-STD	Military Standard
MISB	Motion Imagery Standards Board
MJ2	Motion JPEG 2000 Format
M/P	Magnitude/Phase Data
MSB	Most Significant Bit
MS	Multispectral
MTFC	Modulation Transfer Function Compensation
MXF	Material Exchange Format
NA	Not Allowed
NATO	North Atlantic Treaty Organization
NBPC	Number of Blocks Per Column
NBPP	Number of Bits Per Pixel
NBPR	Number of Blocks Per Row
NCOLS	Number of Columns
NFS	Network File System
NGA	National Geospatial-intelligence Agency (formerly NIMA)
NIMA	National Imagery and Mapping Agency
NITF	National Imagery Transmission Format
NITFS	National Imagery Transmission Format Standard
NPJE	NSIF Preferred JPEG 2000 Encoding
NPPBH	Number of Pixels Per Block Horizontal
NPPBV	Number of Pixels Per Block Vertical
NR	Not Recommended
NROWS	Number of Rows
NSIF	NATO Secondary Imagery Format
NTB	NITFS Technical Board
P-C-R-L	Position-Component-Resolution-Layer
PLM	Packet Length Main header marker segment
PLT	Packet Length Tile-part header marker segment
POC	Progression Order Change marker segment
PPM	Packed Packet headers Main header marker segment
PPT	Packed Packet headers Tile-part header marker segment
PVTYPE	Pixel Value Type
QCC	Quantization Component marker segment
QCD	Quantization Default marker segment
RCT	Reversible Component Transform

RGB	Red, Green, Blue (IREP value)
RGN	Region of interest marker segment
ROI	Region of Interest
R-L-C-P	Resolution-Layer-Component-Position
RP	Recommended Practice
R-P-C-L	Resolution-Position-Component-Layer
RRDS	Reduced Resolution Data Sets
SC29/WG1	Sub Committee 29/Working Group 1
SIZ	Size marker segment
SOC	Start of Codestream marker
SOD	Start of Data marker
SOP	Start of Packet marker
SOT	Start of Tile-part marker segment
STANAG	Standardization Agreement
TLM	Tile-part Lengths Marker segment
TRE	Tagged Record Extension
TTC	Tonal Transfer Curve
VPH	Video Phase History data
YCbCr	Y, brightness; Cb, chrominance (blue); Chrominance (red)

5 Conformance

Conformance is a necessary step towards achieving interoperability between different imagery applications and operating systems. The JPEG 2000 standard is large and has many sophisticated encoding options. ISO/IEC 15444-1 defines profiles for encoders to help foster interoperability (see Annex A.10 and Table A-45). The JPEG 2000 profiles restrict some of the encoding parameters that used during compression. The goal of these profiles is to provide points of interoperability that applications can rely upon to help guarantee that different vendors can interchange and properly interpret JPEG 2000 codestreams.

ISO/IEC 15444-1 defines three profiles. Profile-0 is intended for mobile applications and it places severe restrictions on the allowed compression options. Profile-1 is intended for use in the vast majority of applications and in fact many of today's mobile devices can support Profile-1. The limitations on compression options in Profile-1 are less severe than in Profile-0. The most significant restriction present in either profile deals with tiling of the image during compression. Other restrictions deal with matters such as progression orders, code-block size and termination, and allowing component-specific encoding parameters.

Of course an encoder can choose to do more than what is allowed by either Profile-0 or Profile-1. In this case we will say that the encoder is using a "No Restrictions" profile (NR profile). There are very good reasons why an encoder might want to exceed the limits defined by Profile-1. For example, an encoder might want to tile an image using a tile size larger the 1,024 x 1,024 samples. In this case the encoder has exceeded the limitations of Profile-1 and is using the NR profile. It may comply with Profile-1 in all other aspects, but it must still be considered as using the NR profile. The profile level of a

JPEG 2000 compressed codestream is indicated within the SIZ marker segment by the Rsiz value (see Annex A.5.1 of ISO/IEC 15444-1).

ISO/IEC 15444-4 (JPEG 2000 conformance testing document) defines decoder "Cclasses" (see Annex A.2 and Table A-1). A Cclass represents a conformance level for a JPEG 2000 decoder. Cclasses place limits on things such as the size of an image that the decoder must be able to decode or the number of codestream code blocks that the decoder can process. Cclass limitations basically are concerned with complexity and memory issues within a decoder. They represent *minimum* performance guarantees on behalf of the decoder. There are four Cclasses defined in ISO/IEC 15444-4, Cclass 0, Cclass 1, Cclass 2 and a "No Restrictions" Cclass. ISO/IEC 15444-4 also includes test JPEG 2000 codestreams that may be used to test decoder compliance.

Any JPEG 2000 encoder must comply with ISO/IEC 15444-1 and may be categorized by the Profile(s) that it supports. Similarly any JPEG 2000 decoder must comply with ISO/IEC 15444-1 and may be tested for compliance and categorized using the test codestreams and Cclasses found in ISO/IEC 15444-4. These basic conformance requirements supersede any additional requirements defined within this document. This BIIF profile places *additional* constraints and requirements on top of those found within the Profiles and Cclasses of JPEG 2000. Therefore, any JPEG 2000 encoder and decoder that claims conformance to one or more of the preferred encodings contained within the BIIF profile shall conform not only to requirements and constraints of its preferred encoding(s), but it shall also conform to ISO/IEC 15444-1 and ISO/IEC 15444-4.

Additionally, products that conform to the BPJ2K01.10 profile will also meet the conformance requirements of ISO/IEC 12087-5.

Sections 5.1 through 5.7 give detailed conformance requirements for the various preferred encodings contained within the BIIF profile. Section 5.1 gives conformance requirements for ISO/IEC 15444-1 and ISO/IEC 15444-4 that apply to *all* preferred encodings. Section 5.2 defines conformance requirements for NSIF01.01 JPEG 2000 encoders and decoders. The remaining sections apply to *specific* preferred encodings. The conformance guidance is given in terms of decoder conformance and then encoder conformance.

The Joint Interoperability Test Command (JITC) at Ft. Huachuca, AZ maintains a set of test files that illustrate the use of these preferred encodings. The JITC is continually updating its test suite and also provides ISO/IEC 15444-1 Profile-0 and Profile-1 test files. See http://www.gwg.nga.mil/ntb/baseline/software/testfile/Jpeg2000/index.htm, for the current JITC JPEG 2000 test file suite. As resources become available, the JITC will continue to add test files to their website for community use. NATO partners and implementers are encouraged to contact the JITC to help establish suites of test files and to expand the repertoire of available files.

5.1 ISO/IEC 15444-1 and ISO/IEC 15444-4 Conformance

All JPEG 2000 decoders that claim conformance to *any* of the profiles contained within this document, shall be capable of decoding any codestream created by a compliant

ISO/IEC 15444-1 Profile-1 encoder within the restrictions set in ISO/IEC 15444-4 for Cclass 2 and subject to the NSIF Complexity Level (CLEVEL) supported by the decoder. CLEVELs are defined in Annex D of the NSIF01.01. A decoder might exceed certain Cclass 2 compliance attributes depending upon their supported CLEVEL. This is not a concern since the Cclasses represent minimum performance guarantees.

All software-based JPEG 2000 encoders that claim conformance to *any* of the profiles contained within this document shall be capable of generating JPEG 2000 codestreams that are compliant with ISO/IEC 15444-1 Profile-1. Note specialty hardware-based JPEG 2000 encoders are exempt from this requirement. *This requirement does not mean that all JPEG 2000 codestreams generated by an encoder will be ISO/IEC 15444-1 Profile-1 compliant.* In other words, an encoder may be placed into an operational mode where it generates codestreams that are not ISO/IEC 15444-1 Profile-1 compliant (this depends on their preferred encoding). For example, LVSD systems that make use of LPJE might routinely generate JPEG 2000 codestreams that are not compliant to ISO/IEC 15444-1 Profile-1. This would typically happen due to the image tile size being larger than 1,024 x 1,024 samples in size. *All* software-based encoders shall, however, be capable of generating an ISO/IEC 15444-1 Profile-1 compliant codestreams when they are asked to do so.

5.2 NSIF01.01 Conformance

All NSIF01.01 compliant BPJ2K01.10 decoders shall be able to fully decode JPEG 2000 codestreams that conform to NPJE, EPJE and TPJE within the restrictions set in ISO/IEC 15444-4 for Cclass 2 and subject to their NSIF CLEVEL. Note that decoders might exceed certain Cclass 2 compliance attributes depending upon their supported CLEVEL. NSIF01.10 decoders shall conform to the requirements of section 5.1. NSIF01.01 compliant decoders are not required to decode codestreams compliant to any other preferred encodings unless that codestream has been created within the bounds of ISO/IEC 15444-1 Profile-1. Note, this may be easily detected by examining the Rsiz value in the SIZ marker segment.

NSIF01.01 compliant BPJ2K01.10 encoders shall produce codestreams that are compliant to their implemented BPJ2K01.10 preferred encoding(s) and as constrained by their NSIF CLEVEL. NSIF compliant BPJ2K01.10 encoders that support the encoding recommendations of Appendix D (NPJE), Appendix E (EPJE) or Appendix F (TPJE) shall have a mode of operation where compressed data is produced within the constrained limits detailed in Appendix D Appendix E or Appendix F There is no requirement for NSIF compliant BPJ2K01.10 encoders to produce codestreams that are compliant to the encoding recommendations of Appendix G (LPJE) or Appendix H (SPJE).

5.3 NPJE (Appendix D) Conformance

All NPJE compliant BPJ2K01.10 decoders shall be able to fully decode JPEG 2000 codestreams that conform to NPJE within the restrictions set in ISO/IEC 15444-4 for Cclass 2 and subject to their NSIF Complexity Level (CLEVEL). Note that decoders might exceed certain Cclass 2 compliance attributes depending upon their supported CLEVEL. NPJE decoders shall conform to the requirements of section 5.1. NPJE compliant decoders are not required to decode codestreams compliant to any other preferred encodings unless that codestream has been created within the bounds of ISO/IEC 15444-1 Profile-1.

NPJE compliant BPJ2K01.10 encoders shall produce codestreams that are compliant to the NPJE profile (Appendix D) and as constrained by their NSIF CLEVEL. NPJE compliant BPJ2K01.10 encoders shall have a mode of operation where compressed data is produced within the constrained limits detailed in Appendix D . There is no requirement for NPJE compliant BPJ2K01.10 encoders to produce codestreams that are compliant to any other preferred encodings in BPJ2K01.10.

5.4 EPJE (Appendix E) Conformance

All EPJE compliant BPJ2K01.10 decoders shall be able to fully decode JPEG 2000 codestreams that conform to NPJE and EPJE within the restrictions set in ISO/IEC 15444-4 for Cclass 2 and subject to their NSIF Complexity Level (CLEVEL). Note that decoders might exceed certain Cclass 2 compliance attributes depending upon their supported CLEVEL. EPJE decoders shall conform to the requirements of section 5.1. EPJE compliant decoders are not required to decode codestreams compliant to any other preferred encodings (except NPJE) unless that codestream has been created within the bounds of ISO/IEC 15444-1 Profile-1.

EPJE compliant BPJ2K01.10 encoders shall produce codestreams that are compliant to the EPJE profile (Appendix E) and as constrained by their NSIF CLEVEL. EPJE compliant BPJ2K01.10 encoders shall have a mode of operation where compressed data is produced within the constrained limits detailed in Appendix E . There is no requirement for EPJE compliant BPJ2K01.10 encoders to produce codestreams that are compliant to any other preferred encodings in BPJ2K01.10.

5.5 TPJE (Appendix F) Conformance

All TPJE compliant BPJ2K01.10 decoders shall be able to fully decode JPEG 2000 codestreams that conform to NPJE, EPJE and TPJE within the restrictions set in ISO/IEC 15444-4 for Cclass 2 and subject to the NSIF Complexity Level (CLEVEL) supported by the implementation. Note that decoders might exceed certain Cclass 2 compliance attributes depending upon their supported CLEVEL. TPJE decoders shall conform to the requirements of section 5.1. TPJE compliant decoders are not required to decode codestreams compliant to any other preferred encodings (except NPJE and EPJE) unless that codestream has been created within the bounds of ISO/IEC 15444-1 Profile-1.

TPJE compliant BPJ2K01.10 encoders shall produce codestreams that are compliant to TPJE (Appendix F) and as constrained by their NSIF CLEVEL. TPJE compliant BPJ2K01.10 encoders shall have a mode of operation where compressed data is produced within the constrained limits detailed in Appendix F . There is no requirement for TPJE compliant BPJ2K01.10 encoders to produce codestreams that are compliant to any other preferred encodings in BPJ2K01.10. The TPJE profile is a superset of the NPJE and EPJE profiles. As such, a TPJE encoder is capable of producing NPJE and EPJE compliant codestreams. There is no *requirement*, however, for a TPJE encoder to produce NPJE or EPJE compliant codestreams. This decision is left up to the implementation.

5.6 LPJE (Appendix G) Conformance

All LPJE compliant BPJ2K01.10 decoders shall be able to fully decode JPEG 2000 codestreams that conform to LPJE and *all* other profiles within this document. Furthermore, LPJE compliant BPJ2K01.10 decoders shall decode *any* ISO/IEC 15444-1 Profile-1 codestream subject to *no* ISO/IEC 15444-4 Cclass restrictions. In other words, an LPJE decoder is expected to fully decode any ISO/IEC 15444-1 Profile-1 codestream. LPJE decoders shall conform to the requirements of section 5.1. Many LPJE codestreams will exceed the boundaries of ISO/IEC 15444-1 Profile-1 and LPJE decoders will possess resources that exceed those of ISO/IEC 15444-4 Cclass 2. This is necessary due to the large frame sizes of LVSD imagery.

LPJE compliant BPJ2K01.10 encoders shall produce codestreams that are compliant to LPJE (Appendix G). If an implementation chooses to use NSIF01.01 as a file wrapper for LVSD compressed codestreams, then the LVSD encoder in the implementation shall conform to its chosen NSIF CLEVEL. LPJE compliant BPJ2K01.10 encoders shall have a mode of operation where compressed data is produced within the constrained limits detailed in Appendix G . There is no requirement for LPJE compliant BPJ2K01.10 encoders to produce codestreams that are compliant to the encoding recommendations of any other profile in BPJ2K01.10. LPJE is a superset of all other BPJ2K01.10 preferred encodings and as such, an LPJE encoder is capable of producing NPJE, EPJE, TPJE and SPJE compliant codestreams. There is no *requirement*, however, for an LPJE encoder to produce codestreams compliant to any other profile in this document. This decision is left up to the implementation.

5.7 SPJE (Appendix H) Conformance

The SPJE (Appendix H) is included in this document for informative purposes. The preferred encoding described in Appendix H is intended to be equivalent to the JPEG 2000 profile described within STANAG 7023. Since STANAG 7023 still retains its JPEG 2000 profile, it remains the normative reference for the use of JPEG 2000 with NATO primary air reconnaissance imagery. If any inconsistencies are found between Appendix H and STANAG 7023, the text of STANAG 7023 takes precedence. The custodian of STANAG 7023 asked that SPJE be created within the BIIF profile so that a future version of STANAG 7023 may be written that simply points to this document for JPEG 2000 guidance. The SPJE has been reviewed by the STANAG 7023 custodian and it is believed to be correct. However, to avoid any potential conformance issues with the existence of two redundant profiles, implementers shall conform to STANAG 7023 and AEDP-9 at this time.

Although STANAG 7023 is the normative reference for this preferred encoding, we shall make some observations regarding decoders and encoders. SPJE allows a wide range of compression options that exceed those in ISO/IEC 1544-1 Profile-1. SPJE compliant BPJ2K01.10 decoders should be able to fully decode JPEG 2000 codestreams that conform to the SPJE profile and *all* other profiles within this document except possibly the LPJE profile. Furthermore, SPJE compliant BPJ2K01.10 decoders should be able to decode *any* ISO/IEC 15444-1 Profile-1 codestream subject to *no* ISO/IEC 15444-4 Cclass restrictions. In other words, an SPJE decoder is expected to fully decode any ISO/IEC 15444-1 Profile-1 codestream. SPJE decoders shall conform to the requirements of section 5.1. Note that SPJE codestreams may exceed the boundaries of ISO/IEC 15444-1 Profile-1 and SPJE decoders may possess resources that exceed those of ISO/IEC 15444-4 Cclass 2.

SPJE compliant BPJ2K01.10 encoders shall produce codestreams that are compliant to Appendix H . SPJE compliant BPJ2K01.10 encoders shall have a mode of operation where compressed data is produced within the constrained limits detailed in STANAG 7023 (Appendix H). SPJE is a superset of the NPJE, EPJE and TPJE and as such, an SPJE encoder should be capable of producing NPJE, EPJE and TPJE compliant codestreams. There is no *requirement*, however, for an SPJE encoder to produce codestreams compliant to any other profile in this document. This decision is left up to the implementation.

6 Profile Registration

This profile is registered under the provisions and procedures defined in Annex C of ISO/IEC 12087-5:1998 and through the ISO/IEC processes found in ISO/IEC 9973.

7 JPEG 2000 Profile and Limitations

The following limitations are defined for the BPJ2K01.10. The basis of this section is the limits that are associated with ISO/IEC 15444-1, Profile-1. All compliant BPJ2K01.10 decoders, regardless of which profiles they support, shall be able to properly decode compressed data that is within the limits of this profile. All NSIF01.01 compliant encoders shall produce compressed data that is within the limits of this profile. LVSD, TPJE, and SPJE compliant encoders (software-based only, see section 5.1) shall be capable of creating codestreams compliant to Profile-1 when they are requested to do so. It is recommended that LVSD, TPJE and SPJE compliant encoders produce codestreams compliant to the following limitations whenever possible. It is furthermore recommended that all encoders adhere to the preferred encoding recommendations in this section when producing codestreams compliant to this Profile-1.

7.1 Markers and Marker Segments Limits

Markers and marker segments are defined in Table 7-1. This table defines each marker's value, whether it required (Req.), not allowed (NA), or optional (Opt.), and if there are any restrictions or dependencies. There are only three places that a marker can be present: the main header, tile header, or the bitstream. The bitstream, as defined in JPEG 2000 Part 1, is the codestream but does not include the main header or the tile header. Each of these markers and marker segments are further defined in this section.

Table 7-1 Marker and marker segment requirements within a JPEG 2000 codestream

Marker	Value	Main Header	Tile-part Header	Bitstream	Restrictions/Remarks
SOC	0xFF4F	Req.	NA	NA	Required as first marker in the codestream.
SOT	0xFF90	NA	Req.	NA	Required as first marker in each tile-part.
SOD	0xFF93	NA	Req.	NA	Last marker in each tile-part header.
EOC	0xFFD9	NA	NA	Req.	Required as the last marker in the codestream.
SIZ	0xFF51	Req.	NA	NA	Required as second marker (first marker segment) in main header. Immediately follows SOC.
COD	0xFF52	Req.	Opt.	NA	One and only one required in main header. Optionally no more than one shall appear in first tile-part header of a tile. Indicates usage of SOP and EPH.
COC	0xFF53	Opt.	Opt.	NA	Optional. No more than one COC per component in main header or in the first tile-part header of a tile.
RGN	0xFF5E	Opt.	Opt.	NA	Optional. No more than one per component in main header or first tile-part header of a tile. When present in main header applies to all tiles except those with a RGN marker segment. When used in a tile-part header, applies only to one component within that tile.
QCD	0xFF5C	Req.	Opt.	NA	One and only one required in main header. No more than one in first tile-part headers.
QCC	0xFF5D	Opt.	Opt.	NA	Optional. No more than one per component in main header or first tile-part headers.
POC	0xFF5F	Opt.	Opt.	NA	Optional. No more than one in the main or any tile-part header. Must appear in a tile-part header before the packets which it describes. Required if there are progression order changes different from the main or tile header COD.
TLM	0xFF55	Opt.	NA	NA	Optional. There may be multiple TLM marker segments in main header.
PLM	0xFF57	Opt.	NA	NA	Optional. There may be multiple PLM marker segments in main header.

Table 7-1 Marker and marker segment requirements within a JPEG 2000 codestream					
Marker	**Value**	**Main Header**	**Tile-part Header**	**Bitstream**	**Restrictions/Remarks**
PLT	0xFF58	NA	Opt.	NA	Optional. There may be multiple PLT marker segments per tile. Must appear in any tile-part header before the packets whose lengths they describe.
PPM	0xFF60	Opt.	NA	NA	If a PPM marker segment is present, all packet headers shall be found in main header and a PPT marker segment is not allowed.
PPT	0xFF61	NA	Opt.	NA	If a PPT marker segment appears in a tile-part header, all packet headers for the given tile shall follow. The PPT marker segment must appear in any tile-part header before the packets whose headers are contained in the PPT marker segment appear.
SOP	0xFF91	NA	NA	Opt.	May be used in front of each packet, shall not be used unless indicated in the proper COD marker segment. Whether or not an SOP marker is used, Nsop must be incremented for each packet in the bitstream. If packet headers are moved into a PPT or PPM marker segment, the SOP markers may appear immediately before the packet bodies in the bitstream.
EPH	0xFF92	Opt.	Opt.	Opt.	Shall not be present unless indicated in the proper COD marker segment. If EPH markers are signaled, they must appear for every packet header. If the packet headers are moved into a PPM or PPT marker segment, the EPH markers shall appear after the packet headers in the PPM or PPT marker segments.
CRG	0xFF63	Opt.	NA	NA	Only one CRG may appear in the main header and it applies for all tiles. This marker segment has no effect on decoding the codestream. It is for informational purposes only.
COM	0xFF64	Opt.	Opt.	NA	Informational only, repeat as many times as desired in the main or tile-part headers. This marker segment has no effect on decoding the codestream.

7.2 Delimiting Markers and Marker Segments

The delimiting markers shall be present in all JPEG 2000 compressed imagery. Each delimiting marker must be present in a compliant JPEG 2000 codestream. A codestream shall have only one SOC and EOC marker and at least one tile-part. Each tile-part has one SOT and one SOD marker.

Table 7-2 Start of Codestream (15444-1 Annex A.4.1)

Parameter	Size (bits)	Values	Notes
SOC	16	0xFF4F	Start of codestream marker.

Table 7-3 Start of tile-part (15444-1 Annex A.4.2)

Parameter	Size (bits)	Values	Notes
SOT	16	0xFF90	Start of tile-part marker code.
Lsot	16	10	Length of marker segment.
Isot	16	0 – 65,534	Tile index. Tiles are in raster order starting at index 0.
Psot	32	0, 14 – (2^{32} –1)	The length in bytes from the beginning of SOT marker segment of the tile-part to the end of the data of that tile-part. It is recommended a Psot of 0 be replaced by the actual tile length when a JPEG 2000 codestream is incorporated into NSIF. If Psot = 0 is maintained in an NSIF file, the current tile part will be interpreted to extend to the end of the current NSIF image segment. If Psot=0 is maintained in any JPEG 2000 format, the tile-part is interpreted to extend to the end of the file.
TPsot	8	0 – 254	Tile-Part index.
TNsot	8	0 – 255	0 = Number of tile-parts of this tile in the codestream is not defined in this header. 1 – 255 number of tile-parts of this tile in the codestream.

Table 7-4 Start of data marker (15444-1 Annex A.4.3)

Parameter	Size (bits)	Values	Notes
SOD	16	0xFF93	Start of data marker.

Table 7-5 End of codestream (15444-1 Annex A.4.4)

Parameter	Size (bits)	Values	Notes
EOC	16	0xFFD9	End of codestream marker.

7.3 Fixed Information Marker Segment

This marker segment includes information required to properly decode the image. There shall be a SIZ marker segment in the main header immediately after the SOC marker segment.

Table 7-6 Image and tile size (15444-1 Annex A.5.1)

Parameter	Size (bits)	Values	Notes
SIZ	16	0xFF51	Image and tile size marker.
Lsiz	16	41 – 49,190	Length of marker segment.
Rsiz	16	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010 0000 0000 0000 0011 0000 0000 0000 0100	Rsiz = 0000 0000 0000 0000. Indicates that the full capabilities described by ISO/IEC IS15444-1 are required. Rsiz = 0000 0000 0000 0001. Indicates that the codestream is Profile 0 compliant. Rsiz = 0000 0000 0000 0010. Indicates that the codestream is Profile-1 compliant. Rsiz = 0000 0000 0000 0011 and Rsiz = 0000 0000 0000 0100 see below* This profile will be compliant with 0000 0000 0000 0010 = ISO profile 1.
Xsiz	32	$1 – (2^{31} – 1)$	Reference grid width. This profile is limited to Xsiz < 2^{31}.
Ysiz	32	$1 – (2^{31} – 1)$	Reference grid height. This profile is limited to Ysiz < 2^{31}.
XOsiz	32	$0 – (2^{31} – 2)$	Horizontal offset from the origin of the reference grid to the left side of the image area. This profile is limited to XOsiz < 2^{31}–1.
YOsiz	32	$0 – (2^{31} – 2)$	Vertical offset from the origin of the reference grid to the top of image area. This profile is limited to YOsiz < 2^{31}–1.
XTsiz	32	$1 – (2^{31} – 1)$	Width of one reference tile with respect to the reference grid. For this profile: XTsiz/min($XRsiz^i$, $YRsiz^i$) ≤ 1024 XTsiz = YTsiz or one tile for the whole image: YTsiz+YTOsiz> = Ysiz XTsiz+XTOsiz> = Xsiz
YTsiz	32	$1 – (2^{31} – 1)$	Width of one reference tile with respect to the reference grid. For this profile: XTsiz/min(XRsizi, YRsizi) ≤ 1024 XTsiz = YTsiz or one tile for the whole image: YTsiz+YTOsiz> = Ysiz XTsiz+XTOsiz> = Xsiz
XTOsiz	32	$0 – (2^{31} – 2)$	Horizontal offset from the origin of the reference grid to the left edge of the first tile. This profile is limited to XTOsiz < 2^{31}–1.

Table 7-6 Image and tile size (15444-1 Annex A.5.1)			
Parameter	**Size (bits)**	**Values**	**Notes**
YTOsiz	32	$0 - (2^{31} - 2)$	Vertical offset from the origin of the reference grid to the top edge of the first tile. This profile is limited to YTOsiz < $2^{31}-1$.
Csiz	16	1 – 16,384	The number of components in the image.
$Ssiz^i$	8	0000 0000 – 0010 0101 or 1000 0000 – 1010 0101	0xxx xxxx Unsigned data 1xxx xxxx signed data x000 0000 – x010 0101 bit depth of data = value + 1
$XRsiz^i$	8	1 – 255	Horizontal sub-sampling on the reference grid with respect to the i^{th} component.
$YRsiz^i$	8	1 – 255	Vertical sub-sampling on the reference grid with respect to the i^{th} component.

* Two new values for the Rsiz parameter have been added in ISO/IEC 15444-1 Amendment 1, 2006. These values were assigned to the Digital Cinema Initiative to indicate which DCI profile (2K or 4K) the codestream adheres to. See the document "Digital Cinema System Specification, Version 1.1", April 12, 2007 for further details. NSIF implementations are not allowed to set Rsiz = 0000 0000 0000 0011 or Rsiz = 0000 0000 0000 0100.

7.4 Functional Marker Segments

The functional marker segments define what parameters were used in the compression of a given tile or an image. These marker segments apply to the entire tile when in the tile-part header and to the image when in the main header. Markers in the tile header supersede markers in the main header. Table 7-7 gives the COD marker segment. The COD marker segment is required in the main header of the codestream and it contains the default encoding parameters applied to all components and tiles unless it is overridden (see COC below).

Table 7-7 Coding style default (15444-1 Annex A.6.1)			
Parameter	**Size (bits)**	**Values**	**Notes**
COD	16	0xFF52	Coding style default marker.
Lcod	16	Lcod = 12 (maximal precincts) Lcod = 13 + N_{Levels} (user-defined)	Length of marker segment.
Scod	8	0000 0000 – 0000 0111 (see Table 7-8)	Coding style parameters.
SGcod	32	Defined below	
Progression order	8	0000 0000 – 0000 0100 (see Table 7-9)	Defines the progression order.
Number of layers (N_{Layers})	16	1 – 65,535	Number of layers in the image.

Table 7-7 Coding style default (15444-1 Annex A.6.1)

Parameter	Size (bits)	Values	Notes
Multiple component transform	8	0000 0000 – 0000 0001	0000 0000 = No component transform used 0000 0001 = Component transform used
SPcod	Variable	Defined below	
Number of decomposition levels (N_{Levels})	8	0 – 32	Number of wavelet transform decomposition levels.
Code-block width	8	0000 0000 – 0000 1000	Code-block width exponent offset value, xcb = value+2. For this profile this is limited to xcb ≤ 6.
Code-block height	8	0000 0000 – 0000 1000	Code-block width exponent offset value, ycb = value+2. For this profile this is limited to ycb ≤ 6.
Code-block style	8	0000 0000 – 0011 1111 (see Table 7-10)	Arithmetic coding parameters.
Transformation	8	0000 0000 – 0000 0001	Wavelet filter 0000 0000 = 9-7 irreversible filter 0000 0001 = 5-3 reversible filter
Precinct size	Variable	NA or 0000 0000 – 1111 1111 (see Table 7-11)	Precinct size (only if defined, Scod = xxxx xxx1).

Code block sizes in JPEG 2000 are limited to a maximum number of 4,096 coefficients within the code-block. Furthermore, the minimum dimension of a code-block in the row or columnar dimension is 4. Code-blocks represent the fundamental limit on spatial region of access within the JPEG 2000 codestream (barring any code-block/precinct interactions). Code-blocks also prevent error propagation in the entropy encoder. Small code-blocks, therefore, would seem like a good idea. The larger the code-blocks are, however, the greater the entropy coding efficiency of the arithmetic encoder will be. In general, code-blocks of size 32 x 32 or 64 x 64 are good choices. Hardware implementations may want to use the smaller code-block size to minimize memory costs. Arguments can also be made for using rectangular (4 x 1,024) code-blocks when performing stripe processing on very large images.

Table 7-8 Scod coding style parameters (15444-1 Annex A.6.1)

Value (bits)	Coding style
0000 0xx0	Entropy coder, precincts with PPx = 15 and PPy = 15 (maximal precincts)
0000 0xx1	Entropy coder with user-defined precincts
0000 0x0x	No SOP marker segments used

Table 7-8 Scod coding style parameters (15444-1 Annex A.6.1)	
Value (bits)	**Coding style**
0000 0x1x	SOP marker segments may be used
0000 00xx	No EPH marker used
0000 01xx	EPH marker shall be used

Table 7-9 Progression order (SGcod or Ppoc parameters, 15444-1 Annex A.6.1)	
Value (bits)	**Progression order**
0000 0000	Layer-resolution level-component-position progression (L-R-C-P)
0000 0001	Resolution level-layer-component-position progression (R-L-C-P)
0000 0010	Resolution level-position-component-layer progression (R-P-C-L)
0000 0011	Position-component-resolution level-layer progression (P-C-R-L)
0000 0100	Component-position-resolution level-layer progression (C-P-R-L)

Table 7-10 Code-block style (SPcod and SPcoc parameters, 15444-1 Annex A.6.1)	
Value (bits)	**Code-block style**
00xx xxx0 00xx xxx1	No selective arithmetic coding bypass Selective arithmetic coding bypass
00xx xx0x 00xx xx1x	No reset of context probabilities on coding pass boundaries Reset context probabilities on coding pass boundaries
00xx x0xx 00xx x1xx	No termination on each coding pass Termination on each coding pass
00xx 0xxx 00xx 1xxx	No vertically causal context Vertically causal context
00x0 xxxx 00x1 xxxx	No predictable termination Predictable termination
000x xxxx 001x xxxx	No segmentation symbols are used Segmentation symbols are used

Table 7-11 Precinct width and height parameters (15444-1 Annex A.6.1)	
Value (bits) MSB LSB	**Code-block style**
xxxx 0000 xxxx 1111	4 LSBs are the precinct width exponent, PPx = Value.
0000 xxxx 1111 xxxx	4 MSBs are the precinct height exponent, PPy = Value.

JPEG 2000 allows for several options regarding "code-block style" (see Table 7-10). These parameters are used to control the behavior of the arithmetic encoder. Annex D in ISO/IEC 15444-1 describes the meaning of these parameters. In general, these options relate to speeding up the arithmetic encoding (selective bypass) and improving error resiliency/detection (segmentation symbols, termination, resetting context probabilities, vertically causal). Some of the above options also reduce memory costs to a small extent. For more discussion on these options see Annex J in ISO/IEC 15444-1 as well.

The COD coding parameters can be overridden. A COC marker segment may be used to override the coding parameters for a single specified component. To override the coding parameters for more than one component, multiple COC marker segments must be used (to override coding parameters for all components a tile-part COD marker segment may be used). If the COC marker appears in the main header, then the default coding parameters (as defined by the main header COD marker segment) for the specified component are replaced by those in the COC marker segment for the entire image. If the COC marker segment appears in a first tile-part header, then the coding parameters for the specified component are replaced for that tile only. We can express the relationships between main and tile-part COD and COC marker segments as follows:

Tile-part COC > Tile-part COD > Main Header COC > Main Header COD

This illustrates that tile-part COC marker segments supersede tile-part COD marker segments, which supersede main header COC marker segments, which supersede the main header COD marker segment. Table 7-12 gives the COC marker segment. Note that the progression order, number of layers and use of the multiple component transform cannot be modified on a per component basis.

Table 7-12 Coding style component (15444-1 Annex A.6.2)

Parameter	Size (bits)	Values	Notes
COC	16	0xFF53	Coding style component marker.
Lcoc	16	9 (max precincts & Csiz < 257) 10 (max precincts & Csiz ≥ 257) $10 + N_{Levels}$ (user-defined & Csiz < 257) $11 + N_{Levels}$ (user-defined & Csiz ≥ 257	Length of marker segment.

Table 7-12 Coding style component (15444-1 Annex A.6.2)			
Parameter	**Size (bits)**	**Values**	**Notes**
Ccoc	8 16	0 – 255 (8 bits, Csiz < 257) 0 – 16,383 (16 bits, Csiz ≥ 257)	Component index to which this marker segment applies.
Scoc	8	0000 0000 – 0000 0001 (see Table 7-13)	Coding style parameters.
SPcoc	Variable	Defined below	
Number of decomposition levels (N_{Levels})	8	0 – 32	Number of wavelet transform decomposition levels.
Code-block width	8	0000 0000 – 0000 1000	Code-block width exponent offset value, xcb = value+2. For this profile this is limited to xcb ≤ 6.
Code-block height	8	0000 0000 – 0000 1000	Code-block width exponent offset value, ycb = value+2. For this profile this is limited to ycb ≤ 6.
Code-block style	8	0000 0000 – 0011 1111 (see Table 7-10)	Arithmetic coding parameters.
Transformation	8	0000 0000 – 0000 0001	Wavelet filter 0000 0000 = 9-7 irreversible filter 0000 0001 = 5-3 reversible filter
Precinct size	Variable	NA or 0000 0000 – 1111 1111 (see Table 7-11)	Precinct size (only if defined, Scoc = xxxx xxx1).

Table 7-13 Scoc coding style parameters (15444-1 Annex A.6.2)	
Value (bits)	**Coding style**
0000 0000	Entropy coder, precincts with PPx = 15 and PPy = 15 (maximal precincts)
0000 0001	Entropy coder with user-defined precincts

The region of interest marker segment (RGN) is shown in Table 7-14. This marker segment is used for region of interest coding. Encoders can shift the wavelet coefficients corresponding to a spatial region of interest up (by left bit-shift) above the most significant bitplane of the remaining background wavelet coefficients. This has the effect that these wavelet coefficients will be encoded first before all other wavelet coefficients. The RGN marker segment alerts the decoder that this has been done so that the process can be reversed during decoding.

Table 7-14 Region of interest (15444-1 Annex A.6.3)			
Parameter	**Size (bits)**	**Values**	**Notes**
RGN	16	0xFF5E	Region of interest marker.
Lrgn	16	5 – 6	Length of this marker segment.
Crgn	8 16	0 – 255 (8 bits, Csiz < 257) 0 – 16,383 (16 bits, Csiz ≥ 257)	Component index to which this marker segment applies.
Srgn	8	0000 0000 Implicit ROI (maximum shift)	All other values reserved.
SPrgn	8	0 – 37	Binary shifting of ROI coefficients above the background.

The QCD marker segment (see Table 7-15) is used in the main header to indicate quantization step-sizes that are valid for all tile-parts. The QCD marker segment is required in the main header – the values in this marker segment in the main header are used for components that do not override these values with a main header QCC and for all tiles that do not override these values with a tile-specific QCD or QCC in that tile's header.

Table 7-15 Quantization default (15444-1 Annex A.6.4)			
Parameter	**Size (bits)**	**Values**	**Notes**
QCD	16	0xFF5C	Quantization default marker.
Lqcd	16	No quantization: $Lqcd = 4 + 3 \cdot N_{Levels}$ Scalar quantization derived: Lqcd = 5 Scalar quantization expounded: $Lqcd = 5 + 6 \cdot N_{Levels}$	Length of this marker segment. For the 5-3R wavelet, no quantization is used. For the 9-7I wavelet, scalar derived or expounded quantization is used.
Sqcd	8	xxx0 0000 xxx0 0001 xxx0 0010 000x xxxx - 111x xxxx (see Table 7-16)	With 5-3R filter: No quantization. With 9-7I filter: Scalar expounded quantization or scalar derived. Number of guard bits 0 – 7.
SPqcd[i]	8 (5-3R) 16 (9-7I)	variable	With 5-3R wavelet With 9-7I wavelet

Table 7-16 Quantization values (Sqcd and Sqcc parameters, 15444-1 Annex A.6.4)			
Value (bits)	**Quantization Style**	**SPqcd^{i} Size (bits)**	**SPqcd or SPqcc usage**
xxx0 0000	No quantization (5-3R wavelet)	8	Table 7-17
xxx0 0001	Scalar derived (values signaled for NL_{LL} subband only)	16	Table 7-18
xxx0 0010	Scalar expounded. One step size signaled for each subband.	16	Table 7-18
000x xxxx – 111x xxxx	Number of guard bits (0 – 7)		

Table 7-17 Reversible step size (SPqcd^{i} and SPqcc^{i} parameters, 15444-1 Annex A.6.4)	
Value (bits)	**Reversible step size values**
0000 0xxx – 1111 1xxx	Exponent, ε_b, of the reversible dynamic range signaled for each subband. See equation E.2 in ISO/IEC 15444-1 (note 15444-1 makes reference to equation E.5, this is incorrect).

Table 7-18 Quantization step size (SPqcd^{i} and SPqcc^{i} parameters, 15444-1 Annex A.6.4)	
Value (bits)	**Quantization step size values**
xxxx x000 0000 0000 – xxxx x111 1111 1111	Mantissa, μ_b, of the quantization step size value. See Equation E.3 in ISO/IEC 15444-1.
0000 0xxx xxxx xxxx – 1111 1xxx xxxx xxxx	Exponent, ε_b, of the quantization step size value. See Equation E.3 in ISO/IEC 15444-1.

The QCC marker segment may be used to override the quantization parameters for a single specified component. To override the quantization parameters for more than one component, multiple QCC marker segments must be used (to override coding parameters for all components a tile-part QCD marker segment may be used). If the QCC marker appears in the main header, then the default quantization parameters (as defined by the main header QCD marker segment) for the specified component are replaced by those in the QCC marker segment for the entire image. If the QCC marker segment appears in a first tile-part header, then the quantization parameters for the specified component are replaced for that tile only. We can express the relationships between main and tile-part QCD and QCC marker segments as follows:

Tile-part QCC > Tile-part QCD > Main Header QCC > Main Header QCD

Table 7-19 gives the QCC marker. Two guard bits are recommended for both 5-3R and 9-7I wavelet processing. Two guard bits are sufficient for the vast majority of imagery. There can arise some circumstances where two guard bits are not sufficient but these are very rare.

Table 7-19 Quantization component (15444-1 Annex A.6.5)

Parameter	Size (bits)	Values	Notes
QCC	16	0xFF5D	Quantization component marker.
Lqcc	16	For Csiz < 257 No quantization: Lqcd = 5 + 3·N_{Levels} Scalar quantization derived: Lqcd = 6 Scalar quantization expounded: Lqcd = 6 + 6·N_{Levels}	Length of marker segment.
		For Csiz ≥ 257 No quantization: Lqcd = 6 + 3·N_{Levels} Scalar quantization derived: Lqcd = 7 Scalar quantization expounded: Lqcd = 7 + 6·N_{Levels}	
Cqcc	8 16	0 – 255 (8 bits, Csiz < 257) 0 – 16,383 (16 bits, Csiz ≥ 257)	Component index to which this marker segment applies.
Sqcc	8	xxx0 0000 xxx0 0001 xxx0 0010 000x xxxx - 111x xxxx (see Table 7-16)	With 5-3R filter: No quantization. With 9-7I filter: Scalar expounded quantization or scalar derived. Number of guard bits 0 – 7.
SPqcc[i]	8 (5-3R) 16 (9-7I)	variable	With 5-3R wavelet With 9-7I wavelet

The POC marker segment, Table 7-20, is used to change progression orders within a codestream. POC marker segments require a good understanding of JPEG 2000 codestream construction and a sophisticated JPEG 2000 codec. ISO/IEC 15444-1 Profile-1 compliant decoders are required to handle POC marker segments. The POC marker segment allows full control over the ordering of codestream data within a file. For most applications the POC marker segment is not necessary, one progression order will suffice for all codestream data.

The Digital Cinema Initiative (DCI) has prescribed the use of a POC marker segment within their 4K codestream distributions (see the DCI System Specification). The POC marker segment is used to organize 4K codestreams so that it is easier for 2K compliant systems to parse out a 2K version of the codestreams.

Table 7-20 Progression Order Change (15444-1 Annex A.6.6)

Parameter	Size (bits)	Values	Notes
POC	16	0xFF5F	Progression order change marker segment.
Lpoc	16	9 – 65,535	Length of marker segment.
$RSpoc^i$	8	0 – 32	Resolution level index (inclusive) for the start of i^{th} progression. One value for each progression change.
$CSpoc^i$	8 (Csiz < 257) 16 (Csiz ≥ 257)	0 – 255 (Csiz < 257) 0 – 16,383 (Csiz ≥ 257)	Component index (inclusive) for the start of i^{th} progression. Components are indexed 0, 1, 2, *etc.* One value for each progression change.
$LYEpoc^i$	16	1 – 65,535	Layer index (exclusive) for the end of i^{th} progression. Layer index always starts at zero for every progression. Packets that have already been included in the codestream are not included again. One value for each progression change.
$REpoc^i$	8	$(RSpoc^i + 1) – 33$	Resolution Level index (exclusive) for the end of i^{th} progression. One value for each progression change.
$CEpoc^i$	8 (Csiz < 257) 16 (Csiz ≥ 257)	$(CSpoc^i + 1)$ – 255, 0 (Csiz < 257) $(CSpoc^i + 1)$ – 16,384, 0 (Csiz ≥ 257) Note: 0 is interpreted as 256	Component index (exclusive) for the end of i^{th} progression. Components are indexed 0, 1, 2, *etc.* One value for each progression change.
$Ppoc^i$	8	0000 0000 – 0000 0100 (see Table 7-9)	Progression order for i^{th} progression. One value for each progression change.

7.5 Pointer Marker Segments

The pointer markers segments are used to gain quick access to desired data for parsing, chipping, and decoding. The marker segments define either lengths of a data set or pointers to the start of a data set. The tile-part length marker segment (see Table 7-21) has the same length information as the start of tile marker segments in each tile-part, but this in formation is collected up front in the main header. This marker segment can be used to quickly access and chip a given tile or set of tiles in a compressed image.

Table 7-21 Tile-part lengths (15444-1 Annex A.7.1)

Parameter	Size (bits)	Values	Notes
TLM	16	0xFF55	Tile-part lengths marker.
Ltlm	16	$Ltlm = \begin{cases} 4+2\cdot N_{tpm} & ST=0,\ SP=0 \\ 4+3\cdot N_{tpm} & ST=1,\ SP=0 \\ 4+4\cdot N_{tpm} & ST=2,\ SP=0 \\ 4+4\cdot N_{tpm} & ST=0,\ SP=1 \\ 4+5\cdot N_{tpm} & ST=1,\ SP=1 \\ 4+6\cdot N_{tpm} & ST=2,\ SP=1 \end{cases}$ N_{tpm} = number of tile-parts in this TLM marker segment	Length of marker segment. See Table 7-22.
Ztlm	8	0 – 255	Index of this marker segment relative to all other TLM marker segments present in the current header.
Stlm	8	0x00 0000 0x01 0000 0x10 0000 00xx 0000 01xx 0000	See Table 7-22.
$Ttlm^i$	0 if ST = 0 8 if ST = 1 16 if ST = 2	Tiles in order 0 – 254 0 – 65,534	Tile index for the i^{th} tile-part. Either none or one value for every tile-part.
$Ptlm^i$	16 if SP = 0 32 if SP = 1	14 – 65,535 14 – (2^{32} – 1)	The length, in bytes, from the beginning of the SOT marker of the i^{th} tile-part to the end of the codestream data for that tile-part. There should be one Ptlm for every tile-part.

Table 7-22 Stlm size parameters (15444-1 Annex A.7.1)

Value (bits)	Size parameters
0x00 0000	ST = 0; Ttlm parameter is 0 bits, only one tile-part per tile and the tiles are in index order without omission or repetition.
0x01 0000	ST = 1: Ttlm parameter 8 bits.
0x10 0000	ST = 2: Ttlm parameter 16 bits.

Table 7-22 Stlm size parameters (15444-1 Annex A.7.1)	
Value (bits)	**Size parameters**
00xx 0000	SP = 0; Ptlm parameter 16 bits.
01xx 0000	SP = 1; Ptlm parameter 32 bits.
	All other values reserved.

The entropy-coded wavelet coefficient data in JPEG 2000 are organized into packets comprised of a packet header and packet body. Both the packet header and packet body are entropy-coded and have variable length. To locate a desired piece of codestream data, a decoder must parse and decode the packet headers to learn the lengths of the entropy-coded packet bodies which contain the wavelet coefficient information. The packet length marker segments (PLM and PLT) collect together the lengths of the packets so that a decoder may rapidly skip through packets to find the ones it wants without having to decode all intermediate packet headers.

The PLM marker segment (packet lengths, main header) collects packet length information across all tile-parts and places the lengths in the main header. The marker segment is shown in Table 7-23. It is possible to overload the PLM marker segment due to the size of the $Nplm^i$ field being 8 bits. If there are more than 255 packets in a tile-part or the total amount of length data for a given tile-part requires more than 255 bytes for any given tile-part, then the PLM marker segment shall not be used. The amount of information contained in a PLM (or PLT) marker segment can be considerable. The length data might therefore be spread across more than one PLM or PLT marker segment. The Zplm (or Zplt) parameter is an index that is used to reassemble the data into the correct order.

Table 7-23 Packet length, main header (15444-1 Annex A.7.2)			
Parameter	**Size (bits)**	**Values**	**Notes**
PLM	16	0xFF57	Packet lengths, main header, marker.
Lplm	16	4 – 65,535	Length of marker segment.
Zplm	8	0 – 255	Index of this marker segment relative to all other PLM marker segments present in the main header.
$Nplm^i$	8	0 – 255	Number of bytes of Iplm information for the i^{th} tile-part in the order found in the codestream. One value for each tile-part.
$Iplm^{ij}$	variable	variable (see Table 7-24)	Length of the j^{th} packet in the i^{th} tile-part. If packet headers are stored with the packet bodies this length includes the packet header. If packet headers are stored in a PPM or PPT marker segment this length does not include the packet header length. One range of values for each tile-part. One value for each packet in the tile.

Table 7-24 Iplm, Iplt packet lengths (15444-1 Annex A.7.2)			
Parameter	**Size (bits)**	**Values**	**Meaning of Iplm or Iplt values**
Packet lengths	8 bits repeated as necessary	0xxx xxxx	Last 7 bits of packet length, terminate number
		1xxx xxxx	Continue reading
		x000 0000 – x111 1111	7 bits of packet length

The packet length has been broken into 7 bit segments which are sent in order from the most significant segment to the least significant segment. Furthermore, the bits in the most significant segment are right justified to the byte boundary. For example, a packet length of 128 is signaled as 1000 0001 0000 0000, while a length of 512 is signaled as 1000 0100 0000 0000.

The PLT marker segment performs a similar function to the PLM marker segment, except the length information is embedded in the tile-part headers. The PLT need not be placed in the first tile-part header for a tile, but it must appear in a tile-part header prior to the packets whose lengths it contains. The PLT marker segment does not suffer from the limitations of the PLM marker segment since it does not aggregate length information across multiple tiles.

Table 7-25 Packet length, tile-part headers (15444-1 Annex A.7.3)			
Parameter	**Size (bits)**	**Values**	**Notes**
PLT	16	0xFF58	Packet lengths, tile-part header, marker.
Lplt	16	4 – 65,535	Length of marker segment.
Zplt	8	0 – 255	Index of this marker segment relative to all other PLT marker segments in the current tile-part header.
$Iplt^i$	variable	variable (see Table 7-24)	Length of the i^{th} packet in this tile. If packet headers are stored with the packet bodies this length includes the packet header. If packet headers are stored in a PPM or PPT marker segment this length does not include the packet header length. One value for each packet in the tile.

ISO/IEC 15444-1 Profile-1 allows use of both PLM and PLT in the same codestream, but it is not recommended by this profile. Due to the limitations of the PLM marker segment, the PLT marker segment is preferred. However, ISO/IEC 15444-1 Profile-1 compliant decoders must properly handle the PLM marker segment.

There are two more marker segments that may be used to assist in the parsing of packet data. These are the PPM (packed packet headers, main header) and the PPT (packed packet headers, tile-part header). These marker segments aggregate the packet headers (not the packet bodies) into either the main header or tile-part headers. This allows decoders to bulk read the headers and process them instead of parsing them out of the codestream. The idea is to minimize the number of disk reads needed to parse the codestream. If a decoder has all of the packet header information for a codestream, it knows exactly how many coding passes and bytes of entropy-coded data from each code-block are present within each layer of the codestream. This allows for fast parsing decisions to be made.

If the PPM or PPT marker segments are used to relocate the packet headers, then the PLM and PLT marker segments packet length information describes only the packet bodies. It does not include the packet headers in the lengths. The presence of PPM and PPT marker segments also influences the behavior of the SOP and EPH marker segments (see below). Table 7-26 gives the PPM marker segment. The PPM marker segment aggregates packet header information across all tile-parts into the main header. This allows a decoder to quickly decode all packet header information at once, but it comes at a price.

The PPM marker segment loads all of the packet headers into the main header of the file. While the packet header information is necessary to understand the entropy-encoded wavelet coefficients, it conveys no image information. The packet headers are side information necessary to understand the layout of the compressed image data. The amount of packet header information can be considerable for a large image. In progressive transmission systems (*e.g.* JPIP streaming, FTP transfer) placing all of this data up front ("front-loading") rather than letting it be distributed throughout the bitstream can affect performance. The recipient of the streamed data might have to wait an unacceptable amount of time before receiving any useable imagery data.

Table 7-26 Packed packet headers, main header (15444-1 Annex A.7.4)

Parameter	Size (bits)	Values	Notes
PPM	16	0xFF60	Packed packet headers, main header, marker.
Lppm	16	8 – 65,535	Length of marker segment. Note there is an error in ISO/IEC 15444-1.
Zppm	8	0 – 255	Index of this marker segment relative to all other PPM marker segments present in the main header.
$Nppm^i$	32	$0 – (2^{32} – 1)$	Number of bytes of Ippm information for the i^{th} tile-part in the order found in the codestream. One value for each tile-part.
$Ippm^{ij}$	variable	Packet header data	The packet header data is the same as that which would appear in the bitstream (see Annex B.10 of ISO/IEC 15444-1).

The PPT marker segment is given in Table 7-27. It performs a similar function to the PPM marker segment, but it places the packet header information in tile-part headers instead. PPT marker segments need not appear in the first tile-part header, they need only appear in a tile-part header that is located in the compressed file before the packet data that the packet headers describe. The PPT marker segment is a compromise between placing all of the packet headers in the main header and letting them be distributed through out the bitstream.

Table 7-27 Packed packet headers, tile-part header (15444-1 Annex A.7.5)			
Parameter	**Size (bits)**	**Values**	**Notes**
PPT	16	0xFF61	Packed packet headers, tile-part header, marker.
Lppt	16	4 – 65,535	Length of marker segment.
Zppt	8	0 – 255	Index of this marker segment relative to all other PPT marker segments present in the tile-part header.
Ippt[i]	variable	Packet header data	The packet header data is the same as that which would appear in the bitstream (see Annex B.10 of ISO/IEC 15444-1).

ISO/IEC 15444-1 Profile-1 allows usage of the PPM and PPT marker segments. The PPT marker segment is preferred over the PPM marker segment due to potential “front-loading” of the codestream. In cases where there is only one tile in the image, the PPM marker segment is as efficient as the PPT marker segment. If either marker segment is used, then the packet headers shall not appear in the bitstream data. Furthermore, if the PPM marker segment is used, the PPT marker segment shall not be used. The converse is true as well.

7.6 In Bitstream Marker and Marker Segment

ISO/IEC 15444-1 defines an in bitstream marker segment (SOP) and marker (EPH) that may be used to improve error resiliency when operating in noisy environments. Usage of these marker segments is signaled through the COD marker segment. The SOP marker segment may be placed in front of each packet in the bitstream. The SOP marker segment has a 16 bit ring counter that can be used in the detection of missing or corrupted packets to help determine which packet is missing or corrupted. If the SOP marker segment is used, it need not appear for each packet in the bitstream. The SOP ring counter (Nsop) must be incremented for each packet, however.

Table 7-28 Start of Packet (15444-1 Annex A.8.1)			
Parameter	**Size (bits)**	**Values**	**Notes**
SOP	16	0xFF91	Start of packet marker.
Lsop	16	4	Length of marker segment.
Nsop	16	0 – 65,535	Packet sequence number.

Table 7-28 shows the SOP marker segment. Nsop is incremented for each packet and if it goes beyond 65,535, the count is reset back to 0 and started again. If PPM or PPT marker segments are used to move the packet headers, the SOP marker segment does not move. Instead it is placed in front of the packet body rather than in front of the packet header.

Table 7-29 End of Packet Header (15444-1 Annex A.8.2)			
Parameter	**Size (bits)**	**Values**	**Notes**
EPH	16	0xFF92	End of packet header marker.

Table 7-29 gives the EPH marker. The EPH marker is placed in the bitstream right after the packet header and before the packet body. If the EPH marker is used, it must appear for each packet. If a packet header is corrupted, the EPH marker allows some measure of recovery by delineating the end of the packet header. This prevents the decoding of the packet header from becoming confused and interpreting the packet body as part of the packet header. If PPM or PPT marker segments are used to relocate the packet headers, then the EPH marker segments are moved along with the packet headers.

The use of the SOP and EPH markers are not recommended in general. Their use is encouraged in situations where noisy transmission might corrupt the codestream and there are no other system mechanisms in place to handle bit errors. System designers must fully understand the total system conops before using these markers. Many communication protocols have built-in mechanisms to handle dropped or corrupted data transmission. In these cases, usage of the SOP and EOP markers is unnecessary.

Some overhead price is paid for using SOP and EPH and implementers must consider this in their system design. Trying to recover from a bit error in a JPEG 2000 codestream requires an in-depth understanding of the codestream structures, wavelets and entropy coding. It is reasonable to assume that different codec implementations will perform differently under similar bit error environments. It is also important to realize that where a bit error occurs can have widely varying effects on the decoded image quality. The JPEG 2000 committee has developed more robust error correction and control procedures in ISO/IEC 15444-11:2007, Information technology -- JPEG 2000 image coding system: Wireless. A future version of this profile may include this standard as an option.

7.7 Informational Marker Segments

The informational marker segments are not required for decoding but may assist in the interpretation of the data. Their use is not recommended and BPJ2K01.10 compliant systems are not required to generate or interpret these marker segments. Component registration (CRG) allows each component to be registered to each other for display. The Comment marker (COM) allows for the unstructured data to be included into the file. It is *strictly prohibited* to put information in these marker segments that is necessary to decode the codestream.

Table 7-30 Component Registration (15444-1 Annex A.9.1)			
Parameter	**Size (bits)**	**Values**	**Notes**
CRG	16	0xFF63	Component registration marker.
Lcrg	16	6 – 65,534	Length of marker segment.
$Xcrg^i$	16	0 – 65,535	Value of horizontal offset in units of 1/65536 of the horizontal separation $XRsiz^i$, for the i^{th} component.
$Ycrg^i$	16	0 – 65,535	Value of vertical offset in units of 1/65536 of the vertical separation $YRsiz^i$, for the i^{th} component.

Table 7-31 Comment (15444-1 Annex A.9.2)			
Parameter	**Size (bits)**	**Values**	**Notes**
COM	16	0xFF64	Comment marker.
Lcom	16	5 – 65,535	Length of marker segment.
Rcom	16	0 = General binary 1 = General Latin (IS 8859-15:1999)	Registration values. Indicates type of data in marker segment.
$Ccom^i$	8	0 - 255	Data.

7.8 Low-Low Subband Restrictions

JPEG 2000 Part 1 Profile-1 has a requirement for the LL subband resolution. The restriction is as follows: If one tile is used for whole image, (Xsiz – XOsiz)/D(I) ≤ 128 and (Ysiz – YOsiz)/D(I) ≤ 128, where D(I) = $2^{\text{number of decomposition levels}}$ in SPcod or SPcoc, for I = component 0 to 3. This means the lowest resolution version of an untiled image must be no larger than 128x128 for the first four image components.

8 NSIF with JPEG 2000

Up to this point, all requirements and recommendations have referred only to the JPEG 2000 codestream without mention of the file format in which it is embedded. However, several portions in the NSIF file format contain information directly related to the codestream content. This section shows how the JPEG 2000 codestream fits into the context of the overall file format (see Figure 8-1 below) and provides information about NSIF file header, image subheader and Tagged Record Extension (TRE) settings that are related to the JPEG 2000 codestream content. Note that the guidance in this section not only applies to NSIF files, but to the BIIF and NITF file formats as well when using JPEG 2000.

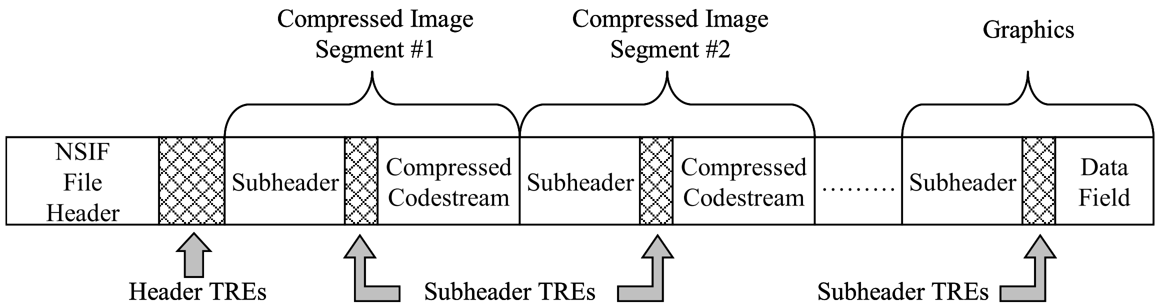

Figure 8-1 NSIF format with embedded JPEG 2000 compressed codestream

8.1 JPEG 2000 File Formats within NSIF

It is recommended that encoders only use the JPC codestream within NSIF. For ease of integrating an image from commercial products several varieties of JPEG 2000 file formats may be placed within the NSIF file format.

The following list defines the current JPEG 2000 standard file formats:

JPC The minimal file that gives only the information required to decode the data,

JP2 The minimal interchange format which includes JPC and other information that improves the display of the image,

(For this profile, only JPC and JP2 formats are allowed for file creation.)

JPX This format includes JP2 information and several XML based metadata fields,

(Under this profile, implementations are only required to decode JP2 compatible JPX files.)

MJ2 This format includes the information for Motion JPEG 2000 and includes all information in JP2.

(MJ2 is not a feature of ISO/IEC 15444 Part-1; support for MJ2 is left for a future version of this profile.)

Implementers need to give due consideration of the format features as they relate to the NSIF format. Information should not be included in JPEG 2000 formats that conflict with information in the NSIF format. Also, these formats should not be used to replace or hide information that should be placed in the NSIF format. For all JPEG 2000 formats, the information within the JPC portions of the bitstream is given precedence, with the exception of information about the length of the compressed image in the NSIF format.

All of the JPEG 2000 standard file formats are based on "boxes." Boxes contain either header information or compressed/image data. If a decoder does not understand a box, it is expected to ignore the given box. It is recommended that only the file format that includes the information that is critical to the NSIF application be used. Using only the minimal file format needed will reduce overhead and redundancy (between the NSIF and JPEG 2000 file formats).

The JPEG 2000 file formats include metadata that is not present in an NSIF file. Specifically, the Enumerated color spaces and ICC profiles enable more accurate color renditions of data. Table 8-1 describes the color interpolation/rendering techniques supported in JPEG 2000 Part 1 and Part 2. NSIF implementations must interpret and apply the color space information.

Table 8-1 JPEG 2000 file format color interpolation/rendering support

Color Encoding	Part 1 JP2	Part 2 JPX Baseline
Color Interpolation/ Rendering	Enumerated Color Space • sRGB, • Grayscale space • sYCC Restricted ICC Profile	JP2 supported values Extended Enumerated • e-sRGB • ROMM-RGB • CIE-Lab • CIE-Jab Any ICC profile

8.2 Population of NSIF Image Subheader Fields When Using JPEG 2000

JPEG 2000 represents not only a new approach to image compression; it requires users and implementers to think of image representation and retrieval in new ways. Not surprisingly, there are some idiosyncrasies that must be dealt with when utilizing JPEG 2000 within the NSIF framework. Some fields within the image subheader of a compressed image segment using JPEG 2000 take on subtly different meanings than they had under other compression technologies like DCT-based JPEG.

The following sections highlight those fields whose interpretation changes under JPEG 2000. The goal of this discussion is to clarify how marker segment parameters in the JPEG 2000 codestream relate to fields in the NSIF image segment subheaders. It is important to remember in the following sections that "JPEG 2000" refers to those compression technologies contained within ISO/IEC 15444-1, Part 1 of the JPEG 2000 standard. It is envisioned that other JPEG 2000 compression technologies, particularly from Part 2 (ISO/IEC 15444-2) of the standard, will eventually be adopted within NSIF. When that time comes, future versions of NSIF will undoubtedly revisit the topic of NSIF header and subheader field population.

8.2.1 Background

JPEG 2000 contains many new technologies; three of these technologies have a profound impact on NSIF. These technologies are the reference grid, tiling, and layering.

8.2.1.1 Reference Grid

The reference grid is a fundamental construct of JPEG 2000. It determines the relative spatial sampling (reference grid sampling factors) between JPEG 2000 components (i.e. image bands) and how these components interact with tiles, precincts, code blocks, and subbands of the wavelet transform. The reference grid provides great flexibility in terms of image tiling, cropping, sampling, and rotation, but this flexibility comes at a cost in complexity. The reference grid allows components with different spatial sampling to be encoded within the same JPEG 2000 codestream. This is a feature that was not available in DCT-based JPEG.

The BIIF standard specifies a coordinate space (reference grid), indexed by row and column counts, that is called the Common Coordinate System (CCS). The CCS coordinate system applies to the NSIF and NITF formats as well (it is best described by the BIIF standard, ISO/IEC 12087-5). The CCS serves as the governing reference grid common to all image segment arrays and graphic segment grids that may be included within a BIIF, NSIF or NITF file. As is the case for uncompressed image arrays, the first sample (pixel) of the first line is the reference point for placing a JPEG 2000 compressed image array in the CCS. In the JPEG 2000 reference grid, the first sample (pixel) is the sample (pixel) pointed to by (XOsiz, YOsiz). The positive x and y directions in the JPEG 2000 reference grid correlate to the positive column and row directions, respectively, in the CCS grid. The row/column index value placed in the image subheader Image Location (ILOC) field identifies the offset in the CCS grid where the first sample (pixel) of the image is to be located. Note that the ILOC row/column value is relative to the position of the BIIF segment (image or graphic) reference point to which the image segment is attached (IALVL>000), or is relative to the CCS origin when the image segment is unattached (IALVL=000).

In their uncompressed state, all the samples (pixels) from all image segments within an NSIF file have a one-to-one spacing/separation correlation with the CCS reference grid. The values in the NROWS and NCOLS fields of the image subheader define the dimensions of the array in the CCS. The same one-to-one relationship is true for graphic segment reference grids; each unit of the graphic grid index correlates to one unit of the CCS grid index. The reference point for positioning graphics in the CCS is the origin of the graphic's reference grid. The absolute positional relationship of image and graphic segments placed in the CCS must be preserved when displaying or otherwise portraying an NSIF composition consisting of multiple image and/or graphic segments. Due consideration must be taken to preserve the display context defined by the CCS when performing visualization manipulations (e.g. zooming, roaming, rotating, etc.) of an NSIF composition, especially one containing multiple image and/or graphic segments.

8.2.1.2 Tiling

It is recommended that the image be tiled within a JPEG 2000 codestream. The tiles in JPEG 2000, similar to the NSIF blocks, can be encoded or decoded, displayed and manipulated independently. Each tile is transformed and coded separately, but the encoded tiles all appear within the same JPEG 2000 codestream. This is a fundamental difference between JPEG 2000 tiling and the current blocking (tiling) within NSIF where each tile is an independent JPEG encoded image. This difference affects how certain NSIF image subheader fields must be interpreted (see Table 8-2).

8.2.1.3 Progression Order and Layering

Progression order and layering are two new concepts that give JPEG 2000 great flexibility in how a codestream can be ordered and stored. JPEG 2000 relies upon embedded coding to progressively refine the resolution and quality of an image as more of a compressed file is decoded. Layering is a fundamental aspect of the embedded coding in JPEG 2000. There are five different layering progressions that can be used within JPEG 2000 that determine the order in which compressed data is stored and/or transmitted.

The impact that layering has upon NSIF is in band ordering. Within a JPEG 2000 codestream, the compressed data for a given component in a tile, or image for the one tile case, may be scattered throughout the codestream. Thus the data for a particular component is not contiguous within the codestream and notions of band ordering are of little relevance. The reason behind this behavior is that if it is desired to receive or send a color image, for example, at a reduced quality or resolution without significant reordering of the codestream, you must incorporate information from all components at all resolutions and all quality layers throughout the entire codestream. If each component were stored contiguously within the codestream a server would have to seek to different parts of the codestream to assemble the necessary information to transmit. While JPEG 2000 does not completely remove the necessity of a server parsing a codestream, it helps alleviate the amount of work a server must perform.

8.2.1.4 Sub-sampling on the Reference Grid

The JPEG 2000 reference grid provides a means for sub-sampling of image components relative to the reference grid. Wavelet subbands are mapped to the reference grid at different resolutions, as are other constructs (e.g. codeblocks). The component column and row sub-sampling (separation) factors are XRsiz and YRsiz. For various circumstances (such as when none of the components has (XRsiz, YRsiz) set to (1,1)), there is potential for ambiguity regarding the intended dimensions for the reconstituted image array. When decoding and displaying such a codestream, the NROWS and NCOLS values in the NSIF Image Subheader define the dimensions (image size) of the image array to be portrayed in the NSIF CCS.

It is recommended that at least one image component within an image not be separated or subsampled on the JPEG 2000 reference grid during the original full-resolution compression of image arrays (i.e. at least one component of the image should have (XRsiz, YRsiz) set to (1,1)). When, because of collection system characteristics, repackaging of an existing codestream, parsing a reduced resolution from an existing codestream, or other function wherein samples need to be separated, the component sub-sampling (separation) factors, XRsiz and YRsiz, are to be used to indicate the amount of separation.

NOTE: Proper use of image support data (contained in NSIF TRE) is often dependent on maintaining rigorous correlation of the column (X) and row (Y) indices of the original collected image array associated with the coverage parameters expressed in the support data. Passing the correct column (X) and row (Y) indices of a pixel (sample) within the domain of the support data to an exploitation tool is critical to obtaining trusted results from the tool.

8.2.1.5 Reduced Resolutions

When a reduced resolution is parsed from a JPEG 2000 compressed image, it is common practice to use the sub-sampling factors, XRsiz and YRsiz, to separate the samples on the reference grid while retaining the original image array size factor values, Xsiz and Ysiz. The NSIF image subheader NCOLS, NROWS, and IMAG fields play a critical role in maintaining rigorous correlation of the column (X) and row (Y) indices of the original collected image array associated with the coverage parameters expressed in the support data. The value in the IMAG field must represent the reduction factor of the NCOLS and NROWS values in the reconstituted reduced resolution image as compared to the NCOLS and NROWS values of the original full resolution image array upon which the support data (TRE) is based.

Reduced resolution images decoded or parsed from a JPEG 2000 codestream can be used to form a traditional Reduced Resolution Data Set (RRDS) for use by electronic light table (ELT) software packages. For the interested reader, additional guidance concerning the proper softcopy display of RRDS imagery produced from a JPEG 2000 codestream can be found in the Softcopy Image Processing Standard version 2.0 (SIPS 2.0) and its appendices, as appropriate. This standard is available from the U.S. National Geospatial-Intelligence Agency's (NGA) Image Quality & Utility (NIQU) office.

8.2.1.6 Asymmetric Sample Size

Some image sensors use sampling rates that are different in the X and Y sampling directions for a given component (typically a single component collection). This results in asymmetric sample sizes. Proper correlation of image support data (contained in NSIF TRE) with x and y (column and row) indices must be considered when packaging this data as JPEG 2000 compressed within the NSIF file format. The TRE associated with these types of sensors have a field value that indicates whether the samples have, or have not, been corrected for asymmetry (anamorphic correction flag). The component column and row sub-sampling (separation) factors (XRsiz and YRsiz) can potentially be used to address asymmetric sample values; however, if used, it must be done in consideration of the image support data flag for asymmetric/anamorphic correction.

For example, consider a sensor that obtains samples in the X (column) direction twice as frequently as samples taken in the Y (row) direction. An image of 50 rows would have 100 samples in each row.

For the uncompressed and uncorrected case:

NCOLS=100, NROWS=50, and the asymmetric/anamorphic correction flag=no.

For the uncompressed, but corrected case (re-sampled in row direction):

NCOLS=100, NROWS=100, and the asymmetric/anamorphic correction flag=yes.

Now consider the above example when JPEG 2000 compressed.

For the uncorrected case:

NCOLS=100, NROWS=50, (XRsiz, YRsiz) = (1,1), and the asymmetric/ anamorphic correction flag=no.

For the corrected case, there are two options:

1) Re-sample (up-sample) in the row direction prior to compression resulting in: NCOLS=100, NROWS=100, (XRsiz, YRsiz) = (1,1), and the asymmetric/ anamorphic correction flag=yes.

2) Make use of the JPEG 2000 sub-sampling/separation factors instead of up-sampling prior to compression:
NCOLS=100, NROWS=100, (XRsiz, YRsiz) = (1,2), and the asymmetric/ anamorphic correction flag=yes.

For the interested reader, the SIPS 2.0 standard, available from NIQU (see above), provides guidance for the production and softcopy display of RRDS imagery generated from JPEG 2000 codestreams containing asymmetric sample sizes.

8.2.2 NSIF Image Subheader Fields

The image subheader fields potentially affected when applying JPEG 2000 compression to the image data are listed in Table 8-2. The table includes directions for populating each of the affected fields.

Table 8-2 NSIF Field JPEG 2000 Guideline

NSIF Field Name	JPEG 2000 Based Value	Notes
NROWS NCOLS N/8 Range: 00000001 – 99999999	NROWS and NCOLS are set to the desired size of the image plane that will be ready for display. For the original image, $\text{NROWS} = Ysiz - YOsiz$ $\text{NCOLS} = Xsiz - XOsiz$ For extraction and repackaging of reduced resolutions (i.e., R1) the NROWS and NCOLS should be set as follows, $\text{NROWS} = \left\lceil \frac{Ysiz}{IMAG\_NEW} \right\rceil - \left\lceil \frac{YOsiz}{IMAG\_NEW} \right\rceil$ $\text{NCOLS} = \left\lceil \frac{Xsiz}{IMAG\_NEW} \right\rceil - \left\lceil \frac{XOsiz}{IMAG\_NEW} \right\rceil$ For non-symmetrical pixels (either defined by metadata or $XRsiz^i$ or $YRsiz^i$) the display should be adjusted to make pixels square. (Refer to section 8.2.1.4 of this document.)	Number of Rows/Columns. For images that start with IMAG not equal to 1, then the NROWS and NCOLS should be set to the desired size of the image plane that will be ready for display.

Table 8-2 NSIF Field JPEG 2000 Guideline

NSIF Field Name	JPEG 2000 Based Value	Notes
PVTYPE A/3 Range: B INT SI R C	$max(Ssiz^i) = 0 \Rightarrow$ PVTYPE = B $1 \leq max(Ssiz^i) \leq 37 \Rightarrow$ PVTYPE = INT $128 \leq max(Ssiz^i) \leq 165 \Rightarrow$ PVTYPE = SI If the MSB of $Ssiz^i$ is 1 then the component is signed. If the codestream contains both signed and unsigned data then the PVTYPE = SI. JPEG 2000, Part1, Profile-1 supports neither Real (R) nor Complex (C) pixel value types.	Pixel Value Type. Determined for entire image by examination of the values of $Ssiz^i$ for each component. JPEG 2000 components can be from 1 to 38 bits deep. The value indicated by the lower 7 bits of $Ssiz^i$ is one less than the component bitdepth.
IREP A/8 Range: MONO RGB RGB/LUT MULTI NODISPLY NVECTOR POLAR VPH YCbCr601	IREP follows whatever would be appropriate for the uncompressed image. SGcod = xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxx1 indicates internal JPEG 2000 three component transform Note: IREP=YCbCr601 only if the color transform is used external to JPEG 2000. If the color transform is used internal to JPEG 2000, then use the value that represents the decoded data (i.e., IREP = MONO for single band imagery, IREP = RGB for three band imagery and IREP = MULTI for multi-band imagery). Several color interpolation methods are supported in the JP2 and JPX file formats. These techniques are described in ISO/IEC 15444-1 and ISO/IEC 15444-2.	Image Representation. The three-band component transform in JPEG 2000 is the RGB to YCbCr transform for the ICT and a reversible approximation of the same transform for the RCT. This transform is used solely to aid compression. Although the transform may be used with any three component image, it is most effective in aiding compression with RGB imagery.
ABPP N/2 Range: 01 – 96	ABPP = NBPP	Actual Bits Per Pixel.
IC A/2 Range: NC, NM C1, M1 C3, M3 C4, M4 C5, M5 C6, M6 C7, M7 C8, M8 I1	C8, M8 Mask tables may be used with JPEG 2000 compression. See section 8.5 for details.	Image Compression. Flag to signal the compression algorithm.

Table 8-2 NSIF Field JPEG 2000 Guideline		
NSIF Field Name	**JPEG 2000 Based Value**	**Notes**
COMRAT A/4 Range: 1D 2DS 2DH XX.Y nn.n	Set to the approximate number of bits-per-pixel-per-band (bpppb) for the compressed image. The truncation point value for R0 (full resolution). Ideally this value is computed based on compressed image codestream length. Can also be computed as an expected value based on the compression statistics of other resolution levels (R1, R2, etc.) within the codestream or from compression statistics derived from similar images encoded in a similar fashion. Nxyz = JPEG 2000 numerically lossless, where "xyz" indicates the expected achieved bitrate (in bpppb) for the final layer of each tile. The decimal point is implicit and assumed to be one digit from the right (i.e. xy.z). Vxyz = JPEG 2000 visually lossless, where "xyz" is the target or expected bitrate (in bpppb) for the final layer of each tile. The decimal point is implicit and assumed to be one digit from the right (i.e. xy.z). wxyz = JPEG 2000 lossy, where "wxyz" is the target or expected bitrate (in bpppb) for the final layer of each tile. Note: When there is no decimal point, the decimal point is implicit and assumed to be in the middle (i.e. wx.yz). “Numerically lossless” means that the image if fully decoded would have identical content to the original uncompressed data, for every pixel present in the compressed file. This property holds for spatially chipped image regions. Therefore a spatial chip may be labeled as Nxyz, but reduced quality and reduced resolution images cannot.	Compression Rate. Used to designate the degree of compression and consequently the quality of the expanded image as compared to the image prior to compression. It is not safe make quality comparisons based upon reduced resolution COMRAT values computed via different techniques. Instead, use information from the J2KLRA TRE to make any distinctions of this kind.

Table 8-2 NSIF Field JPEG 2000 Guideline		
NSIF Field Name	**JPEG 2000 Based Value**	**Notes**
COMRAT A/4 Range: 1D 2DS 2DH XX.Y nn.n	"Visually lossless" means that the image if fully decoded would present a display that is visually indistinguishable from the original uncompressed data over the same spatial area and subset of bands. If image resolution is reduced then the data is no longer visually lossless. Spatially chipped or band extracted data, however, could remain Vxyz. If the reader needs information about the original compression type (Nxyz, Vxyz, or wxyz) for a file with COMRAT of wxyz, the J2KLRA and HISTOA TREs may be helpful. In addition, certain codestream compression parameters (see COD and COC marker segments) in the codestream can help identify the original compression type and quality.	
NBANDS N/1 Range: 0, 1-9 XBANDS N/5 Range: 00010 – 99999	$Csiz \leq 9 \Rightarrow$ NBANDS = $Csiz$ $Csiz \geq 10 \Rightarrow$ NBANDS = 0, XBANDS = $Csiz$	Number of Bands. Number of components in the image segment.
NLUTS N/1 Range: 0 – 4 NELUTn N/5 Range: 00001 – 65536 LUTDnm Derived from previous two values.	If using an external format palletized LUT(s) (i.e., JP2 format), then it must be transferred to the NSIF LUT header.	Look Up Tables. Look-up tables provide a means, on a per-band basis, to substitute pixel value expressions. For example, assign 24-bit color values for up to 256 pixel values available in an 8-bit-per-pixel, palletized, pseudo-color image.
IMODE A/1 Range: B P R S	IMODE = B IMODE B (interleave by block (tile)) is the most appropriate selection of the available choices for JPEG 2000.	Image Mode. Type of data interleaving.

Table 8-2 NSIF Field JPEG 2000 Guideline

NSIF Field Name	JPEG 2000 Based Value	Notes
NBPR NBPC N/4 Range: 0001 – 9999	$\text{NBPR} = \left\lceil \frac{Xsiz - XTOsiz}{XTsiz} \right\rceil$ $\text{NBPC} = \left\lceil \frac{Ysiz - YTOsiz}{YTsiz} \right\rceil$	Number of Blocks Per Row/Column. Computes the blocking (tiling) geometry of the pixel array. The formula takes into consideration tile offsets on the reference grid (XTOsiz, YTOsiz).
NPPBH NPPBV N/4 Range: 0000, 0001 – 8192	If NBPR = 1 and the NPPBH equation below is greater than 8192 then NPPBH value should be set to = 0. NCOLS shall contain the actual size of the block and image in the horizontal direction. If NBPC = 1 and the NPPBV is greater than 8192 then NPPBV value should be set to = 0. NROWS shall contain the actual size of the block and image in the vertical direction. $\text{NPPBH} = \left\lfloor \frac{XTsiz}{IMAG\_NEW} \right\rfloor$ $\text{NPPBV} = \left\lfloor \frac{YTsiz}{IMAG\_NEW} \right\rfloor$	Number of Pixels Per Block Horizontally and Vertically. Computes the maximal number of image samples in the row and column dimension within a tile. Note that border tiles might have fewer samples.
NBPP N/2 Range: 01 - 96	$\text{NBPP} = \max_i \left(Ssiz^i \ \& \ 0x7F \right) + 1$ This field contains the largest bitdepth in the JPEG 2000 SIZ marker ($Ssiz^i$), accounting for signed and unsigned data types.	Number of Bits Per Pixel. Determines the maximum component bitdepth (max over all components i). "$Ssiz^i$ & 0x07" is a bitwise "AND" function that strips off the lower 7 bits of $Ssiz^i$ where the component bitdepth is stored (the MSB of $Ssiz^i$ indicates signed or unsigned component samples). The "+ 1" is needed since the stored number in $Ssiz^i$ is the bitdepth – 1.

Table 8-2 NSIF Field JPEG 2000 Guideline

NSIF Field Name	JPEG 2000 Based Value	Notes
ILOC N/10 Range: (-rrrr,-cccc) - (rrrrr,ccccc)	The relative offset location in the CCS relative to the segment it is attached to (symbol, pixel, ...) i.e., the CCS location of the pixel pointed to by (XOsiz, YOsiz).	Image Location. Identifies the relative offset location in the Common Coordinate System (CCS) of the first pixel of the first line of the image.
IMAG A/4 Range: .nnn n.nn nn.n nnnn /nnn	Populate this field with the value of the highest resolution available in the compressed image data relative to the original source image. If the JPEG 2000 codestream is repackaged to generate a reduced resolution, then the resolution reduction must be indicated here. IMAG_NEW = IMAG_OLD/reduction factor. For example: a R1 repackaging would have IMAG = IMAG_NEW = /2	Image Magnification. Contains the approximate image magnification (or reduction) factor of the image data relative to the original (as collected) source image.

8.3 Recommended J2KLRA TRE

The J2KLRA TRE was developed to allow providers and users of NPJE and EPJE compressed data to quickly access the compressed data, but is available to be used by other encodings. The TRE provides users information about number of resolution levels, number of quality layers, and number of bands in both the original data and derived products. This information might be critical in the selection and ordering of data from a library. The J2KLRA TRE is recommended to be included with any original compressed data and compliant derived compressed products (i.e., parsing and repackaging).

Table 8-3 Recommended J2KLRA TRE

Field	Name/description	Size bytes and format	Req. or Con.	Value Range
CETAG	Unique Extension Type Identifier Unique TRE identifier.	6, BCS-A	R	J2KLRA
CEL	Length of User-Defined Data Length in bytes of data contained in subsequent TRE fields. (TRE length is 11 plus	5, BCS-N	R	Variable Calculated for each specific TRE.

Table 8-3 Recommended J2KLRA TRE				
Field	**Name/description**	**Size bytes and format**	**Req. or Con.**	**Value Range**
	the value given in the CEL field).			
ORIG	Original compressed data Indicates if the image is in the same original JPEG 2000 compression or it has been parsed to a new JPEG 2000 compression. The conditional fields (NLEVELS_I, NLAYERS_I, NBANDS_I) are present if this field indicates a parsed stream.	1, BCS-N	R	0 – Original NPJE 1 – Parsed NPJE 2 – Original EPJE 3 – Parsed EPJE 4 – Original TPJE 5 – Parsed TPJE 6 – Original LPJE 7 – Parsed LPJE 8 – Original other 9 – Parsed other
Original compressed image information (the first JPEG 2000 Compression)				
NLEVELS_O	Number of wavelet levels in original image Indicates the number of wavelet decompositions levels performed in the original image.	2, BCS-N	R	00 – 32
NBANDS_O	Number of bands in original image Indicates the number of bands in original image.	5, BCS-N	R	00001 – 16384
NLAYERS_O	Number of layers in original image Indicates the number of bands in original image.	3, BCS-N	R	001 – 999
Layer information (This is the start of a repeating section for n = 0 to NLAYERS_O – 1).				
$LAYER\_ID_n$	Layer ID Number Indicates the number of layer being described. Layers are numbered from 0 to NLAYERS_O – 1. 0 is the layer with the lowest bitrate.	3, BCS-N	R	000 – 998
$BITRATE_n$	Bitrate Indicates the accumulated bitrate target associated with this and associated lower layers. This is defined in bits per pixel per band. It may happen that the bitrate was not achieved due to data characteristics. Note for JPEG 2000 numerically lossless quality, the bitrate for the final	9, BCS-A	R	Value 00.000000 – 37.000000*

Table 8-3 Recommended J2KLRA TRE				
Field	**Name/description**	**Size bytes and format**	**Req. or Con.**	**Value Range**
	layer is an expected value based on past performance. If there is not a target bitrate, report the achieved bitrate.			
(This is the end of a repeating section).				
Conditional field if the data has been parsed				
NLEVELS_I	Number of wavelet levels in this image Indicates the number of wavelet decompositions levels included in this image as defined in the COD marker segment.	2, BCS-N	C	00 – 32
NBANDS_I	Number of bands in this image Indicates the number of bands in this image as defined in the SIZ marker segment.	5, BCS-N	C	00001 – 16384
NLAYERS_I	Number of layers in this image Indicates the number of layers in this image as defined in the COD marker segment.	3, BCS-N	C	001 – 999
* The component sample precision is limited by the number of guard bits, quantization, growth of coefficients at each decomposition level, and the number of coding passes that can be signaled. Not all combinations of coding styles will allow the coding of 37 bit samples per band.				

During primary compression, all TRE fields, except for the conditional Nxxxx_I fields, are populated. When an image is repackaged, the CEL and ORIG fields are updated and the NLEVELS_I, NBANDS_I, and NLAYERS_I fields are added or replaced (if they already exist).

8.4 Image Segments When Using JPEG 2000

A JPEG 2000 codestream can contain at most 65,535 tiles, so every image segment containing JPEG 2000 compressed data will contain at most 65,535 tiles. Typically the file format restrictions on image segment length or ILOC offset will be invoked before this number of tiles is reached. However, when the bitrate is very low, the image segment length or ILOC offset limitation will not be exceeded. It is then possible that an image segment will need to be terminated due to the number of tiles.

When multiple NSIF image segments are created from a single image, it is strongly recommended that the same image compression parameters (quantization, layering, wavelet transform filter and levels, progression order, coding defaults) be used across segments. This ensures that the individual JPEG 2000 codestreams used to represent the multi-segment image plane all have the same main header QCD/QCC and COD/COC marker segments. This practice facilitates NSIF image file spatial chipping operations that happen to cross over the boundaries between individual image segments contained in the same file.

Certain collection systems using the NSIF file format may choose to provide multi-segment imagery as a series of individual NSIF-formatted single image segment files. These individual files are then "linked" via support data so as to provide the end-user information about the full multi-segment image scene extent. In these cases, it is also strongly recommended that the same image compression parameters (quantization, layering, wavelet transform filter, levels, progression order, and coding defaults) be used across image files containing segments of the same overall image plane. Again, the reason for this recommendation is the facilitation of spatial chipping and/or re-segmentation of the image plane when crossing boundaries between the individual segment codestreams of these separate NSIF files.

8.5 Image Data Mask When Using JPEG 2000

Use of NSIF image data mask tables with JPEG 2000 compressed data may be beneficial to the user community in providing decreased file size and processing time. This clause addresses the use of image Block Mask Records (BMR), Pad Pixel Mask Records (TMR), and the Output Pad Pixel Code (TPXCD) with JPEG 2000 compressed data. The use of Transparent Mask Records (TMR) is not currently defined for use with JPEG 2000, and therefore, shall not be used.

8.5.1 Block Mask Records When Using JPEG 2000.

The nature of the JPEG 2000 codestream warrants a modified approach for use of block mask tables as compared to their use with uncompressed data. For JPEG 2000, the BMRnBNDm records are not used to indicate the offset to the block (JPEG 2000 tile) since JPEG partitioning and interleave schemes prevent useful application of this approach. The approach for JPEG 2000 takes advantage of the compressed data structure that internally accounts for all tiles (NSIF blocks) present, empty or missing in the codestream. Since the BMRnBNDm records are not needed to preserve raster order or provide random access, they shall be used to designate the treatment of the blocks (tiles) for presentation. Blocks (tiles), whether present, empty or missing, may be designated for treatment as either "transparent," or to be decoded and processed for presentation as would be done if no block mask records were present. The record value of 0x00000000 shall designate that the block (tile) is to be decoded and processed for presentation. The record value of 0xFFFFFFFF shall designate the block (tile) is to be presented as "transparent".

8.5.2 Pad Pixel Mask Records When Using JPEG 2000.

The nature of the JPEG 2000 codestream warrants a modified approach for use of pad pixel mask tables as compared to their use with uncompressed data. For JPEG 2000, the TMRnBNDm records are not used to indicate the offset to the block (JPEG 2000 tile) since JPEG partitioning and interleave schemes prevent useful application of this approach. The TMRnBNDm records shall be used to designate the treatment of the blocks (tiles) for presentation. Blocks (tiles) may be designated as containing the designated Pad Output Pixel Code (TPXCD), or not. The record value of 0x00000000 shall designate that the block (tile) is present and contains pad pixels. The record value of 0xFFFFFFFF shall designate the block (tile), whether present, empty or missing, does not contain pad pixels.

The output pixel code which represents pad pixels is identified within the image data mask by the Pad Output Pixel Code (TPXCD) field. When an interpret application identifies pad pixel values, it may replace them with a user defined value at the time of presentation except when the value of TPXCD is zero (0x00). When the TPXCD value is zero (0x00), the pad pixel will be treated as “Transparent” for presentation.

The ability to preserve exact pixel values designated for special handling as pad pixels through a compression and decompression process is viable only when the process is numerically (bit-for-bit) lossless. Therefore designation of pad pixels and inclusion of pad pixel mask records when generating a JPEG 2000 compressed image segment is constrained to the lossless case.

Appendix A JPEG 2000 processing

The following sections describe the procedures that are required to utilize the functionality of JPEG 2000. The main processes are defined according to the recommended compressed data stream. The processes include compression, parsing, decompression, enhancing, and repackaging. To achieve a given functionality, several processes are required to be strung together. For example, if a library would like to chip data, the data must be parsed and then repackaged to supply the user with a final product.

A.1 End-to-End Overview

There are three main parts of any image collection and distribution system. The first part of the end-to-end system is the collection system, which is the originator of the image. The collection system will send the collected image to the second part, a distributor or library, so that it is available to the users. The distributor commonly has the ability to change attributes (i.e., file format, compression type and ratio, image size, image resolution) of the image to best support the user. The main goal of the first two parts is to support the third part or user of the data. The data is exploited and turned into information and action by the user.

A.1.1 Current End-to-End CONOPS

Currently, most end-to-end systems use multiple compression algorithms to meet all of the requirements of the users. The compression usually consists of lossless compression, high quality (visually lossless) lossy compression, medium quality lossy compression, and high compression for bandwidth constrained users. Most of the processing that is performed for distribution, transmission, and display is completed in the image space and not in the compressed space. Therefore, the first process that is performed on any compressed data is to decompress and the last process when transmitting or storing is to recompress. This aspect of the current architecture results in concatenation of multiple compressions being performed on a given image. In Figure A-1 processing data would include RRDS generation, spatial chipping, changing quality/compression ratio, and thumbnail production.

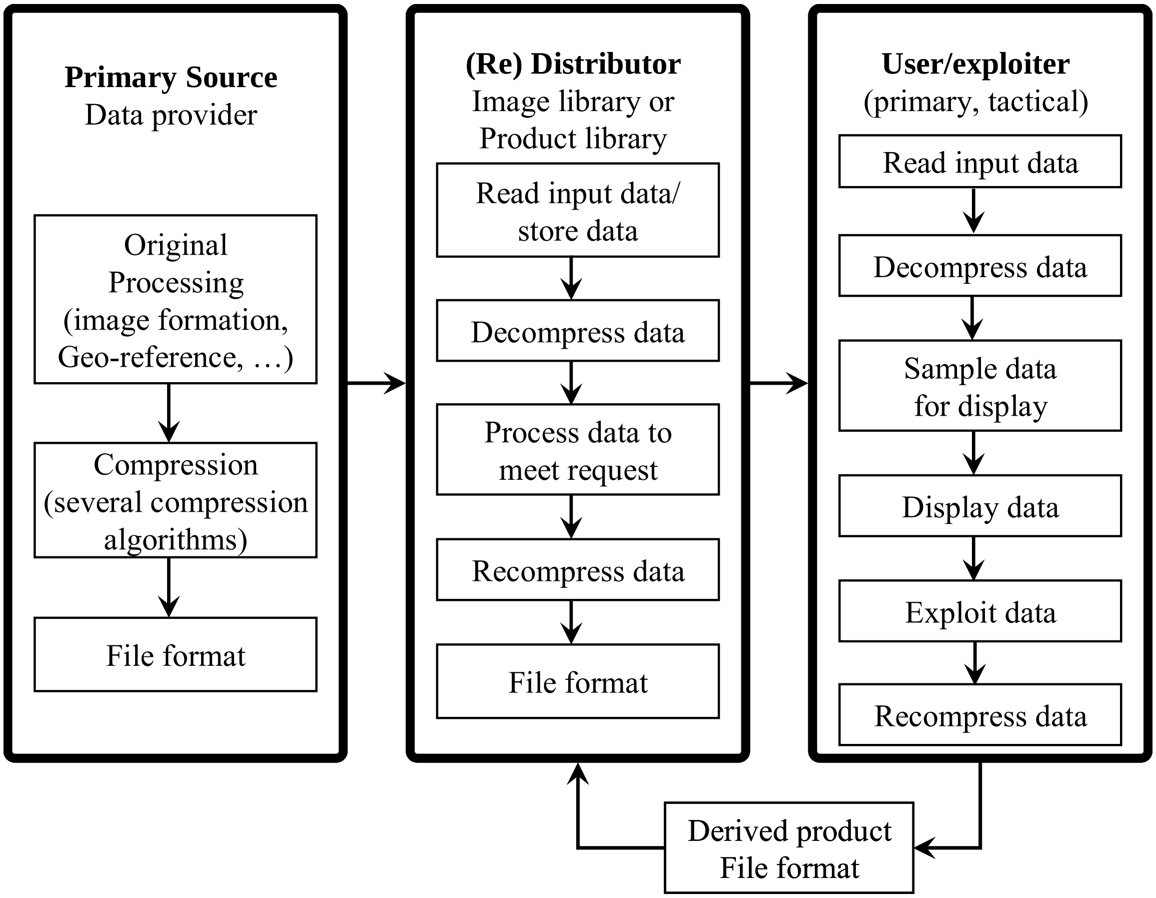

Figure A-1 Current general end-to-end CONOPS

A.1.2 JPEG 2000 End-to-End CONOPS

In the JPEG 2000 CONOPS, JPEG 2000 can meet all of the requirements for the collection, distributor, and user, so it is the only compression algorithm used. There is no concatenation of compression algorithms. Also, since much of the processing can be completed in the compressed space, there is no need to decompress the data. Expansion should only be performed immediately prior to exploiting or viewing JPEG 2000 data. Otherwise data should remain compressed, with repackaging being used to satisfy user requirements whenever possible. This CONOPS should significantly reduce the complexity of the distributors' process flow. Figure A-2 includes the parsing of the data to achieve spatial chipping, RRDS generation, thumbnail generation, and changing quality/compression ratio.

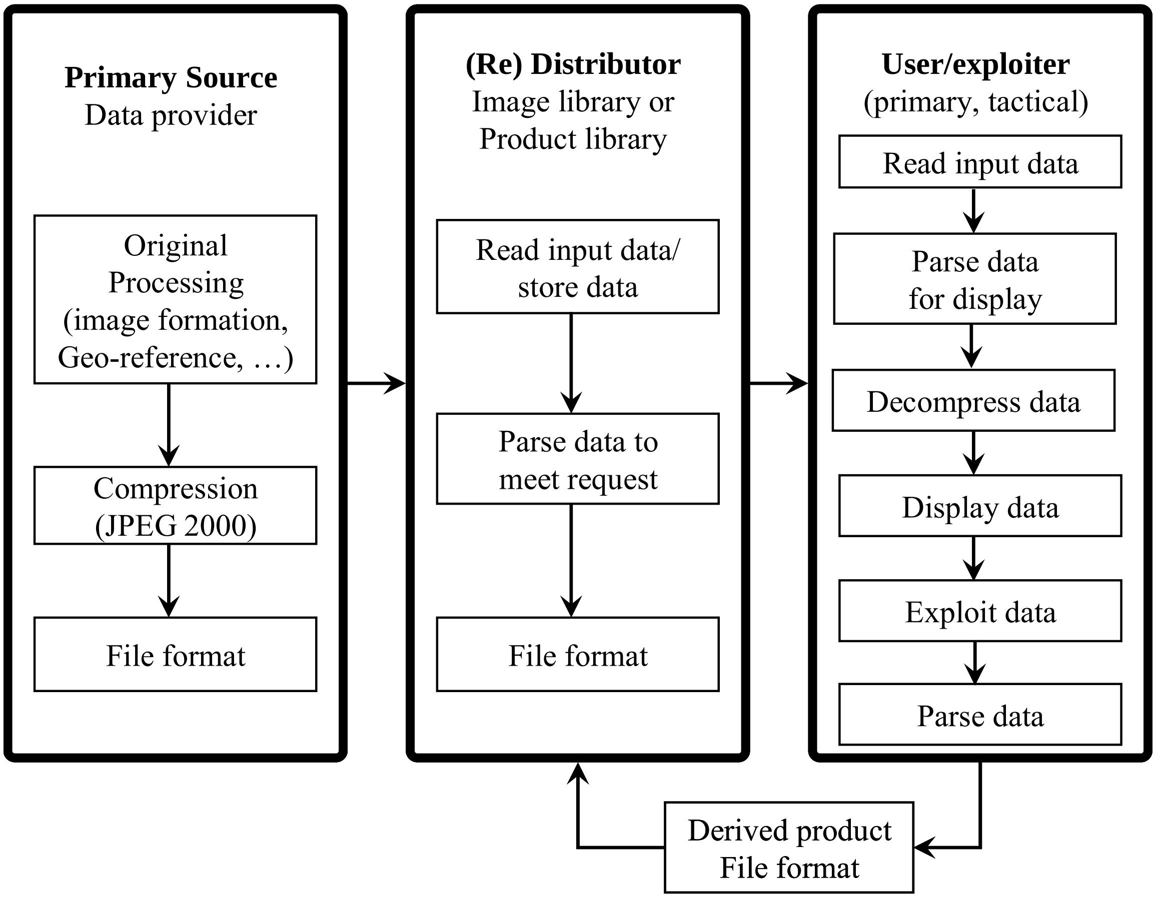

Figure A-2 JPEG 2000 end-to-end CONOPS

A.2 JPEG 2000 Encoder Overview

Figure A-3 illustrates the high-level processing chain of a JPEG 2000 Part 1 encoder. Portions of the processing chain enclosed by dashed boxes represent optional or conditional elements.

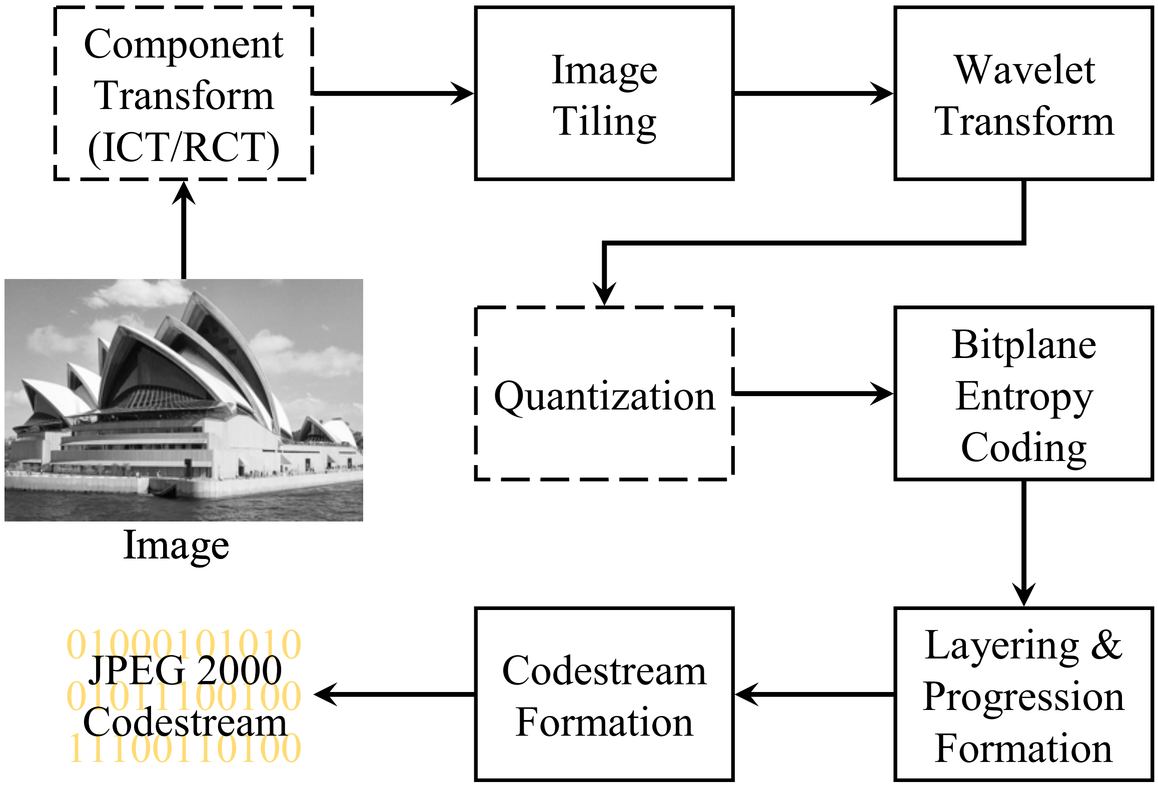

Figure A-3 JPEG 2000 encoder block diagram

A.2.1 Component Transform

The JPEG 2000 standard (ISO/IEC 15444-1) specifies two component transforms that may be used with three-band imagery to improve compression performance. The Reversible Component Transform (RCT) may be used with the 5-3R integer reversible wavelet transformation and the Irreversible Component Transform (ICT) may be used with the 9-7I irreversible wavelet transformation. The component transforms are applied prior to the wavelet transform.

The ICT/RCT transforms approximate the RGB to YCbCr601 transform. If the colorspace of the input image is not RGB or something that closely approximates it, usage of the component transform is not appropriate. The ICT is the 601 transform (see MIL-STD 188-198A and ISO/IEC 10918-1) and the RCT is an integer reversible approximation of the transform. Since both transforms are point transforms in the component dimension, the sequence of processing operations may be swapped with image tiling as long as no external sub-sampling of chrominance bands is performed. With the use of the wavelet transform, chrominance sub-sampling is unnecessary and is not recommended.

A.2.2 Image Tiling

Input images to the JPEG 2000 encoder may be spatially tiled (see ISO/IEC 15444-1, Annex B). Tiling is performed to ease memory resources needed for compression and expansion. Tiles can also be used to improve random access to spatial areas in large images. In terms of the wavelet transform, entropy coding, layer and progression formation, each tile is treated independently.

Non-tiled images are considered to consist of a single tile encompassing the image. For a given image, all tiles have the same nominal size. Tiles may be rectangular in dimension and spatially offset relative to the top left corner of the image. The image need not be an integer number of tile widths and tile heights in size. This leaves open the possibility that the tiles on the image border may be of reduced size in the vertical or horizontal dimension and possibly both. These incomplete tiles are not padded, but rather they are clipped to the image border.

A.2.3 Wavelet Transform

Two different wavelet transforms are described in ISO/IEC 15444-1, Annex F. One is a 5-3 (5-3R) integer reversible wavelet transform that allows for lossless encoding/decoding whenever all bits in a compressed file are received. The other wavelet transform is an irreversible 9-7 (9-7I) transform. While the inputs and outputs of the 5-3R wavelet transform are integers, the inputs of the 9-7I transform are integers or real numbers and the outputs are real numbers.

Both wavelet transforms may be implemented using the lifting technique described in the JPEG 2000 standard. Normalization of the analysis and synthesis wavelet filters is described in ISO/IEC 15444-1 JPEG 2000 standard, however, any conforming filtering and normalization methods can be used. The wavelet transform in two dimensions is implemented as separable one-dimensional transforms in the row and column directions.

The wavelet transform used in JPEG 2000 is hierarchical in nature. It forms a multi-resolution version of the image similar to reduced resolution datasets (RRDS). Since the wavelet transform is multi-resolution in nature, JPEG 2000 alleviates the need for R-set generation.

A.2.4 Quantization

Quantization is used with the 9-7I wavelet transform to map its real-valued output back into integers. During quantization, the floating-point wavelet coefficients are divided by a number and the results are rounded to integers. These integers are quantization indices and they represent an approximation to the true wavelet coefficients. It is these quantization indices that are entropy coded into the output codestream. The process of quantization reduces fidelity in the reconstructed image.

Quantization step sizes can be adjusted on a subband-by-subband basis to match image statistics and maximize reconstructed image fidelity. The JPEG 2000 standard does not discuss how to do this, but [Taubman & Marcellin] provides a good place to start. In [J2K_Guide], a "base" quantization step size technique is described that provides very good performance in a rate-distortion sense.

The 5-3R wavelet transform does not use explicit quantization as described above. Instead, it uses "embedded" quantization. Embedded quantization and embedded coding is perhaps the most powerful feature of JPEG 2000. It enables progressive transmission and file truncation, and allows great flexibility in the parsing and repackaging of codestreams (see section A.3). Rather than explicitly quantize the wavelet coefficients, you can simply choose to not send all of the bits in the binary representation of the coefficients.

Embedded quantization is effectively the same as quantization by powers of two. Rather than divide a wavelet coefficient by a number to reduce precision, you simply throw away bits in the binary representation of the coefficient. Embedded and explicit quantization may be combined, and in fact they are for the 9-7I wavelet transform. After explicit quantization of the floating point 9-7I wavelet coefficients, the resulting integer representation may undergo embedded quantization.

A.2.5 Bitplane Entropy Coding

After quantization, the wavelet coefficients are entropy coded a bitplane at a time. JPEG 2000 uses an adaptive arithmetic entropy coder which is much more efficient in terms of rate-distortion performance than the Huffman coder used in JPEG, [ISO/IEC 10918-1]. The improved rate-distortion performance comes at a higher computational cost. The bitplane coding of the wavelet coefficients enables embedded coding in JPEG 2000.

The JPEG 2000 arithmetic coder uses local information in the wavelet domain from adjacent wavelet coefficients to help it statistically predict what the next bit for the current wavelet coefficient will be. Furthermore, the JPEG 2000 arithmetic coder is adaptive. It modifies its wavelet coefficient bitplane probability models based on the values it has seen so far. This implies that there is a causal procedure that must be observed to successfully decode the data.

To help offset the number of coefficients that must be decoded to decode a small region in an image, the wavelet coefficients within each wavelet subband are grouped into smaller units called code blocks. The wavelet coefficients within each code block are coded independently by the arithmetic coder. Therefore, you need only decode wavelet coefficients within the code blocks that spatially correspond with a region of interest.

A.2.6 Layering and Progression Formation

After the wavelet coefficient bitplanes have been entropy coded, they are placed into a codestream. The data may be arranged in many different ways in order to achieve different functionality. For example you may place the data in the file such that if you truncate the file and decode the resulting image, you will always have the "best possible" full resolution reconstructed image. If you define "best possible" to mean "maximize the signal-to-noise ratio" then you have what is typically called an "SNR-progression" or "progression in quality".

Conversely you might wish to organize the compressed data in the codestream so that you always receive the best image quality as you go from low-resolution to high-resolution (i.e. high R-set number to low R-set number). Thus when you truncate the file you will have the best possible R5, for example. If you decode more of the file you will then have the best R4, etc. This is an example of "progression in resolution." JPEG 2000 has several different progression types that can be used to order the final codestream.

Furthermore, within a given progression, it is advantageous to "layer" the coded data. A layer represents an improvement in quality. It is a collection of some number of bitplanes from each code block in each wavelet subband. To keep track of where the various bitplanes are located throughout the compressed codestream, a mechanism known as packet headers is used. A packet is simply the collection of contributions from the code blocks in the wavelet subbands of the current resolution level for the current layer (Note: the HL, LH, and HH subband contributions for the current resolution level are collected together in a packet, the LL subband is in its own packet).

To understand the need for layers, it's best to consider the progression in quality case. From an SNR perspective, it is usually better to include some data from higher resolution levels in the reconstructed image before you include all data from the lower resolution levels. In other words, it is better to get the most significant bits from a high resolution level than the least significant bits from the low resolution level preceding it.

Forming the layers and choosing the progression order for a codestream is an important task, particularly if you wish to truncate or parse (see section A.3) a file to meet desired rate/storage constraints. The JPEG 2000 standard does not give normative guidance on how to form layers. In ISO/IEC 15444-1, Annex J, and [Taubman & Marcellin], guidance is given regarding the formation of layers under a mean-squared-error (MSE) constraint.

A.2.7 Codestream Formation

Once the layering and ordering of the entropy coded data is complete, the final codestream (J2C or JPC) may be formed. This includes generation of all the necessary marker segments and filling in all of the marker segment parameters. Additionally, the codestream may be wrapped in the NSIF or JP2 file format. See ISO/IEC 15444-1 for detailed descriptions of the various marker segments.

A.3 JPEG 2000 Decoder Overview

Figure A-4 illustrates the high-level processing chain of a JPEG 2000 Part 1 decoder. Portions of the processing chain enclosed by dashed boxes represent optional or conditional elements.

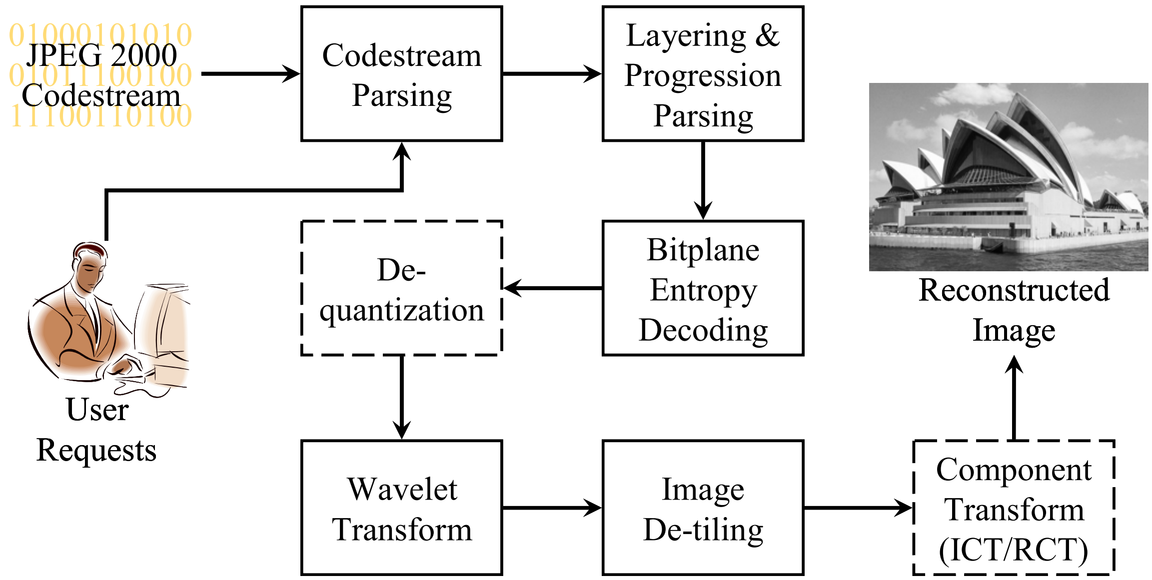

Figure A-4 JPEG 2000 decoder block diagram

A.3.1 Codestream Parsing

Given a JPEG 2000 codestream and the user requests, a JPEG 2000 decoder must determine which portions of the compressed codestream need to be decoded. Many times a user will request the entire image, in which case no parsing need be done. However, the user may request a reduced resolution version of the image, a reduced level of quality to meet a time or storage constraint, a particular tile from the image, or even a general spatial area from the image. The user may also request all of these functions for the same image. In these cases, some parsing of the codestream will be necessary.

There are certain marker segments and encoding options that make the task of codestream parsing easier. Examples are the PLT and TLM marker segments, usage of layers, and the type of progression order used in encoding. The decoder cannot expect that these options have been used. In general, the image might have come from a source that does not know anything about a particular concept of operation. Therefore, decoders must be very flexible and fully featured.

Codestream parsing refers to those parsing operations that occur at the codestream, not the bitstream level. For example, if the user wants a particular tile, the codestream parser will locate the tile within the codestream without accessing individual packets. The codestream parser function will also determine other characteristics about the encoded image such as its size, marker segment usage, bit depth, etc. Once the decoder actually starts manipulating the entropy coded data of the codestream it has moved into the realm of layer and progression parsing.

A.3.2 Layer and Progression Parsing

If a user requests a particular level of quality, resolution level, spatial, area, etc., the decoder must perform layer and progression parsing. Depending upon encoding options (e.g. progression order, layer formation, etc.), this parsing task can be very simple or quite complex. For example, if no layers were used in the encoding process and the user requests a reduced quality version of the image, the simplest approach to use to satisfy the request might be file truncation. However, this will almost certainly yield a poorer quality image than if some intelligent parsing of the codestream were performed. Understanding the JPEG 2000 codestream is essential to achieve proper parsing functionality.

Resolution-based parsing in the absence of a sufficient number of wavelet transform decomposition levels is not defined by this profile. If a user requests an R6 version of the original image and only five levels or fewer of wavelet transform exist in the codestream, the serving application must generate the R6 image. Many techniques might be used to perform this function, including use of the wavelet transform. The procedure used, however, is left up to the implementation.

A.3.3 Bitplane Entropy Decoding

Once the necessary entropy coded data needed to satisfy the user's request has been found, it must be decoded. It is quite possible that a subset of the total codestream has been requested. In this case the decoder must properly track what bitplanes are being decoded in which code blocks to ensure that the data is properly ordered for further processing.

A.3.4 Dequantization

For the 9-7I wavelet transform, explicit dequantization must be done to turn the entropy coded quantization indices back into an approximation of the wavelet coefficients. Embedded quantization of the 9-7I quantization indices (and 5-3R wavelet coefficients) may also occur. The JPEG 2000 standard describes how to reconstruct wavelet coefficients when all of the bitplanes have not been received.

A.3.5 Inverse Wavelet Transform

After the wavelet coefficients have been reconstituted, the inverse wavelet transform is performed. If the user requested a spatial region in the image, it is the decoder's responsibility in the layering and progression-parsing phase to determine which code blocks must be accessed and what wavelet coefficients to decode so that the desired spatial region can be reconstructed. To fully reconstruct a given spatial region, a small number of wavelet coefficients surrounding this region must be decoded as well. The number of coefficients needed is a function of the wavelet filter.

The inverse wavelet transform does not always return an image with pixels in the data structure required for display or further processing. For example, the 9-7I transform returns floating point data. This is converted to an image pixel value (e.g. 8-bit unsigned integer) prior to display or further processing, if needed. Typically, the closest integer value is used, but other conversions may be used to improve image viewing.

A.3.6 Image De-tiling

The image tiles are re-assembled spatially to form the reconstructed image. This process includes any necessary upsampling needed to bring all codestream components to the same spatial resolution. Note that ISO/IEC 15444-1 does not prescribe any upsampling algorithm.

A.3.7 Inverse Component Transform

The inverse ICT/RCT component transform is applied to return the image data back to its original domain. This processing may be swapped with the image de-tiling if no external sub-sampling was performed. If a chrominance sub-sampling algorithm was used, de-tiling should be performed prior to the inverse component transform.

A.4 Enhancing Procedure

Currently, it is recommended that the image is decompressed and standard enhancement procedures are performed on the decompressed image. Most image display and exploitation systems perform the basic enhancement chains of MTFC, DRA, and TTC.

A.5 Parsing

Parsing divides the compressed bitstream into segments that will be retained (for image expansion or repackaging) and segments that will be ignored/removed. Parsing is not a standalone process, but instead is a critical first stage of any application that ingests JPEG 2000 data. Although several types of parsing exist, tile parsing and packet parsing are the only ones required at this time. The more advanced codeblock parsing is not described here.

A.5.1 Tile Parsing

For the NPJE profile, all data for a single tile in the codestream is contiguous and tiles appear in raster order with no tiles missing. The byte length for each compressed tile is contained both in the main header TLM and the tile header Psot field. Once the main header length is determined (the main header ends as soon as the first SOT marker is encountered), this info can be used to find the exact codestream locations for every tile (See Figure A-5. Note that for NPJE $Ptlm^i = Psot^i$).

$$\text{Tile M bounds:} \quad \begin{array}{c} MainHeaderLength + \sum_{i=0}^{M-1} Ptlm^i \\ \text{to} \\ MainHeaderLength - 1 + \sum_{i=0}^{M} Ptlm^i \end{array}$$

Equation A-1

In other BPJ2K01.10 preferred encodings (not NPJE), tiles can be separated into tile-parts with one SOT corresponding to a single tile-part. When this happens, the above formula is no longer accurate, and other techniques (available in JPEG 2000 software libraries) must be used to locate each tile within the codestream. For example, the EPJE profile uses tile-parts to split tile codestreams into tile-part based codestreams at each resolution level. The tile-part codestreams across all tiles from like resolution levels are laid out contiguously in the compressed file. This means that the codestream from a given tile is interleaved with the codestreams from other tiles across the image. In this case tile parsing is a more challenging task.

Figure A-5 illustrates the file layout of an NPJE codestream and the relationship of the TLM marker segment lengths ($Ptlm^i$) to the layout of the file. It is clearly seen that tile-based parsing (chipping) of an NPJE codestream is readily accomplished using the relationship in Equation A-1.

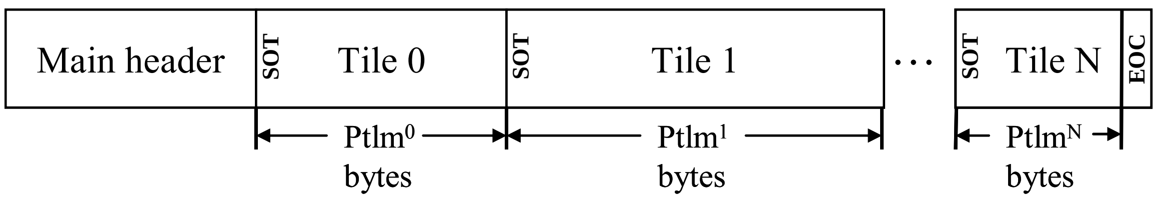

Figure A-5 Locating tile boundaries using tile lengths in NPJE

Tiles are independently processed in JPEG 2000; therefore a given tile will contain all the component, resolution, and quality data for its corresponding spatial location. The tile data might be interleaved with other tile data through the use of tile-parts, but it is relatively straightforward to parse out all tile-parts belonging to a tile and collect them back into a single tile-part. The following descriptions of other processes such as packet parsing and expansion will therefore focus on a single tile. With multiple tiles, each tile can be processed independently.

A.5.2 Packet Parsing

A finer level of data detail can be obtained via packet parsing. Packet parsing determines which packets contain information pertinent to the requested component, resolution, and/or quality within a tile and locates the boundaries of those packets in the codestream.

Packets appear in a specific sequence (identified by information in the COD and POC marker segments). Each packet can be labeled by content: component, precinct, additional quality (layer), and additional resolution. The progression order identifies the order of the ‘for’ loops that traverse the data. The extent of the loops is initially set by the COD marker segment (RSpoc=0, REpoc=SPcod: N_{Levels}+1, CSpoc=0, CEpoc=Csiz, LYEpoc=SPcod: N_{Layers}) or may be modified within a POC marker. (POC markers are not allowed in certain profiles in this document, but may appear in a generic JPEG 2000 codestream.)

We will now consider an NPJE codestream. The recommended progression is L-R-C-P (layer-resolution-component-precinct), of course other progression orders will be found in generic JPEG 2000 imagery and other profiles defined in this document. The L-R-C-P progression orders the codestream packets in the following fashion.

L-R-C-P: For each quality layer q = 0, …, LYEpoc - 1
For each resolution delta r = RSpoc, …, REpoc-1
For each component, c=CSpoc, …, CEpoc-1
For each precinct, p
Packet P(q,r,c,p) appears.

Other progression orders change the sequencing of the "for" loops. See ISO/IEC 15444-1 for details about other progressions.

In addition to identifying each packet by precinct, component, quality and resolution, the parsing operation must also delineate the codestream boundaries of all useful packets. This can be done in two ways; via PLTs or via decoding of packet headers.

If PLT markers are present in the tile header, they can be decoded to indicate the length of each packet in order of appearance (see Figure A-6). This length information combined with the location of the end of the tile header (SOD marker) provides precise locations for all packets. This is the fastest way of finding packet boundaries when a significant number of packets will be ignored, or when packets must be reordered.

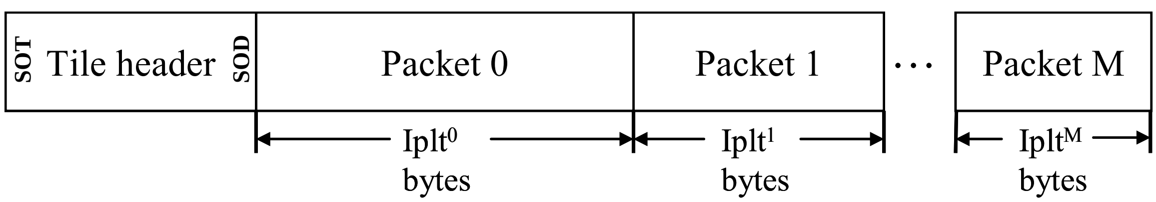

Figure A-6 Locating packet boundaries by length in NPJE

When PLT markers are not present in a tile header, each packet header must be decoded in turn to discover the cumulative packet length (sum of codeblock lengths) and then skip forward to the next packet header. Since packet lengths are encoded as a combination of other lengths within the packet header, this process will take longer than reading packet lengths directly from PLT markers. However, all JPEG 2000 file parsers must be able to decode packet lengths from packet headers since PLTs are not guaranteed in all codestreams.

A.5.2.1 Packet Parsing to Extract Quality Layers

For archival purposes, primary data providers will often compress imagery at very high bitrates (i.e. fairly low compression ratios) in order to preserve either visually lossless quality or numerically lossless quality in the image that ends up being archived in a library somewhere. However, most users are located at sites that do not have the bandwidth to receive every incoming image at visually lossless quality. This means that most users end up requesting imagery that is significantly reduced in bitrate relative to the visually lossless or numerically lossless version that was initially output by the data provider.

A slight reduction in image quality for a significant reduction in file size is a beneficial tradeoff for most users. Therefore, one of the most important capabilities that a JPEG 2000 parser will have to fulfill is the extraction of a subset of quality layers at full resolution from a JPEG 2000 compressed codestream. The scalability of a JPEG 2000 codestream that comes from layering will greatly reduce the processing burden on a processor. In legacy systems using traditional compression algorithms, a parser that wants to disseminate less than the full quality was forced to decompress the archived image and recompress it to the desired bitrate for every request. With JPEG 2000, decompression will usually not need to be done until the image is at a workstation ready for display.

Most data providers should partition every JPEG 2000 codestream into multiple layers with target bitrates for the layers that span the full range of possible requirements (in terms of quality vs. bitrate) of the various users of imagery from that provider. This target bitrate information is not signaled in the JPEG 2000 codestream itself; it is either maintained in a configuration controlled database to which the parser has access, or it may also be conveyed through the use of the NSIF J2KLRA TRE that is part of the image segment subheader. A priori knowledge of each layer's target bitrate simplifies a parser's implementation because it can then be determined (in advance) exactly how many packets are needed (based on the index of the desired layer) for extraction of a particular layer. Without a priori knowledge of the layer target bitrates, a parser can still get to a requested bitrate by simply keeping a running sum of the bits used as packets are being extracted and skip to the next tile when the requested rate is exceeded for the current tile.

Layer extraction at full resolution is easy to do in JPEG 2000 with L-R-C-P progression and just one tile-part per tile. Layers are implicitly ordered from lowest bitrate to highest bitrate. The for loop structure described in section A.5.2 for L-R-C-P progression shows that all packets (spanning resolution, component, and position) for a given layer come before any packets of any other layer. Knowledge of the number of wavelet transform levels (N_{Levels}) is conveyed in the COD marker, and that field tells a parser that there are (N_{Levels} +1) resolution levels (and therefore that same number of resolution packets) for each layer.

Assuming that precincts are not used (a reasonable assumption given that 1024x1024 tiles are recommended) and that there are the same number of wavelet transform levels performed for each tile, then a parser that is interested in layer *i* simply needs to extract [i * (N_{Levels} +1) * Csiz] packets from that tile. With L-R-C-P progression, these packets will appear contiguously and the parser does not have to skip from one location to another to extract these packets.

Once the necessary number of packets has been extracted for layer *i*, the parsed packets can be repackaged as a new standalone image by making the following additional modifications:

- Update the number of layers (N_{Layers}) field in the COD marker to correspond to the number of layers in the parsed image.

- Update the TLM and PLT markers to reflect the new offsets (lengths) for tiles in the image and for packets within each tile.

For other progression orders where layers are not in the outermost progression loop, packet parsing to extract a quality layer will require additional seeking within the compressed file to obtain the desired packets. This may increase processing timelines and implementers need to carefully consider how disk access patterns, JPEG 2000 codestream formation and their system conops all interact. No single progression order will performed the best at all server repackaging and parsing tasks. For example, the NPJE profile does not perform well for reduced resolution roam and zoom tasks. The EPJE profile was created to address this issue in electronic light table (ELT) applications.

A.5.2.2 Packet Parsing to Extract Reduced Resolution Data Sets

Extraction of reduced resolution data sets from a codestream with L-R-C-P progression order is fairly straightforward. Again, it is implicit that the resolution packets in each layer are ordered from lowest resolution to highest. One basically follows the same procedure outlined in section A.5.2 for extracting full resolution layers except that packets for resolutions higher than the resolution level of interest are simply skipped.

There is one very important difference to keep in mind when extracting reduced resolution data sets: the target bitrates for the full resolution layers (as maintained in a configuration-controlled database or as spelled out in the J2KLRA TRE) do not apply for reduced resolution data sets. Therefore, to get a reduced resolution image at a specific target bitrate for that resolution, a parser needs to keep track of the running total number of bits used as the packets are being extracted.

When the number of bits is exceeded on the current tile for the requested bitrate at the resolution level of interest, the parser can then skip to the next tile. This is actually not that much different than what needs to be done for layer extraction in the case where the parser does not have a priori knowledge of the target bitrates for each layer. In that case, the parser needs to keep track of the running total bit usage even for layer extraction.

It is worth pointing out that although the layer breakpoints are defined for the full resolution image (R0), the number of full resolution layers also determines the granularity of achievable bitrates for the reduced resolution data sets. Having lots of layers at many different bitrates (particularly very low bitrates) for R0 allows a parser the flexibility to get close to the desired target bitrate at reduced resolutions. Thus, it is a good idea for data providers to use a lot more layers than would be required for just R0 because it facilitates parsing reduced resolution data sets at specific bitrates.

Because resolution parsing changes the dimensions of the parsed image, more marker values need to be updated when repackaging than in the layer parsing case. When parsing a lower resolution out of a codestream, the following modifications need to be made before repackaging the packets as a standalone image:

- update the number of layers (N_{Layers}) field in the COD marker to correspond to the maximum number of layers in the parsed image.

- update the N_{Levels} field in the COD marker.

- update the TLM and PLT markers to reflect the new offsets (lengths) for tiles in the image and for packets within each tile.

- update the SIZ marker fields to account for the new image and tile size.

- shorten QCD marker(s) to be consistent with the new N_{Levels} values.

It should be noted that JPEG 2000 relies on the resolution hierarchy built into each codestream to achieve its compression efficiency. The practice of extracting and disseminating entire suites of R-sets (R1 – R9) should be kept to a minimum with JPEG 2000 because in most cases it is unnecessary. The resolution scalability of JPEG 2000 means that a parser or ELT can extract just the resolution packets of interest from the R0 codestream without having to maintain physically separate copies of reduced resolution data sets on the server. Thus, in most cases it will suffice to just send the R0 image because the other lower resolution versions of the image are embedded in the R0 codestream.

A.6 JPEG 2000 Decoding and RRDS Generation

Parsing first determines what tile packets are present in the bitstream. It is possible that a codestream will have different encoding parameters within a tile. This can include changing the number of wavelet decomposition levels, even to the point where the wavelet transform is turned off. Alternatively, the packet data within a tile might have been parsed and packets are empty for higher resolution levels. When a tile is missing packets for a particular resolution, the application must choose an appropriate decode resolution. A codestream whose components have different sub-sampling factors on the reference grid will have different resolutions for the R0 image. Thus an application must be prepared to handle tiles and components of varying resolution.

As described in section A.5, the parsing operation is used to restrict attention to the portion of the codestream that must be expanded. It is recommended that viewing applications use parsing and selective caching to provide rapid access to thumbnail views, zoom, and pan. This section describes interactions between the parsing, dequantization, inverse wavelet and image pixel creation processes.

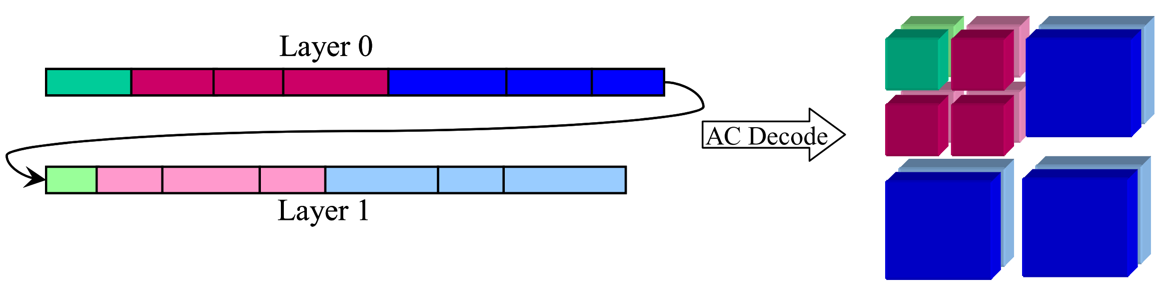

Figure A-7 Parsing from a JPEG 2000 codestream to quantized transform coefficients

Parsing is performed based upon application requirements and in some applications such as interactive viewers may occur multiple times during the compression process. Efficient implementations will store information about packet locations whenever possible, so that later parsing operations only involve file access. Once the data to be expanded is located in the codestream it is processed by the arithmetic decoder and interpreted as a decoded quantized coefficient, as depicted in Figure A-7. The quality of the coefficient values at this point depends upon the number of layers (two in this example) that are parsed and decoded from the bitstream.

Once a quality/truncation point is chosen, the expansion can proceed to the dequantization stage. If we assume an NPJE profile codestream, at R5 it is possible to generate a very high quality image from just a single layer (the truncation point at 0.03125 bpppb). However, if lower RRDS (higher resolutions) are desired, then more available layers should be included in the decoded quantized coefficient to aid in later processing.

[If viewing at multiple quality points is desired within a single application, then the layering of each decoded coefficient must be accessible. This can be achieved by either re-computing the decoded quantized coefficients whenever required, or by storing side information for each coefficient indicating how many bit-planes are included per layer. This type of information can be readily derived during initial decoding.]

Figure A-8 shows how extra coefficient accuracy, included prior to dequantization, improves the reconstructed (dequantized) transform coefficients. Two layers are shown in the figure for demonstration purposes. In practice more layers may be accessible. This figure also indicates that the LL band can be converted to image pixel data without performing an inverse transform.

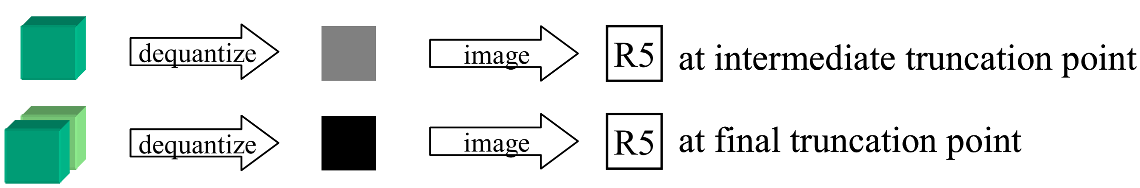

Figure A-8 Quantized transform coefficients synthesized to form image at R5. Extra layer adds extra quality (darker shade) to LL transform coefficients.

Once the floating point LL coefficients for Rx are generated, an integer image can easily be created. However, the LL coefficients will also need to be used to create the RRDS R(x-1). For example, as shown in Figure A-9, the LL for R5 combined with the R4 high-pass data is inverse wavelet transformed to generate the LL coefficients for R4. The integer version of these coefficients is R4, while the floating point version is carried into the R3 computation. This process is continued until R0 (or the lowest available RRDS) is achieved.

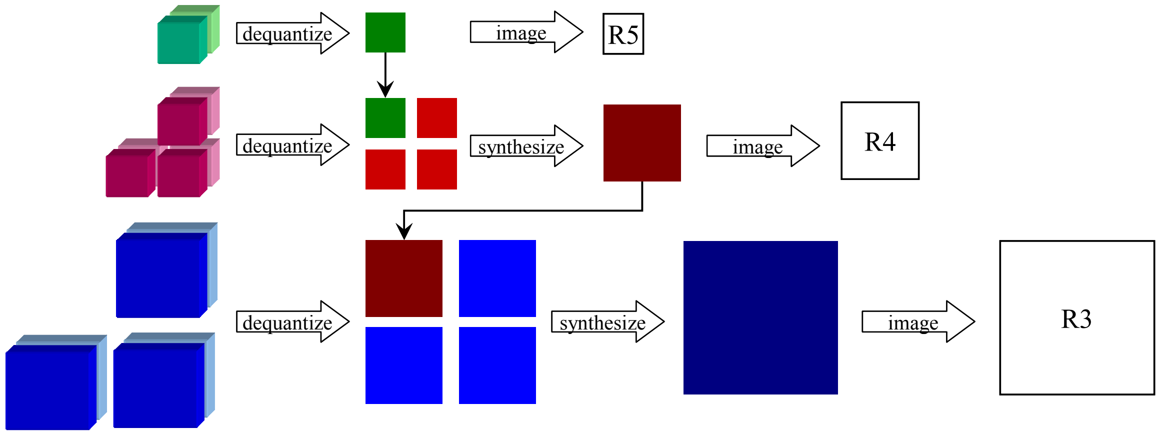

Figure A-9 JPEG 2000 expansion with R-set generation

If no reduced resolution data sets (or a limited subset of) are required in a particular application, then the final image generation stage is only performed for the requested data sets. Image display applications, which can display directly from floating point, may be able to perform the final image creation step within current display software.

If there is more than one tile in the codestream, then each can be decoded independently using the above techniques. Tiles are adjacent and non-overlapping, numbered in raster order, and cover the entire image area. The decoded tile data is placed in the appropriate image pixel location based upon the tile number and this raster ordering.

A.7 Repackaging

Repackaging (also known as transcoding) consists of partially decoding or parsing a JPEG 2000 codestream to create another valid JPEG 2000 codestream. Repackaging can create a valid NPJE file from another NPJE file. Repackaging can also be performed on files that are not compliant with NPJE, but the resulting file is not expected to be compliant to NPJE either. Repackaging data is a common practice for library applications where data is requested at different quality, resolution or size from how the data is stored. Any system that produces a derived product that changes the image in some manor should follow the repackaging procedures in this section.

Figure A-10 shows the various repackaging paths in JPEG 2000. For NPJE repackaging requirements, only repackaging path "1" need be considered. On this path, two relatively short processes replace the much longer expansion to an image and recompression path. In addition to reducing computational complexity, repackaging data prevents the incremental reductions in image quality that occur when data is expanded and recompressed.

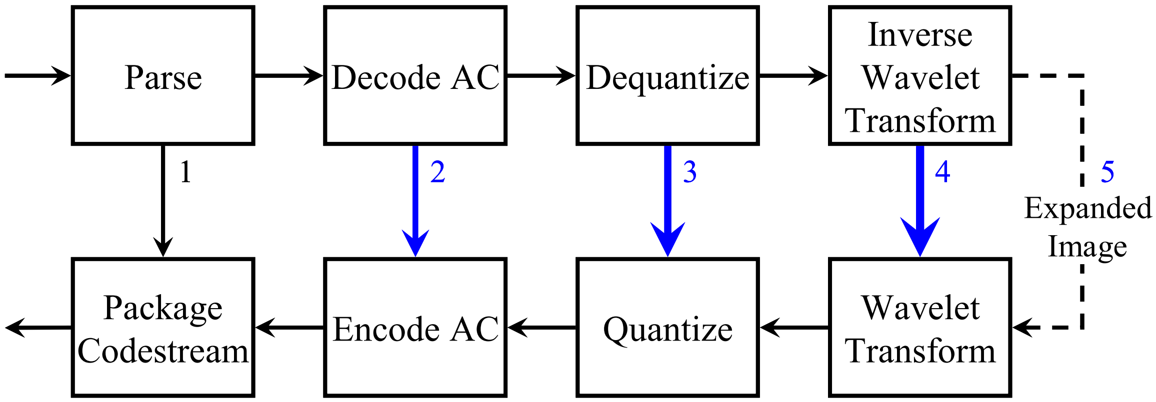

Figure A-10 JPEG 2000 repackaging paths

As shown in Figure A-10, first the compressed tiles/packets to be repackaged are parsed out of the input file. These tiles/packets are then packaged by creating or modifying headers as needed and potentially reordering the packets. Certain JPEG 2000 marker segments and NSIF information will need to be updated during this procedure, but typically a large portion of the file will be copied from information parsed from another file.

Depending upon the type of processing requested, a library server may need to utilize some or all of the enumerated paths shown in Figure A-10. The level of processing and computational complexity increases moving from left to right in the figure. Additionally, higher levels of processing require that lower processing levels be performed as well. For example, if a library must dequantize wavelet coefficients to accomplish a repackaging task, it will also need to rerun the arithmetic coder and repackage the codestream. The most expensive repackaging tasks will require that the image be decoded back into pixel space and encoded again. A common example of this type of processing would be adding tiling to a codestream that was not originally tiled.

Examples of JPEG 2000 marker segments that may change during standard repackaging of an NPJE codestream are:

SIZ marker:	image size, # components, sampling resolution.
COD marker:	N_{Levels}, N_{Layers}, progression order.
QCD marker:	number of subband entries depends on N_{Levels}
Tiles:	Labeling tiles in raster order (Isot), computing tile lengths (Psot), modifying TLM.

Other changes that may occur with JPEG 2000 codestreams that do not follow the NPJE profile or more unusual repackaging operations are:

SIZ marker:	image/tile offsets
Tiles:	Addition of multiple tile parts per tile

Packets: Modifying packet lengths/orders recorded in PLTs. POC addition SOP/EPH deletion/addition.

NSIF fields must be modified to match the new JPEG 2000 header information (see Section 7.2) and the conditional fields of any J2KLRA TRE must be updated to reflect repackaging modifications. In addition any spatial chipping requires the addition or modification of the ICHIPB TRE.

A.7.1 JPEG 2000 Codestream Repackaging Changes

Table A-1 shows which JPEG 2000 and NSIF header fields might require modification within a single image segment for each of the four primary repackaging operations: reduced quality, reduced resolution, fewer components, and reduced spatial extent. We consider as an example repackaging an NPJE codestream to form an EPJE codestream. Obviously Table A-1 is incomplete, not all possible codestream/profile transcoding possibilities can be covered. This table serves as an example of the types of operations that might be necessary.

An ‘x’ entry indicates that the particular header field might change during the repackaging indicated in that column. The entry "remove" indicates that particular fields or portions of the codestream are removed entirely. The entry "remove some" means that some elements are removed entirely, while others remain in place without change.

For spatial chipping there are often two or three choices, depending upon whether an individual tile is retained, emptied, or removed.

In Table A-1, the JPEG 2000 SIZ modifications shown for spatial chipping and resolution correspond to one possible repackaging system. Other repackaging systems are also valid.

Table A-1 Possible JPEG 2000 Codestream Modifications

Header Field	Spatial Chipping	Reduce Quality	Reduce Resolution	Fewer Components	Transcode NPJE to EPJE
JPEG 2000					
SIZ					
Xsiz/Ysiz	x				
XTsiz/YTsiz					
XOsiz/YOsiz	c[1]				
XTOsiz/YTOsiz	c				
Csiz				x	
Ssiz[1]				remove some	
XRsiz/YRsiz			x	remove some	
COD					

[1] C= conditional change. For NPJE with tile size = 1024, no change is required. However, if odd tile sizes (vs. even) are used, then these fields cannot be maintained at zero, and must change based upon the reference grid location of the chip.

Table A-1 Possible JPEG 2000 Codestream Modifications

Header Field	Spatial Chipping	Reduce Quality	Reduce Resolution	Fewer Components	Transcode NPJE to EPJE
JPEG 2000					
Progression Order					x
N_{Layers}		x	x		
N_{Levels}			x		
QCD					
Lqcd			x		
SPqcd			x		
TLM					
Ltlm	x				x
Ztlm	x				x
Stlm					x
Ttlm[i]					x
Ptlm[i]	x if emptied	x	x	x	x
Entire Tile (Header + Data)	remove some				
SOT					
Isot	x				
Psot	x if emptied	x	x	x	x
TPsot					x
TNsot					x
PLT	remove if emptied	remove some			
Lplt			x	x	x
Zplt					x
Iplt[i]			remove some	remove some	x
SOD					
Packets	remove if emptied	remove some	remove some	remove some	reorder across tile-parts
NSIF					
NROWS/NCOLS	x		x		
NBPR/C	x				
NPPBH/V			x		
PVTYPE				x	
COMRAT		x	x		
N/XBANDS				x	
ILOC	x				
IMAG			x		
J2KLRA TRE		x	x	x	x
ICHIPB	x				

A.7.2 Repackaging Across NSIF Image Segment Boundaries

When the spatial region in an NSIF JPEG 2000 codestream being repackaged crosses an image segment boundary, it may be desirable to reduce the number of image segments. This can be accomplished by merging individual repackaged image segment codestreams into one JPEG 2000 codestream. In general this type of repackaging relies on tile-based codestreams with a constant tile size across image segments. The NPJE and EPJE profiles meet this criterion, but not all profiles and codestreams will. If the tile sizes in the relevant image segments differ, it will be necessary to re-tile the image segments. In this case maintaining independent image segments might be preferred.

Repackaging across image segments is done by creating new image size information for the JPEG 2000 main header, merging the tile header info, and concatenating the repackaged tile data from each of the image segments. If the main header QCD/QCC and COD/COC are not identical in each of the separate codestream segments, then one segment must be chosen to provide the defaults, while tiles from the other segments must be modified to include tile specific QCD/QCC and/or COD/COC marker segments. The NSIF image subheader and relevant TREs (ICHIPB, J2KLRA, and HISTOA) should be modified as appropriate for the new segment.

Merging across segment boundaries must obey segment size limitations.

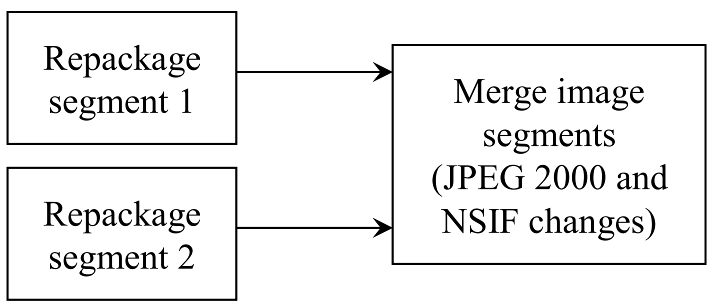

Figure A-11 Merging image segments

A.7.3 Changing the JPEG 2000 Progression Order

When an application requires a change in progression order or layering, transcoding to a new progression order and re-layering is completed without actually changing any of the encoded data. Encoded data packets are rearranged to meet the new progression order and layering. New packet headers are created and the appropriate modifications to COD/COC, PPM/PPT and PLM/PLT marker segments are made. Depending upon the type of progression order change, a POC marker segment may be required and new tile-parts introduced (e.g. EPJE). When this transcoding is complete, the values in the J2KLRA TRE should be updated. The ORIG field should represent the new encoding style, and the NLAYERS_I should be updated to the proper number of layers. The LAYER_ID values should also be updated to represent the new values.

A.8 Advanced Repackaging

The following types of JPEG 2000 transcoding require more advanced repackaging techniques and are not expected to be required for most applications (paths "2", "3", "4", and "5" in Figure A-10). Note that it is usually not necessary to completely decode the codestream (path 5) to generate an image and then recompress it. For most repackaging tasks, the worst case scenario is that only a few wavelet coefficients may need to be recomputed (path 4). Most repackaging tasks can be accomplished along other paths (1, 2, 3).

- Add/remove tiling (change tile size)
- Add/remove precincts (change precinct size)
- Change progression order (add POC)
- Arbitrary cropping (not on tile boundaries)
- Change codeblock size
- Add/Remove ICT/RCT
- Change wavelet transform used
- Change layering scheme
- Remove/add components
- Change number of decomposition levels
- Change quantization
- Create tile-parts
- Add TLM, PLM, PLT marker segments
- Add/remove SOP and EPH marker segments
- Change arithmetic encoding style

A.9 Thumbnail, Overview, and Additional Resolution Generation

In certain cases, a codestream may not contain a sufficient number of resolution levels for purposes such as thumbnail and overview generation. Lower resolutions can be created by using either the 9-7I or 5-3R low-pass wavelet filters for sub-sampling. The same wavelet transform should be used as that employed in generating the original codestream. Additional resolution generation is achieved by reconstructing all tiles at the lowest resolution level (R5 for an NPJE codestream) and forming a single image without tiles. If this image is too large and tile-based processing is needed, then new tiles can be created. JPEG 2000 wavelet processing is then applied to the retiled image to generate further reduced resolution images. The number of additional wavelet decomposition levels to generate depends on the original codestream and the application.

Note that other filtering and decimation techniques may be used to achieve similar effects. Retention of these additional R-sets may be desired for subsequent use by other users. This could alleviate the need for repetitive generation of new R-sets. While this subject is beyond the scope of this standard, it is suggested that JPEG 2000 encoding of such R-sets be used if they are to be retained.

Appendix B JPEG 2000 Process Examples

The following sections have been developed to support the developer in the process and procedures that are required to achieve functionality of JPEG 2000.

B.1 Reduced Resolution Chipping

For certain CONOPS it is useful for the user to request only a lower resolution version of an image. For example, R5 or R4 imagery might be used for generalized route planning or briefings. When this occurs, the packets corresponding to the higher resolutions should not be transmitted.

In addition to eliminating packets based upon resolution, it is also often possible to eliminate quality packets. As a general rule of thumb layers that increase the bits-per-pixel at the desired resolution above 4.3 bpp can be eliminated. For example, if an R3 image is requested, then the R3 bpp for the available layers can be computed (PLTs help with this). As an example, on an NPJE encoded test image the R3 bpp values were:

Layer	R0 image bpp	R3 bpp (image specific)
0	0.03125	1.5
1	0.0625	2.3
2	0.125	3.3
3	0.25	4.3
4	0.5	5.6
……		

So only packets for layers 0-3, and resolutions R5-R3 will be maintained in the transmitted bitstream.

For the lowest resolution, R5, it is possible that the first layer will have more than 4.3 bpp. When the first layer exceeds the requested quality, then it should be included in the parsed file anyway, since the alternative, to send no data at all, is generally unacceptable.

Once the necessary packets for R*x* are located in the bitstream they are repackaged in the same order with appropriate changes in the main/tile headers. As an example, assume that the input NPJE codestream contains R0-R5. In the SIZ marker segment, multiply XRsiz and YRsiz, by $2x$. In the COD marker segment, subtract x from N_{Levels}. Any QCD/QCCs whether in the main or tile headers shall be shortened to reflect the change in N_{Levels}, and the PLTs shall be shortened to eliminate interspersed non-existent packet info. All tile lengths must be updated (TLM and Psots).

Care must be exercised when performing reduced resolution chipping. For the NPJE and EPJE profiles, the tile size is a power of two and there are no tile or image offsets. It is possible to leave XRsiz and YRsiz unchanged in this case and modify the reference grid size (Xsiz and Ysiz). This is not recommended practice, however. In the general case, tiles need not be powers of two in size and the top left corner of tiles cannot be guaranteed to lie on points that are multiples of a power of two. In these cases it is best to leave reference grid and tile offsets unchanged (XOsiz, YOsiz, XTOsiz and YTOsiz) as well as the reference grid size (Xsiz, Ysiz). Only the reference grid sampling factors

($XRsiz^i$, $YRsiz^i$) should be manipulated. By adhering to this guideline, it will not be necessary to recalculate wavelet coefficients

The following NSIF fields and TRE will need to be updated appropriately: NROWS, NCOLS, IMAG, COMRAT, NPPBH, NPPBV, J2KLRA TRE, ICHIPB TRE, and HISTOA TRE. Additionally, in some cases the IDLVL, IALVL, and ILOC fields will also need to be updated.

B.2 Spatial Chipping at Tile Boundaries

For the NPJE profile, spatial chipping is most efficiently performed as the combination of forming a rectangular subset, and emptying the unwanted tiles within the rectangular subset. This allows transmission of oddly shaped regions, such as the vicinity of a road over a long distance or a flight path, while still avoiding large overhead from the original image area.

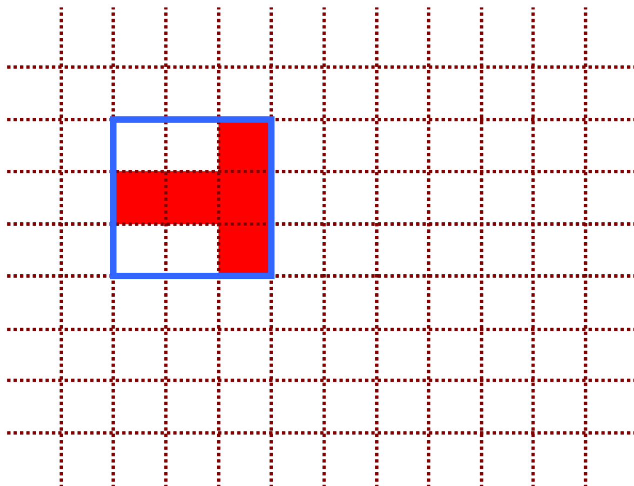

Figure B-1 Chipping: rectangular subset (blue target rubber banding) with four emptied tiles (shown in white). Only shaded tiles retain image data.

Since all tile data is contiguous and tile lengths are included, it is possible to locate tiles without decoding any image data. Moreover TLMs in the main header, make it easy to quickly locate any individual tile. All tiles outside the rectangular subset are removed, and the remaining tiles are renumbered in raster order. Any tiles within the rectangular subset that are to be 'emptied' are then reduced to a minimal tile; SOT segment of length 12 and an SOD (Psot=14). Any 'emptied' tiles will expand to a middle gray value.

When repackaging the codestream, tile renumbering will require changes within the TLM marker segments and the Isot fields. The TLM must be updated to reflect the lengths of empty tiles.

The new rectangular image area will require modifications to Xsiz/Ysiz, and possibly XOsiz/YOsiz, and XTOsiz/YTOsiz. When tile sizes are powers of 2 that are divisible by the sub-sampling values (XRsiz, YRsiz) the image and tile offsets will remain at zero, and Xsiz/Ysiz will be reset to reflect the new image dimensions. This will occur for all imagery that follows the NPJE guidelines. However, if the tile sizes do not satisfy the above conditions, then the new image and tile offsets and image bounding dimensions will need to be set based on the original J2K reference grid boundaries of the chip.

Some of the NSIF header fields also need to be modified during chipping: NROWS, NCOLS, NBPR, and NBPC, and possibly IDLVL, IALVL, and ILOC as appropriate. An ICHIPB TRE must be added (or modified) to correctly identify where the chip resides relative to the originally collected image. Additionally, based on a user-community best practice, the J2KLRA TRE and HISTOA TRE need to be updated as well to record the spatial chipping event.

For imagery that is not compressed to the NPJE guidelines, tile-based chipping is more problematic. As discussed in section A.7, further repacking processing may be needed. It is possible that the imagery is not tiled at all. In this case recoding of the image from the pixel domain will be necessary. The tiles present in the codestream might not be powers of two in size or they might be offset from the reference grid origin. In these cases it is better not to try to relocate the reference grid origin. Instead the reference grid sampling factors ($XRsiz^i$ and $YRsiz^i$) and image and tile offsets (XOsiz, YOsiz, XTOsiz and YTOsiz) should be used to prevent recalculation of wavelet coefficients.

B.3 Spatial Chipping Off-tile Boundaries

When a spatial chip must be created that does not follow tile boundaries, then there are two new elements to consider.

- Resetting the image and possibly tile offsets to retain coding of tile areas whose borders do not change.
- Recoding of tiles that have lost some of their original pixels.

When the upper left corner of the spatial chip is at a tile corner, then image and tile offsets obey the guidelines outlined in Section B.2. However, if the upper left corner is in the middle of a tile the following rules apply:

- Typical case: (when tile sizes are a power of 2 and divisible by image sampling)
    - Tile offsets remain at 0
    - Image offset = Original reference grid offset for the chip modulo tile size
- Very unusual case:
    - Image offset = Reference grid chip offset
    - Tile offset = Reference grid offset of first tile.

Any non-empty tile that does not contain the full complement of pixels that were originally encoded, must have some portion of the edge code-blocks retransformed and recoded, to reflect the change in wavelet transform boundary values. In the example shown in the figure below tiles 2 and 3 must be modified in this manner. Empty partial tiles are identical to empty full tiles so no special handling is needed for the off tile border case.

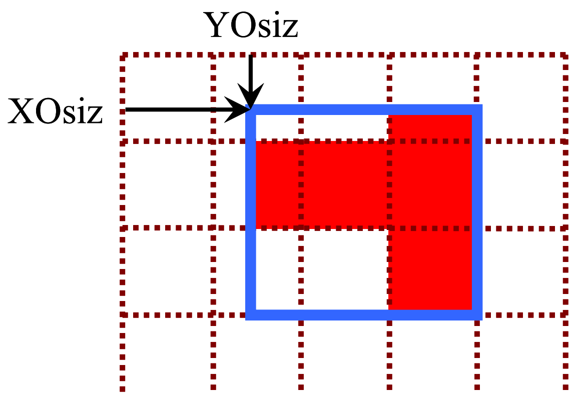

Figure B-2 Spatial Chipping at Non-Tile Boundaries: Only shaded tiles retain image data.

All other repackaging details follow the guidelines shown in Section B.2.

B.4 Quality Chipping

Many imagery requests will require some selection of image quality. The distributor must then parse and repackage a file that contains only the requested bitrate. Since source imagery is stored in quality progressive order, this is a simple operation. The packets associated with the unwanted quality layers are truncated from each tile. Repackaging modifies a small amount of header info (Psot, TLM, PLT truncation and COD: N_{Layers}). The retained packets are not reordered. Only NSIF field COMRAT must be updated.

Alternatively null packets might be used. Null packets contain no codestream data and have only empty packet headers (empty packet headers cost one byte per packet header). Null packets allow the original layering/packet structure to be retained while removing the codestream data. This may be important for some applications where the number of packets is a fixed number. The null packets approach still requires modification of some codestream elements (Psot, TLM, PLT truncation).

In typical usage a combination of location and quality chipping will be used. Spatial chipping to remove/empty tiles should be performed first, and then quality chipping. Repackaging will require changes from both types of chipping.

Appendix C JPEG 2000 Commercial Profiles (ISO/IEC IS 15444-1 Annex A.10)

JPEG 2000 profiles were introduced in ISO/IEC 15444-1 in order to promote interoperability. Two profiles are defined, Profile-0 (Rsiz = 1) and Profile-1 (Rsiz = 2). An implicit "No Restrictions" (Rsiz = 0 or NR Profile) profile exists and it means that the codestream simply conforms to the JPEG 2000 Part 1 standard. Additionally, ISO/IEC 15444-1:2004/AMD 1:2006, defines two additional profiles (Rsiz = 3 and Rsiz = 4, 2K and 4K DCI distributions respectively) for use within the Digital Cinema Initiative (DCI).

The only commercial profiles allowed for use with any BIIF JPEG 2000 preferred encoding defined in this standard are Profile-0, Profile-1 and the NR Profile (i.e. Rsiz = 1, 2 or 0). The DCI profiles are not allowed. If a codestream happens to be compliant to either DCI profile (this is very unlikely, but possible under the LPJE profile), it shall be marked as the NR profile (Rsiz = 0).

Profile-0 and Profile-1 are defined as follows. Profile-0 was developed for low complexity applications (i.e., cell phones, PDA, and other limited systems) and all decoder systems must be compliant to Profile-0. Profile-1 is the expected common commercial application profile. It is expected that common web browsers, digital photographic software products, and image collection systems will be compliant to Profile-1 or higher. NPJE adheres to all of the limitations of Profile-1. Other BIIF profiles within this document do not necessarily adhere to Profile-1. In this case they should be marked as the NR profile. A codestream should always be marked at the lowest complexity profile consistent with its encoding parameters.

Table C-1 Codestream Restrictions

Restrictions	Profile-0	Profile-1
SIZ marker segment		
Profile Indication	Rsiz = 1	Rsiz = 2
Image Size	Xsiz, Ysiz < 2^{31}	Xsiz, Ysiz < 2^{31}
Tiles	Tiles of a dimension 128x128: YTsiz = XTsiz = 128 or one tile for the whole image: YTsiz + YTOsiz ≥ Ysiz XTsiz + XTOsiz ≥ Xsiz	XTsiz/min(XRsiz^i,YRsiz^i) ≤ 1024 XTsiz = YTsiz or one tile for the whole image: YTsiz + YTOsiz ≥ Ysiz XTsiz + XTOsiz ≥ Xsiz
Image & tile origin	XOsiz = YOsiz = 0 XTOsiz = YTOsiz = 0	XOsiz, YOsiz < 2^{31} XTOsiz, YTOsiz < 2^{31}
RGN marker segment	SPrgn ≤ 37	SPrgn ≤ 37
Sub-sampling	XRsiz^i = 1, 2, or 4 YRsiz^i = 1, 2, or 4	No restriction
Code blocks		
Code-block size	xcb = ycb = 5 or xcb = ycb = 6	xcb ≤ 6, ycb ≤ 6
Code-block style	SPcod, SPcoc = 00sp 0t00 (where t, p, s can be 0 or 1) Note: t=1 for termination on each coding pass p=1 for predictive termination	No restriction

Table C-1 Codestream Restrictions

Restrictions	**Profile-0**	**Profile-1**
	s=1 for segmentation symbols	
Marker Locations		
Packed headers (PPM,PPT)	Disallowed	No restriction
COD, COC, QCD, QCC	Main header only	No restriction
Subset Requirements		
LL resolution	If one tile is used for whole image, (Xsiz – XOsiz)/D(I) ≤ 128 and (Ysiz – YOsiz)/D(I) ≤ 128 where D(I)= $2^{\text{number of decomposition levels}}$ in SPcod or SPcoc, for I = component 0 to 2	For each tile in the image, $\lfloor tx1/D(I)\rfloor - \lfloor tx0/D(I)\rfloor \le 128$ and $\lfloor ty1/D(I)\rfloor - \lfloor ty0/D(I)\rfloor \le 128$ where D(I)= $2^{\text{number of decomposition levels}}$ in SPcod or SPcoc, for I = component 0 to 3 Note: tx0, tx1, ty0, ty1 are defined in Annex B of ISO/IEC 15444-1, equations B.7 through B.10.
Parsability	If the POC marker is present, the POC marker shall have $RSpoc^0 = 0$ and $CSpoc^0 = 0$. (Note some compliant decoders might decode only packets associated with the first progression)	No restriction
Tile-parts	Tile-parts with TPsot = 0 of every tile before any tile-parts with TPsot > 0, Tile-parts Isot = 0 to Isot = number_of_tiles – 1, in sequential order for all tile-parts with TPsot = 0	No restriction
Precinct size	"Precinct size" defined by SPcod or SPcoc (Table A-15 and Table A-21 of ISO/IEC 15444-1) must be large enough so there is only one precinct in all resolution levels with dimension less than or equal to 128 by 128. NOTE – Precinct size PPx ≥ 7 and PPy ≥ 7 is sufficient to guarantee only one precinct per subband when XOsiz = 0 and YOsiz = 0.	No restriction

Profile-0 is a very restrictive profile and is of little interest to implementers and programs that are subject to the BIIF Profile of JPEG 2000. It was originally intended for mobile and cell phone applications but modern day cell phones and mobile devices can easily handle most Profile-1 codestreams. Systems that implement BPJ2K01.10 will have resources commensurate with Profile-1 at a minimum. For this reason all decoders must be Profile-1 compliant (see section 5.1). Profile-1 will be the most common type of JPEG 2000 codestream encountered with some NR profile codestreams being present as well. It is expected that NR profile codestreams will be generated by LVSD compliant systems (Appendix G) and possibly by SPJE (Appendix H) and TPJE (Appendix F) systems.

The most common causes for a codestream to be NR profile and not Profile-1 compliant are the tile size and LL resolution constraints. LVSD compliant systems are very likely to create codestreams with tile sizes greater than 1,024 x 1,024 samples. This will likely be true even for hardware-based encoders. It is not difficult to encode larger tiles in hardware and there are proven performance advantages to having larger tile sizes. It is also possible that systems may not generate enough wavelet transform levels to result in a LL subband size of less than 128 x 128 coefficients. In general encoders *should* strive to make the LL subband no larger than 64 x 64 or 32 x 32 samples. JPEG 2000 coding efficiency and access to multiple resolution versions of an image require this.

To better understand why an encoder would need to choose large tile sizes let us consider compression of large images. For very large images it may be necessary to have R6 and lower resolution levels (R7, R8 and up) available to achieve an overview look of the entire image. This will require a combination of large tile size and increased number of wavelet transform levels. For example, consider an image 100,000 x 100,000 pixels (10 Gpixels) in size with 10,000 x 10,000 tiles (100 tiles total). At R6 the LL resolution is 156 x 156 samples for each tile. The total size of the R6 image is therefore 1,560 x 1,560 samples. At R8 the LL resolution becomes 39 x 39 samples for each tile and the total R8 image size is 390 x 390 samples. Six, seven or eight levels might be a reasonable point at which to stop the wavelet transform. Clearly the number of transform levels in this example needed to be greater than five (R5).

Including the proper number of wavelet transform levels is not enough. It is also necessary to ensure that the tiles are sufficiently large. If we had limited the tile size in the example to 1,024 x 1,024 samples and tried to form eight wavelet transform levels (to support an R8 image), the LL resolution in each tile would be a mere 4 x 4 samples. This LL resolution is far too small to be efficiently coded. Furthermore, the additional overhead present in the compressed codestream to support a much larger number of tiles (9,604 tiles vs. 100 tiles) is an unnecessary waste of bits. Systems that handle very large images will typically need to handle very large tiles and therefore must be able to generate and read NR profile codestreams.

Appendix D NSIF Preferred JPEG 2000 Encoding (NPJE)

A BPJ2K01.10 decoder is expected to be able to correctly decode *any ISO/IEC 15444-1 Profile-1* codestream contained in an NSIF file that conforms to the CLEVEL constraints of the system. However, imagery generated/compressed within a large distribution system that includes several levels of collection systems (encoders), libraries/distributors (transcoders), and end users (decoders) is expected to follow more restrictive recommendations, which ensure adequate scalability without resorting to recompression. This section describes one compression and reformatting/repackaging methodology recommended for use within the NSIF standards.

D.1 NPJE Overview

NPJE is recommended for all original image providers in NSIF when performing the initial first-stage compression of collected imagery. Certain sensor types and their concept of operations might require or elect to use other encodings defined in other appendices within this profile. The NPJE encoding, however, *shall* be understood and decoded by all decoders compliant with any other preferred encoding(s) contained within this standard subject the constraints of their systems (all NPJE files are ISO/IEC 15444-1 Profile-1 compliant). Encoders compliant to other encodings contained within this profile are *not* required to generate NPJE compliant codestreams. It is hoped, however, that all encoders will be able to create an NPJE version of their data, perhaps at a reduced quality or resolution.

D.1.1 NPJE Encoding Choices

The following JPEG 2000 parameter choices are recommended for the NPJE preferred encoding.

<u>To enable quality scalability</u>:

- Layer-Resolution-Component-Position (L-R-C-P) progression order should be used with enough quality layers to meet the quality goals (this recommendation includes 19 layers for visually lossless and 20 layers for numerically lossless, of which all layers leading up to and including the final quality requirement for the original image provider should be included).
- The layers are defined to have a diverse set of qualities to support every user from the radiometric/MASINT user all the way to the communication-constrained war-fighter.
- On imagery types for which visual exploitation is the primary function, JPEG 2000 lossy compression using the 9-7I filter yields the most efficient compression and produces the best image quality for a given target bitrate. The overall bitrate for these imagery types should be high enough to meet the quality requirements of the application.
- On imagery types for which radiometric exploitation – requiring extreme numerical accuracy – is the primary function, JPEG 2000 lossless compression using the 5-3R filter is used.

<u>To enable resolution scalability</u>:

- Five levels of wavelet decomposition are performed to ensure that six resolution levels (R0 – R5) can be accessed from the codestream.

<u>To facilitate ease of chipping and parsing</u>:

- Images are tiled with JPEG 2000 at a tile size of 1,024 x 1,024.
- Each tile is self-contained (i.e., one tile-part per tile).
- TLM markers are used to facilitate parsing of individual tiles from the compressed file.
- PLT markers are used to facilitate parsing of individual packets from the compressed file.

<u>To establish uniformity in the JPEG 2000 codestreams produced</u>:

- JPEG 2000 code block size is 64 x 64.
- Maximal precincts (i.e., no spatial segmentation of subbands) are used.
- Image offsets (XOsiz and YOsiz) and tile offsets (XTOsiz and YTOsiz) are set to zero.

Following the above recommendations ensures the following:

- Mutual compatibility and consistency among codestreams that are produced by different data providers.
- Both quality and resolution scalability are enabled, thereby eliminating the need to decompress and recompress (i.e., secondary dissemination can be accomplished merely by parsing).
- Ease of chipping and parsing and localized access into the image without the need for decompression and recompression.

D.1.2 JPEG 2000 Repackaging

Repackaging allows a JPEG 2000 image to be transmitted with lower image quality, lower resolution, fewer components or reduced spatial area, without expansion and recompression. An overview of how this occurs is provided in Appendix A . Repackaging an NPJE codestream will maintain many of the compression parameters presented in this section. However, some repackaging operations might require changes in the compression parameters. In particular, repackaging may eliminate higher quality layers and/or eliminate higher resolution data sets. Furthermore, chipping may require image offsets to vary. Once this occurs, not all of the recommended layers and/or resolutions will be available in the repackaged file. Otherwise, the repackaging recommendations stay true to the compression recommendations.

When a JPEG 2000 codestream is created via repackaging, a few of the parameters shown in Table D-5 through Table D-17 may have a wider range than the values shown for NPJE source image encoding. The parameters where this value broadening are allowed are Xsiz, Ysiz, XOsiz, YOsiz, XRsiz, YRsiz, Ltlm, Ztlm, N_{Levels}, and N_{Layers}. See Appendix A for more information on how specific repackaging operations modify these values.

D.2 NPJE Codestream Structure

Compressed tile data should appear in the JPEG 2000 codestream in raster index order without omission or repetition as shown in Figure D-1. For each tile in the NPJE codestream there is one tile-part. This guarantees that all data for all components within a tile are located in contiguous bytes within the codestream. Figure D-4 illustrates the ordering of compressed data within a single tile.

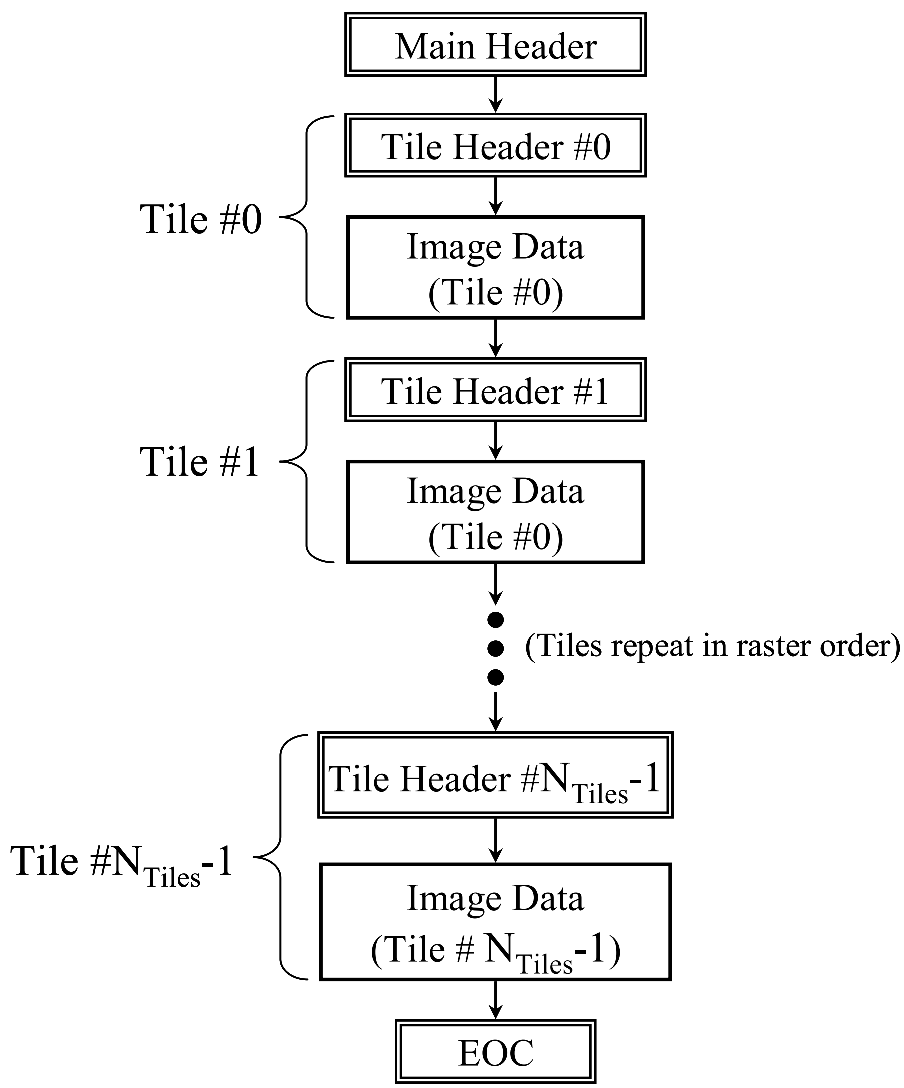

Figure D-1 High-level layout of NPJE JPEG 2000 compressed file

The JPEG 2000 main header should contain markers and marker segments as shown in Figure D-2.

The QCC and COM marker segments are optional marker segments. The COM marker segment allows informative information to be placed within a JPEG 2000 codestream. The QCC marker segment should be used in special cases where the QCD of the main header is inappropriate for a particular component. Such situations can occur with spectral data where the different spectral bands have widely varying dynamic range. The COC marker segment is not recommended for this encoding. It is anticipated that the COC marker segment will be made optional in a future version of NPJE. The marker segments in Figure D-2 may be placed in *any* order, except for the SIZ marker segment. The JPEG 2000 standard requires that the SIZ marker segment immediately follow the start of codestream (SOC) marker. The arrangement of marker segments in Figure D-2 is simply one of many different possible orderings.

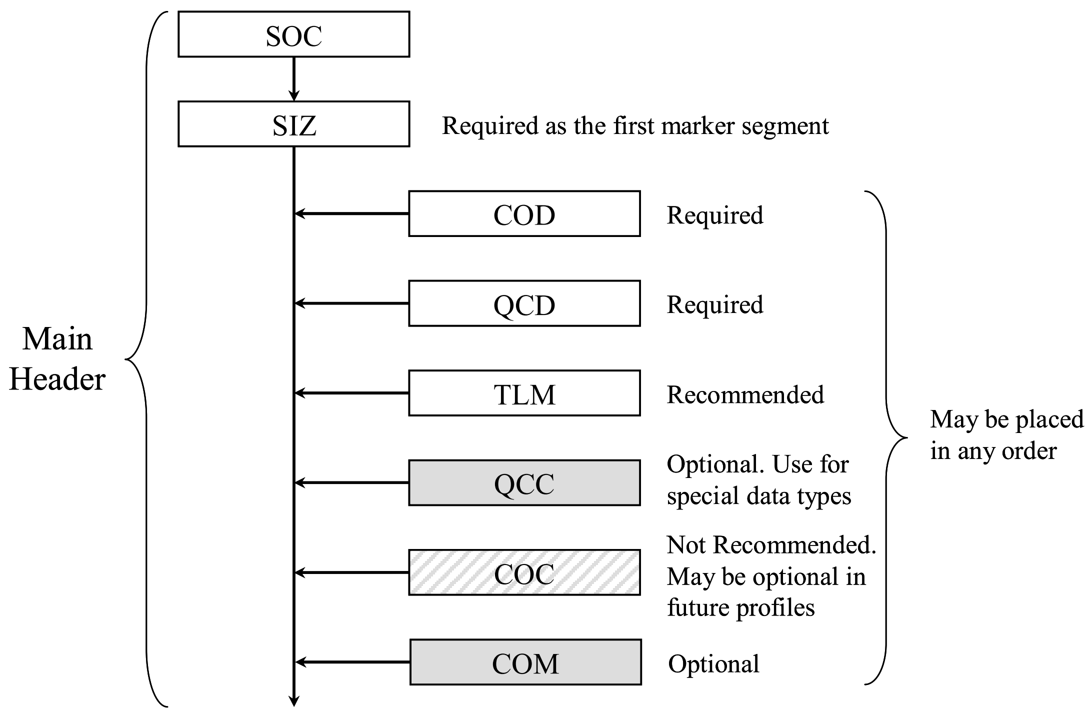

Figure D-2 Layout of JPEG 2000 main header for NPJE

The usage of the COM, QCD and QCC marker segments in a tile header is optional. For tiles with small dimensions, performing the recommended number of wavelet decompositions (N_{Levels}) can lead to empty or one-dimensional wavelet subbands. Usage of a QCD for these tiles might be appropriate to alter their quantization. Similarly, a tile might possess very different pixel statistics than other tiles in the image and changing the quantization parameters might be needed. The QCC marker segment can be used for similar reasons on a specific image component. It can be used in special cases where the QCD of the main (or tile) header is inappropriate for a particular component. Such situations may occur with spectral data where the different spectral bands have widely varying dynamic range.

The COD and COC marker segments are not recommended for use in tile headers in this encoding. It is anticipated that the COD and COC marker segments will be made optional in the tile headers in a future version of NPJE.

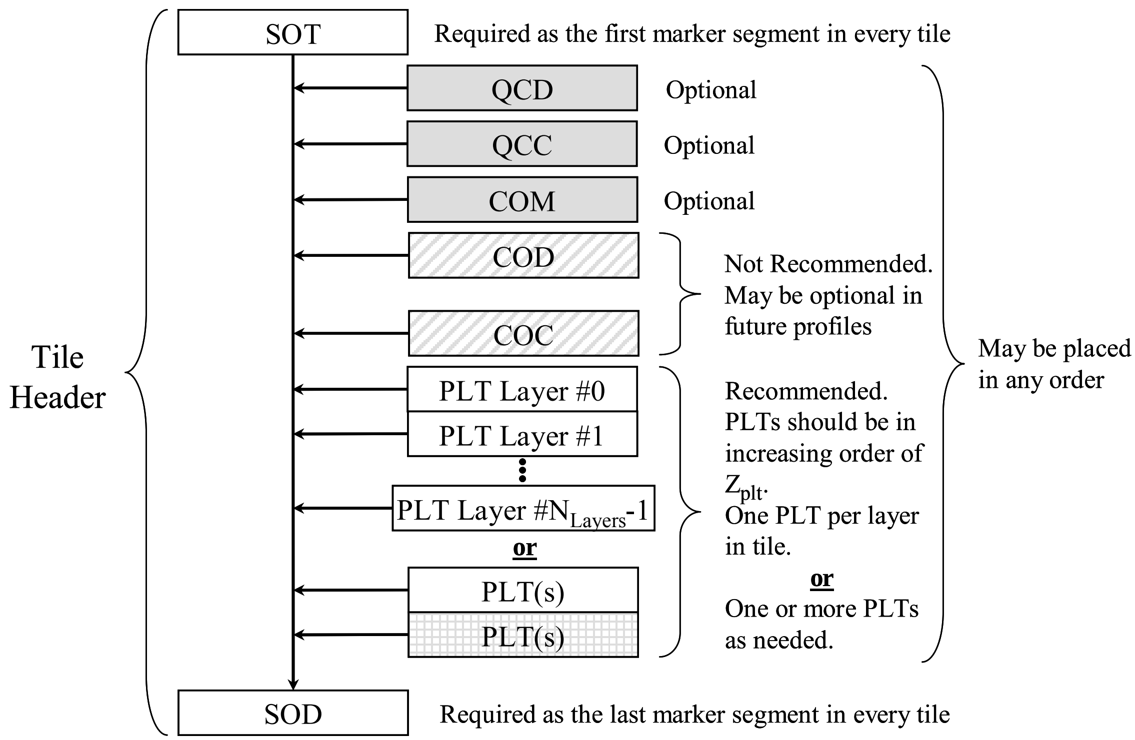

Figure D-3 Layout of a single JPEG 2000 tile header for NPJE

The JPEG 2000 tile header (for each tile) should contain markers and marker segments as shown in Figure D-3. Again the ordering of marker segments within a tile header is unimportant. The figure shows one possible arrangement. Note that in the BPJ2K01.00 version of this document, a PLT marker segment was recommended for each layer in an NPJE codestream. This recommendation has been relaxed in BPJ2K01.10. Implementers are free to use one or more PLT marker segments and not necessarily one per codestream layer. A single PLT marker segment can be used to contain all packet lengths provided the number of packet lengths is not too great. If the number of packets in the codestream is very large, more than one PLT might be needed due to marker segment length restrictions.

Decoders must therefore be able to properly handle one or more PLT marker segments. This is not a new requirement since a codestream can potentially have one or more layers (remember decoders must support ISO/IEC 15444-1 Profile-1 codestreams). Encoders now have the option of using one or more PLT marker segments as they see fit. It is suggested that encoders either use one PLT marker segment per layer, or one PLT marker segment for all layers. This will help to avoid some confusion, but there might be occasions where an intermediate number of PLT marker segments are either necessary or practical.

The JPEG 2000 image bits or bitstream (i.e. non-header data) in each tile should be arranged in Layer- Resolution-Component-Position (L-R-C-P) ordering as shown in Figure D-4. Within the L-R-C-P Bitstream ordering, data is first organized by increasing quality layer. Within each quality layer the data is arranged in order of increasing resolution. Within a given resolution level the data is arranged in component order. If precincts are used during encoding, the data within each layer resolution-component is ordered by precinct in raster order.

The JPEG 2000 codestream is terminated with an end of codestream (EOC) marker. The values with which to populate the aforementioned markers and marker segments are described in the Table D-1 through Table D-17.

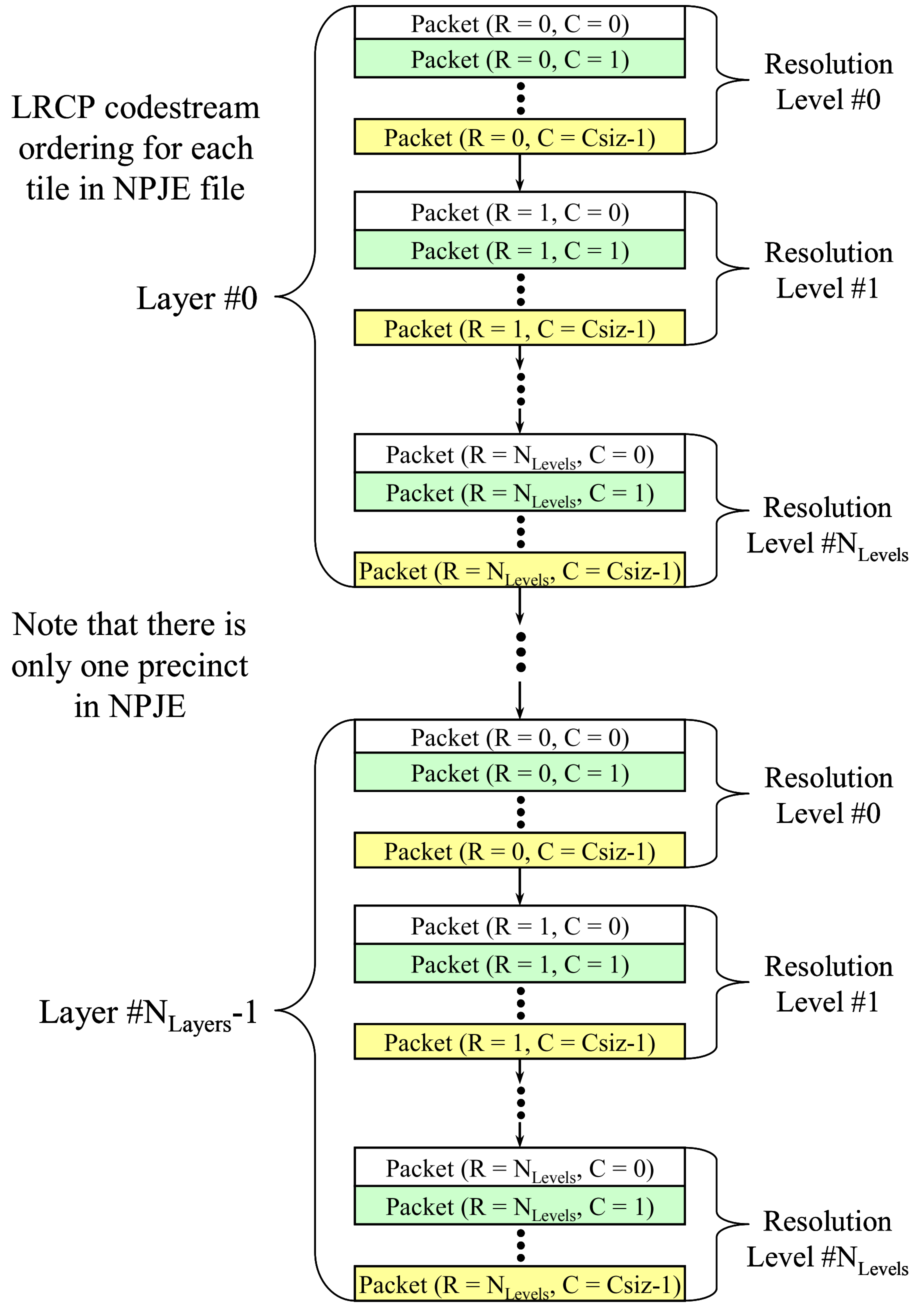

Figure D-4 Layout of bitstream for a single JPEG 2000 tile

D.3 NPJE Header Information

Table D-1 through Table D-3 indicate which headers (main or tile-part) various markers and marker segments appear in for NPJE.

Table D-1 NPJE JPEG 2000 Codestream structure (15444-1 Annex A.3)

Marker or Marker Segment	Size (bits)	Contents	Notes
Main Header	variable	See Table D-2	Single header for the entire image at the start of the JPEG 2000 codestream.
Tile Header	variable	See Table D-3	One tile header per tile. Each tile header is followed by the data packets for that tile. Tiles appear in the codestream in raster index order.
EOC	16	0xFFD9	End of codestream.

Table D-2 NPJE JPEG 2000 Main header contents (15444-1 Annex A.3)

Marker or Marker Segment	Size (bits)	Contents	Notes
SOC	16	0xFF4F	Start of codestream marker.
SIZ	$320 + 24 \cdot Csiz$	See Table D-9	Image and tile size.
COD	112	See Table D-10	Coding style default. 112 bits using max precincts (recommended).
QCD	9-7I: 296, or $56 + 48 \cdot N_{Levels}$ 5-3R: 168, or $48 + 24 \cdot N_{Levels}$	See Table D-11	Quantization default. Computed numbers assume recommended five levels of decomposition ($N_{Levels} = 5$), otherwise use formula. Note that "no quantization" is used with the 5-3R wavelet and "scalar quantization expounded" is used with the 9-7I wavelet.

Table D-2 NPJE JPEG 2000 Main header contents (15444-1 Annex A.3)			
Marker or Marker Segment	**Size (bits)**	**Contents**	**Notes**
QCC	9-7I (Csiz < 257): 304, or $64 + 48 \cdot N_{Levels}$ 9-7I (Csiz ≥ 257): 312, or $72 + 48 \cdot N_{Levels}$ 5-3R (Csiz < 257): 176, or $56 + 24 \cdot N_{Levels}$ 5-3R (Csiz ≥ 257): 184, or $64 + 24 \cdot N_{Levels}$	See Table D-12	Quantization component, optional marker segment. Computed numbers assume recommended five levels of decomposition ($N_{Levels} = 5$), otherwise use formula. Note that "no quantization" is used with the 5-3R wavelet and "scalar quantization expounded" is used with the 9-7I wavelet.
TLM	variable	See Table D-13	Tile-part lengths main header. Describes length of every tile-part in the codestream. The values of each tile-part length are the same values given by the Psot parameter in the SOT marker segment.
COM	variable	See Table D-16	Comment, optional marker segment. Shall not contain information necessary to properly interpret or decode the codestream.

Table D-3 NPJE Tile header contents (15444-1 Annex A.3)			
Marker or Marker Segment	**Size (bits)**	**Contents**	**Notes**
SOT	96	See Table D-6	Start of tile-part.
QCD	9-7I: 296, or $56 + 48 \cdot N_{Levels}$ 5-3R: 168, or $48 + 24 \cdot N_{Levels}$	See Table D-11	Quantization default. If present in the tile header, this marker segment overrides the QCD specified in the main header. See notes in Table D-2.

Table D-3 NPJE Tile header contents (15444-1 Annex A.3)			
Marker or Marker Segment	**Size (bits)**	**Contents**	**Notes**
QCC	9-7I (Csiz < 257): 304, or $64 + 48 \cdot N_{Levels}$ 9-7I (Csiz ≥ 257): 312, or $72 + 48 \cdot N_{Levels}$ 5-3R(Csiz < 257): 176, or $56 + 24 \cdot N_{Levels}$ 5-3R(Csiz ≥ 257): 184, or $64 + 24 \cdot N_{Levels}$	See Table D-12	Quantization component, optional marker segment. If present in the tile header, this marker segment overrides the QCD specified in the main header. See notes in Table D-2.
PLT	variable	See Table D-14	Packet length, tile-part header. The inclusion of this marker segment allows individual packets to be identified in the codestream without requiring that all packet headers be read first.
COM	variable	See Table D-16	Comment, optional marker segment. See above.
SOD	16	0xFF93	Start of data marker.

D.4 Markers and Marker Segments Limits and NPJE

The following markers and marker segments are defined in Table 7-1. Table D-4 describes the recommended usage of each marker, where it is required (Req.), not allowed (NA), optional (opt.), recommended (Rec.) or not recommended (NR) and if there are any restrictions or dependencies. There are only three places that a marker can be present, the main header, tile header, or the bitstream. The values column in the following tables give the full ISO/IEC 15444-1 ranges for comparison.

Table D-4 Marker and marker segment requirements within a NPJE codestream					
Marker	**Value**	**Main header**	**Tile Header**	**bit-stream**	**Restriction/Dependencies**
SOC	0xFF4F	Req.	NA	NA	Required as first marker in main header and therefore the codestream.
SOT	0xFF90	NA	Req.	NA	Required as the first marker in each tile part.
SOD	0xFF93	NA	Req.	NA	Last marker of each tile part header.

Table D-4 Marker and marker segment requirements within a NPJE codestream					
Marker	**Value**	**Main header**	**Tile Header**	**bit-stream**	**Restriction/Dependencies**
EOC	0xFFD9	NA	NA	Req.	Required as last marker in the codestream.
SIZ	0xFF51	Req.	NA	NA	Required as second marker segment in the main header.
COD	0xFF52	Req.	NR	NA	Required in main header, and no more than one in the first tile-part header of a given tile. Indicates the usage of SOP and EPH.
COC	0xFF53	NR	NR	NA	No more than one COC per any given component within the main header or in the first tile-part header of a given tile.
RGN	0xFF5E	Opt.	Opt.	NA	May appear in the main header or first tile-part header of a given tile. When used in the main header it applies to one component across all tiles except those with an RGN marker. In a tile-part header it applies to one component in that tile.
QCD	0xFF5C	Req.	Opt.	NA	One and only one required in the main header. May be at most one in the first tile-part header of a given tile.
QCC	0xFF5D	Opt.	Opt.	NA	No more than one per any given component in the main header or first tile-part header of a given tile.
POC	0xFF5F	NA	NA	NA	The POC marker segment is not allowed in NPJE.
TLM	0xFF55	Rec.	NA	NA	Recommended, there may be multiple TLM marker segments in the main header.
PLM	0xFF57	NA	NA	NA	The PLM marker segment is not allowed in NPJE. The PLT marker segment is recommended instead.
PLT	0xFF58	NA	Rec.	NA	Recommended, there may be multiple PLT marker segments per tile. Must appear in any tile-part header before the packets whose lengths they describe.
PPM	0xFF60	NA	NA	NA	The PPM marker segment is not allowed in NPJE. Packet headers should be distributed throughout the codestream.
PPT	0xFF61	NA	NR	NA	If a PPT marker segment appears in a tile part header, all packet headers for the given tile shall follow. The PPT marker segment must appear in a tile-part header before the packets whose headers are contained in the PPT appear.
SOP	0xFF91	NA	NA	NR	May be used in front of each packet, shall not be used unless indicated in the proper COD marker segment. Whether or not an

Table D-4 Marker and marker segment requirements within a NPJE codestream					
Marker	**Value**	**Main header**	**Tile Header**	**bit-stream**	**Restriction/Dependencies**
					SOP marker segment is used for a given packet, Nsop must be incremented for each packet in the codestream. If packet headers are moved into a PPT or PPM marker segment, the SOP marker segments may appear immediately before the packet bodies in the bitstream.
EPH	0xFF92	NR	NR	NR	Shall not be present unless indicated in the proper COD marker segment. If EPH marker segments are signaled, they must appear for every packet header. If the packet headers are moved into a PPM or PPT marker segment, the EPH markers shall appear after the packet headers in the PPM or PPT marker segments.
CRG	0xFF63	NR	NA	NA	Only one CRG may appear in the main header and it applies for all tiles.
COM	0xFF64	Opt.	Opt.	NA	Repeat as many times as desired in the main or tile-part headers. This marker segment shall have no effect on decoding the bitstream.

D.4.1 Delimiting Markers and Marker Segments

The delimiting markers shall be present in all JPEG 2000 compressed imagery. Each delimiting marker must be present in a compliant JPEG 2000 codestream. A codestream shall have only one SOC and EOC marker, and at least one tile-part. Each tile-part has one SOT and one SOD marker.

Table D-5 Start of codestream (15444-1 Annex A.4.1)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
SOC	16	0xFF4F	0xFF4F	Start of codestream.

Table D-6 Start of tile-part (15444-1 Annex A.4.2)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
SOT	16	0xFF90	0xFF90	Start of tile part marker segment.
Lsot	16	10	10	Length of marker segment in bytes.
Isot	16	0 – 65,534	Tile index	Tile index in raster order starting at index 0.

Table D-6 Start of tile-part (15444-1 Annex A.4.2)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
Psot	32	0, 14 – (2^{32} –1)	Length of tile-part 0, 14 – (2^{32} –1)	The length in bytes from the beginning of SOT marker segment of the tile-part to the end of the data of that tile-part. It is recommended a Psot of 0 be replaced by the actual tile length when a JPEG 2000 codestream is incorporated into NSIF. If Psot = 0 is maintained in an NSIF file, the current tile part will be interpreted to extend to the end of the current NSIF image segment.
Tpsot	8	0 – 254	0	Tile-Part index.
Tnsot	8	0 – 255	1	0 = Number of tile-parts of this tile in the codestream is not defined in this header 1 – 255 number of tile-parts of this tile in the codestream

Table D-7 Start of data marker (15444-1 Annex A.4.3)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
SOD	16	0xFF93	0xFF93	Start of data marker.

Table D-8 End of codestream marker (15444-1 Annex A.4.4)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
EOC	16	0xFFD9	0xFFD9	End of codestream marker.

D.4.2 Fixed Information Marker Segment

This marker segment includes information required to properly decode the image. There shall be a SIZ marker segment in the main header immediately after the SOC marker segment.

Table D-9 Image and tile size (15444-1 Annex A.5.1)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
SIZ	16	0xFF51	0xFF51	Image and tile size marker segment.
Lsiz	16	41 – 49,190	38 + 3·Csiz	Length of marker segment in bytes.
Rsiz	16	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010 0000 0000 0000 0011 0000 0000 0000 0100	0000 0000 0000 0010 (Profile-1)	A value of zero indicates that the full standard capabilities described by ISO/IEC IS15444-1 are required. Values of three and four are reserved for DCI and shall not be used. See below*.
Xsiz	32	1 – (2^{32}-1)	image width (1 – (2^{31}-1))	Width of reference grid. Equal to the image width with no image offset into the reference grid.
Ysiz	32	1 – (2^{32}-1)	image height (1 – (2^{31}-1))	Height of reference grid. Equal to the image height with no image offset into the reference grid
XOsiz	32	0 – (2^{32}-2)	0	Horizontal offset from the origin of the reference grid to the left side of the image area.
YOsiz	32	0 – (2^{32}-2)	0	Vertical offset from the origin of the reference grid to the top of the image area.
XTsiz	32	1 – (2^{32}-1)	1,024	Width of one reference tile with respect to the reference grid.
YTsiz	32	1 – (2^{32}-1)	1,024	Height of one reference tile with respect to the reference grid.
XTOsiz	32	0 – (2^{32}-2)	0	Horizontal offset from the origin of the reference grid to the left edge of the first tile.
YTOsiz	32	0 — (2^{32}-2)	0	Vertical offset from the origin of the reference grid to the top edge of the first tile.
Csiz	16	1 – 16,384	Nbands	The number of components in the image.

Table D-9 Image and tile size (15444-1 Annex A.5.1)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
$Ssiz^i$	8	0000 0000 – 1010 0101	Unsigned: 0 – 31 Signed: 128 – 159	0xxx xxxx Unsigned data 1xxx xxxx signed data x000 0000 – x010 0101 bit depth of data = value + 1
$XRsiz^i$	8	1 – 255	1	Horizontal separation of a sample of the i^{th} component with respect to the reference grid.
$YRsiz^i$	8	1 – 255	1	Vertical separation of a sample of the i^{th} component with respect to the reference grid.

* Two new values for the Rsiz parameter have been added in ISO/IEC 15444-1 Amendment 1, 2006. These values were assigned to the Digital Cinema Initiative to indicate which DCI profile (2K or 4K) the codestream adheres to. See the document "Digital Cinema System Specification**,** Version 1.1"**,** April 12, 2007 for further details. NPJE implementations *shall not* set Rsiz = 0000 0000 0000 0011 or Rsiz = 0000 0000 0000 0100.

D.4.3 Functional Marker Segments

The functional marker segments define what parameters were used in the compression of a given tile or an image. These marker segments apply to the entire tile when in the tile header and to the image when in the main header. Markers in the tile header supersede markers in the main header.

Table D-10 Coding style default (15444-1 Annex A.6.1)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
COD	16	0xFF52	0xFF52	Coding style default marker segment.
Lcod	16	Maximal precincts: Lcod = 12 User-defined precincts: Lcod = 13 + N_{Levels}	12	Length of marker segment in bytes.
Scod	8	xxxx x000 – xxxx x111 (Defined in Table 7-8)	0000 0000	Entropy coder with maximum precinct size. No SOP marker segments shall be used. EPH marker shall not be used.
SGcod	32	Defined below		
Progression order	8	0000 0000 – 0000 0100	0000 0000 (as defined by Table 7-9)	Layer-resolution level-component-position progression provides SNR progression.

Table D-10 Coding style default (15444-1 Annex A.6.1)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
Number of layers (N_{Layers})	16	1 – 65,535	19 for visually lossless (9-7I filter) 20 for numerically lossless (5-3R filter)	The producer of a NPJE file should include all of the recommended layers to meet the system's quality or bitrate requirements within the specified 20 layers, including the non-truncated highest bitrate layer. More layers may be added as required to meet specific bitrate or quality requirements.
Multiple component transform	8	0000 0000 – 0000 0001	0000 0000 or 0000 0001	0000 0000 = No component transform used. 0000 0001 = Component transform used.
SPcod	Variable	Defined below		
Number of decomposition levels (N_{Levels})	8	0 – 32	5	There should be 5 decomposition levels.
Code-block width	8	xxxx 0000 – xxxx 1000	0000 0100	The width of a code-block should be 64, xcb = value + 2.
Code-block height	8	xxxx 0000 – xxxx 1000	0000 0100	The height of a code-block should be 64, ycb = value + 2.
Code-block style	8	xx00 0000 – xx11 1111 (Defined in Table 7-10)	0000 0000 baseline 0000 0001 if arithmetic coder bypass is used	No selective arithmetic coding bypass in baseline. No reset of context probabilities on coding pass boundaries. No termination on each coding pass. No vertically causal context. No predictable termination. No segmentation symbols are used.
Transformation	8	0000 0000 – 0000 0001	0000 0001 0000 0000	5-3 reversible filter for numerically lossless applications (i.e., radiometric data). 9-7 irreversible filter for applications that do not require lossless data.
Precinct size	Variable	0000 0000 – 1111 1111	NA	Not present. The precincts have been defined as maximum size by Scod.

The QCD marker is used in the main header to indicate quantization step-sizes valid for all tile-parts. The QCD marker is required in the main header – the values in this marker segment in the main header are used for all tiles that do not override these values via a tile-specific QCD in that tile's header.

Table D-11 Quantization default (15444-1 Annex A.6.4)

Parameter	Size (bits)	Values	NPJE	Notes
QCD	16	0xFF5C	0xFF5C	Quantization default marker segment.
Lqcd	16	Scalar quantization derived: Lqcd = 5 No quantization: $Lqcd = 4 + 3 \cdot N_{Levels}$ Scalar quantization expounded: $Lqcd = 5 + 6 \cdot N_{Levels}$	For 5-3R wavelet: Lqcd = 19 For 9-7I wavelet: Lqcd = 35	Length of this marker segment in bytes. For the 5-3R wavelet, no quantization is used. For the 9-7I wavelet, expounded quantization is used. See Sqcd.
Sqcd	8	Table 7-16	0100 0000 0100 0010	With 5-3 reversible filter: 2 guard bits and no quantization. With 9-7 irreversible filter: 2 guard bits and scalar expounded quantization.
SPqcd[i]	8 (5-3R) 16 (9-7I)	Table 7-16	Table 7-17 Table 7-18	With 5-3R wavelet. With 9-7I wavelet.

Table D-12 Quantization component (15444-1 Annex A.6.5)

Parameter	Size (bits)	Values	NPJE	Notes
QCC	16	0xFF5D	0xFF5D	Quantization component marker segment.
Lqcc	16	For Csiz < 257 Scalar quantization derived: $Lqcd = 6$ No quantization: $Lqcd = 5 + 3 \cdot N_{Levels}$ Scalar quantization expounded: $Lqcd = 6 + 6 \cdot N_{Levels}$ For Csiz ≥ 257 Scalar quantization derived: $Lqcd = 7$ No quantization: $Lqcd = 6 + 3 \cdot N_{Levels}$ Scalar quantization expounded: $Lqcd = 7 + 6 \cdot N_{Levels}$	For Csiz < 257 For 5-3R wavelet: Lqcd = 20 For 9-7I wavelet: Lqcd = 36 For Csiz ≥ 257 For 5-3R wavelet: Lqcd = 21 For 9-7I wavelet: Lqcd = 37	Length of this marker segment in bytes.
Cqcc	8 16	0 – 255 (8 bits, Csiz < 257) 0 – 16,383 (16 bits, Csiz ≥ 257)	0 – 255 0 – 16,383	Component index to which this marker segment applies.
Sqcc	8	Table 7-16	0100 0000 0100 0010	With 5-3 reversible filter: 2 guard bits and no quantization. With 9-7 irreversible filter: 2 guard bits and scalar expounded quantization.
SPqcc[i]	8 16	Table 7-16	Table 7-17 Table 7-18	With 5-3R wavelet With 9-7I wavelet

D.4.4 Pointer Marker Segments

The pointer markers segments are used to gain quick access to desired data for parsing, chipping, and decoding. The marker segments define either lengths of a data set or pointers to the start of a data set. The tile-part length marker (TLM) segment has the same length information as the start of tile marker segments in each tile-part, but this information is collected up front in the main header. This marker segment can be used to quickly access and chip a given tile or set of tiles in a compressed image. Within each tile header is recommended the packet length marker (PLT) which will then allow quick access to a given packet or group of packets for parsing or decoding data at a given quality or resolution level.

Table D-13 Tile-part lengths, main header (15444-1 Annex A.7.1)

Parameter	Size (bits)	Values	NPJE	Notes
TLM	16	0xFF55	0xFF55	Tile-part lengths marker segment.
Ltlm	16	$Ltlm = \begin{cases} & ST & SP \\ 4+2\cdot N_{tpm} & 0 & 0 \\ 4+3\cdot N_{tpm} & 1 & 0 \\ 4+4\cdot N_{tpm} & 2 & 0 \\ 4+4\cdot N_{tpm} & 0 & 1 \\ 4+5\cdot N_{tpm} & 1 & 1 \\ 4+6\cdot N_{tpm} & 2 & 1 \end{cases}$ N_{tpm} = number of tile-parts in this TLM marker segment	ST = 0, SP = 1 Ltlm = $4 + 4\cdot N_{tiles}$ (8 – 65,535)	Length of this marker segment in bytes. For this encoding the number of tile parts in the marker segment is equal to the number of tiles.
Ztlm	8	0 – 255	0	There shall be only one TLM per compressed codestream.
Stlm	8	x000 xxxx – x110 xxxx	0100 0000 (See Table 7-22)	ST = 0; only one tile-part per tile and the tiles are in index order without omission or repetition. SP = 1; Ptlm parameter has 32 bits.
$Ttlm^i$	0 if ST = 0 8 if ST = 1 16 if ST = 2	Tiles in order 0 – 254 0 – 65,534	N/A	Tiles appear in order, so no bits are used.
$Ptlm^i$	16 if SP = 0 32 if SP = 1	14 – 65,535 $14 – (2^{32} – 1)$	$14 — (2^{32}-1)$	The length, in bytes, from the beginning of the SOT marker of the i^{th} tile-part to the end of the codestream data for that tile-part. There should be one Ptlm for every tile-part; therefore, there is one Ptlm for every tile.

Table D-14 Packet lengths, tile-part header (15444-1 Annex A.7.3)

Parameter	Size (bits)	Values	NPJE	Notes
PLT	16	0xFF58	0xFF58	Packet length, tile-part header, marker segment.
Lplt	16	4 – 65,535	4 – 65,535	Length of this marker segment in bytes.

Table D-14 Packet lengths, tile-part header (15444-1 Annex A.7.3)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
Zplt	8	0 – 255	0 – 18, 19*	Index of this marker segment relative to all other PLT marker segments in the current header. * Applications shall use no more than one PLT per layer (they may use fewer) unless marker segment length constraints require more.
Iplt^i	8 bits repeated as necessary See ISO/IEC 15444-1:2004 Table A-36		0xxx xxxx	Signals that the next seven bits are the last bits indicating the length of the i^{th} packet.
		0000 0000 – 1111 1111	1xxx xxxx	Signals that there are further bits to be included after these next seven bits are included as part of the packet length.
			x000 0000 – x111 1111	7 bits of packet length. All bits associated with the length of the ith packet are concatenated and right justified in the order in which they appear. The packet length shall include the packet header.

D.4.5 Informational Marker Segments

The informational marker segments are not required for decoding but may assist in the decoding, parsing, or displaying of the data. Component registration (CRG) allows each component to be registered to each other for proper display and exploitation. The CRG marker segment is not recommended for use. The Comment marker (COM) allows for the unstructured data to be included into the file. Its use is optional. Implementers may place informative information within a COM marker segment such as software version of the JPEG 2000 codec. *No data shall be placed in a comment marker segment that is necessary to decode or properly interpret a JPEG 2000 codestream.* Neither metadata that might be used to properly interpret and exploit the compressed imagery nor any information that might be used to improve JPEG 2000 decoder performance shall be placed within a COM marker segment.

Table D-15 Component registration (15444-1 Annex A.9.1)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
CRG	16	0xFF63	0xFF63	Component registration marker segment.
Lcrg	16	6 – 65,534		Length of this marker segment in bytes.
$Xcrg^i$	16	0 – 65,535		Value of horizontal offset in units of 1/65536 of the horizontal separation $XRsiz^i$, for the i^{th} component
$Ycrg^i$	16	0 – 65,535		Value of vertical offset in units of 1/65536 of the vertical separation $YRsiz^i$, for the i^{th} component

Table D-16 Comment (15444-1 Annex A.9.2)				
Parameter	**Size (bits)**	**Values**	**NPJE**	**Notes**
COM	16	0xFF64	0xFF64	Comment marker segment.
Lcom	16	5 – 65,535		Length of this marker segment in bytes.
Rcom	16	0 = General binary 1 = General Latin (IS 8859-15:1999)		Registration values. Indicates type of data in marker segment.
$Ccom^i$	8	0 – 255		Data.

D.4.6 Recommended Compression Rate Control

Rate control is intimately tied with layer formation. The rate control mechanism decides how many and which bytes are allocated per layer in the compressed file. The JPEG 2000 standard leaves this matter up to the implementer, but a detailed rate control procedure can be found in the J2K_Guide (see NTB website URL in Section 2). Two common forms of rate control that are used with tiled imagery are:

- Full image rate control
- Individual tile rate control

Full image rate control produces a more uniform image quality by allowing busy tiles (those with fine detail) to contribute more bytes to the compressed file at the expense of less busy tiles. The aggregate bitrate for the entire image is still maintained, but the tile-by-tile bitrate can change. In general, full image bitrate control consumes more memory during compression since compressed data from each tile must be retained to form the final layers. Full image rate control is not conducive to independent tile processing for this same reason.

Individual tile rate control allocates the same number of bytes to each tile regardless of that tile's scene content. This allows every tile to be layered and processed independently. If the scene variation between neighboring tiles varies dramatically, then there might be noticeable image quality differences between tiles at the same layer. For example, tile boundaries may become visible to the user. However, this problem will not manifest itself at visually lossless bitrates.

D.4.7 Recommended Layers

The following quality layers and their applications are recommended for NPJE applications. These 20 recommended layers are appropriate for grayscale (i.e. single-band) imagery and were selected to achieve flexibility to meet multiple requirements of multiple applications. It is recommended that the highest layer that is included in each compressed file from the original data provider be based on the highest quality requirement for that particular system. For example, if the collection system has a requirement to meet 1.0 bpp then the system should include layers 0 – 9 in the original compression but if a system has a requirement for lossless then it should include all 20 layers (0 – 19). If visually lossless quality can be achieved at 3.5 bpp, then layers 0 – 18 should be included and layer 19 would not need to exist. Some systems may change the exact bitrates and number of layers to meet application requirements or quality requirements. For example, a system may have a requirement to meet communication limits that would require exactly 1.4 bpp, which is not currently on the recommended layers.

Note that the layer target bitrates are defined for the full resolution R0 image. When extracting a reduced resolution data set from a JPEG 2000 compressed file, the effective bitrate can be much higher than the target bitrate shown for R0. Therefore, it is important to include a good number of layers at target bitrates well below the useful range for R0. For example, in an 11-bit image, the layer 0 target bitrate of 0.03125 bpp might produce very poor image quality for R0 but it can yield very satisfactory quality for R5. Thus, the first several layers at the very low bitrates are important for quality scalability when extracting reduced resolution data sets.

Table D-17 NPJE Suggested Layers	
Layer	**Bits per pixel per band (bpppb)**
19 (5-3R filter only)	Numerically lossless
18 (Last layer for 9-7I filter)	3.5 (visually lossless)
17	2.8
16	2.3
15	2.0
14	1.7
13	1.5
12	1.3
11	1.2
10	1.1
9	1.0
8	0.9
7	0.8
6	0.7
5	0.6
4	0.5
3	0.25
2	0.125
1	0.0625
0	0.03125

Appendix E Exploitation Preferred JPEG 2000 Encoding (EPJE)

The EPJE preferred encoding is a reordering of NPJE that facilitates rapid access to a variety of resolution levels. EPJE can be created from an original NPJE or it could be compressed directly from the collection system (but NPJE is generally preferred when collecting data). It is particularly beneficial when accessing very large images (i.e., greater than 40K x 40K images) at lower resolutions. EPJE is a formatting alternative and produces identical decoded data to NPJE.

E.1 EPJE Overview

The need for EPJE arose out of the use of ELT (Electronic Light Table) applications. At reduced resolution roaming and viewing, ELT performance with NPJE compliant codestreams was sub-optimal. The issue was determined to be due to disk access patterns when pulling reduced resolutions out of an NPJE file. Figure E-1 illustrates the layout of an NPJE codestream on disk. Each tile codestream is contained in one contiguous tile-part and the bitstream is ordered in the L-R-C-P progression. Note that in this figure the JPEG 2000 notation for resolution (where R0 is the low-resolution and R5 is highest resolution for NPJE) is used.

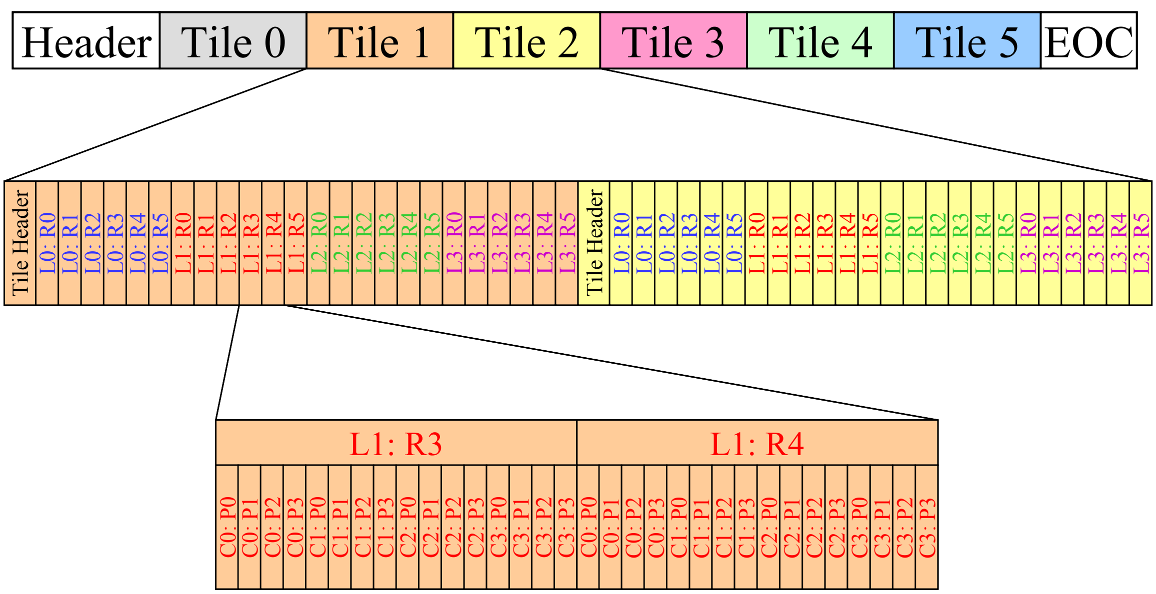

Figure E-1 Layout of NPJE codestream on disk

If we wish to extract the lowest resolution version of the image (R0 in JPEG 2000 terminology or R5 in R-set terminology for an NPJE file) we end up pulling pieces of the file from all over the disk. If the image is very large and we are trying to roam through the image at a high rate of speed at reduced resolution, disk access patterns and I/O bottlenecks will limit the ELT application's performance. This is illustrated in Figure E-2.

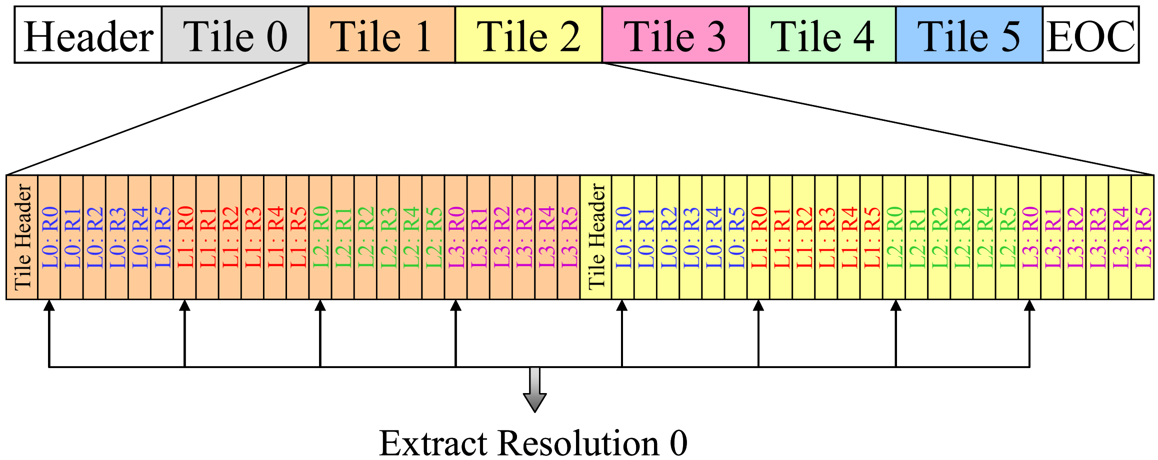

Figure E-2 Extracting resolution 0 from an NPJE codestream

For small imagery and casual viewing, this type of disk access pattern will not matter. It is the nature of ELT applications, however, to view large portions of large images at reduced resolution. This causes the application to pull pieces of the compressed bitstream from many different tiles. Furthermore, the ELT applications are required to quickly roam across large images at a high rate of speed. Thus the file access must be very quick. EPJE was developed as a way to reorder an NPJE file to facilitate ELT performance. Figure E-3 shows the reordering of the above NPJE file into EPJE.

As shown in Figure E-3, EPJE uses tile-parts to collect together like resolutions throughout the file and place them into one contiguous area. Furthermore, the R-L-C-P progression order is used to facilitate this ordering. Extracting a given resolution out of the EPJE file is now more readily accomplished. Far fewer disk seeks are required and the desired codestream data tends to be located in a few contiguous regions on the disk. This greatly improves disk and I/O performance in ELT applications.

It does not take long to realize, however, that tile-based spatial chipping is more difficult under EPJE. With NPJE codestreams tile-based chipping was a trivial task. With EPJE, tile-based chipping will require several disk seeks and reading of data to reassemble a tile codestream. This tradeoff is not a detriment for ELT users, however, since they are willing to wait for a spatial chipping operation to be performed. The improved ELT application performance is much more important.

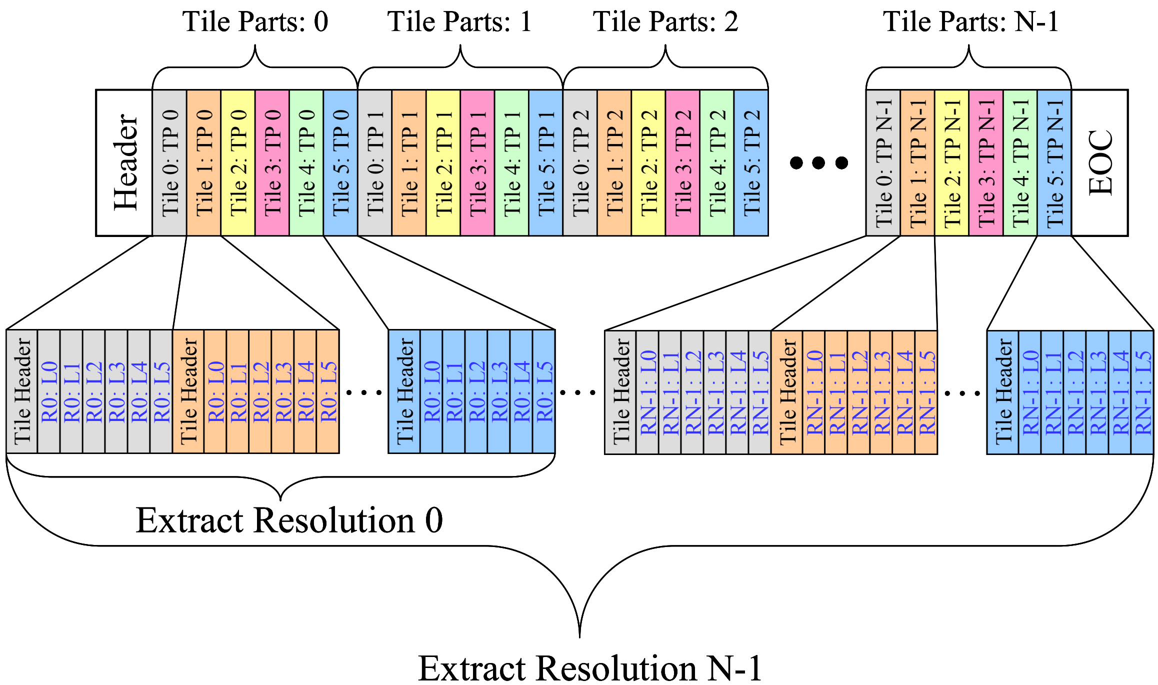

Figure E-3 Extracting resolutions from an EPJE codestream

E.1.1 EPJE Encoding Choices

As alternate formats with identical decodable content, EPJE and NPJE use the same structural basis for transform and coding. All NPJE restrictions that impact bitstream content (such as tile size and offset, codeblock size, number of decomposition levels, number of layers, and maximal size precincts) apply to EPJE as well. The key feature of EPJE is a resolution progression ordering divided into resolution-based tile-parts (i.e. a separate tile-part for each resolution level in a tile). Tile-parts are ordered so like resolution data is contiguous and within each resolution level tiles appear in raster order.

To handle the location information for the increased number of tile-parts, multiple TLMs are permitted in EPJE (See ISO 15444-1 Appendix A.7.1 for TLM format). Only one PLT appears in each tile-part header, containing packet lengths only for layers within the tile-part. Optional markers allowed in NPJE are also allowed in EPJE. Optional tile header markers (such as QCD) appear only in the first tile-part header for the given tile.

E.1.2 JPEG 2000 Repackaging

Repackaging allows a JPEG 2000 image to be transmitted with lower image quality, lower resolution, fewer components or reduced spatial area, without expansion and recompression. An overview of how this occurs is provided in Appendix A . The same caveats apply for EPJE as for NPJE. Repackaging will preserve most compression choices, but some codestream constructs might be altered. Removal of quality layers or resolutions as well as chipping operations can alter the codestream such that it will no longer strictly conform to EPJE. Appendix A provides more detail on how repackaging of codestreams can affect a JPEG 2000 codestream or file.

When a JPEG 2000 codestream is created via repackaging, a few of the parameters have a wider EPJE range than the values shown for source image encoding. The parameters where this value broadening are allowed are Xsiz, Ysiz, XOsiz, YOsiz, XRsiz, YRsiz, N_{Levels}, N_{Layers} and TNsot. It is important to note here that transcoding from NPJE to EPJE and back should lead to *no degradation or change* in decoded pixel values as long as all bitstream data is retained.

E.2 EPJE Codestream Structure

Within EPJE, each tile is broken into multiple tile-parts based upon resolution. Within each resolution level, compressed tile data should appear in the JPEG 2000 codestream in raster index order without omission or repetition as shown in Figure E-4. For each tile there are multiple tile-parts (one for each resolution level). This guarantees all data for a given resolution level is contiguous within the codestream, but any individual tile contains several separate pieces. Figure E-6 illustrates the ordering of compressed data within a single tile-part.

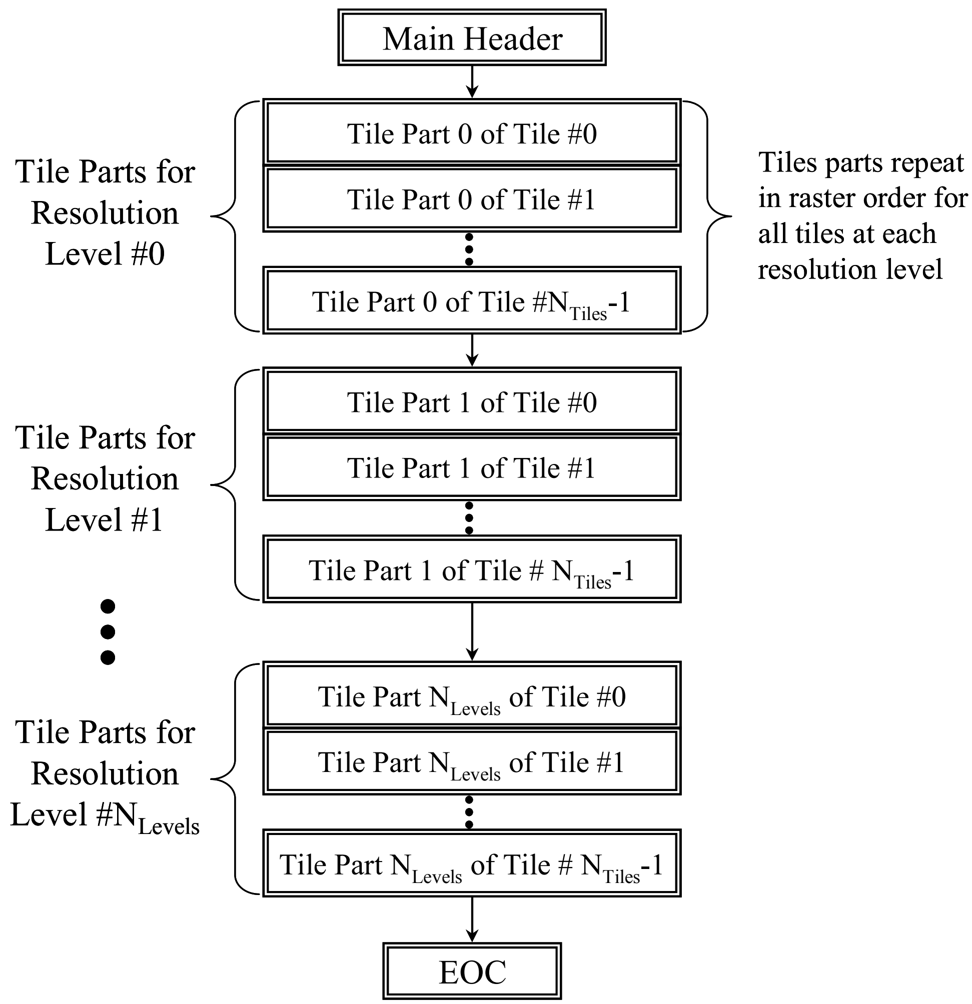

Figure E-4 High-level layout of EPJE JPEG 2000 compressed file

The JPEG 2000 main header should contain markers and marker segments as shown in Figure E-5. The only difference from the NPJE main header is that multiple TLMs are allowed (and probably required for big datasets). TLMs are required, not optional in EPJE. The SIZ, QCD and QCC marker segments for EPJE are identical to those in NPJE. The QCD and QCC marker segments may be used in the main header, or optionally in the first tile-part header for any tile. The COM marker segment is optional and may be used to convey non-critical information. No marker segments other than COM, SOT, PLT, and SOD shall appear in later tile-parts of a tile.

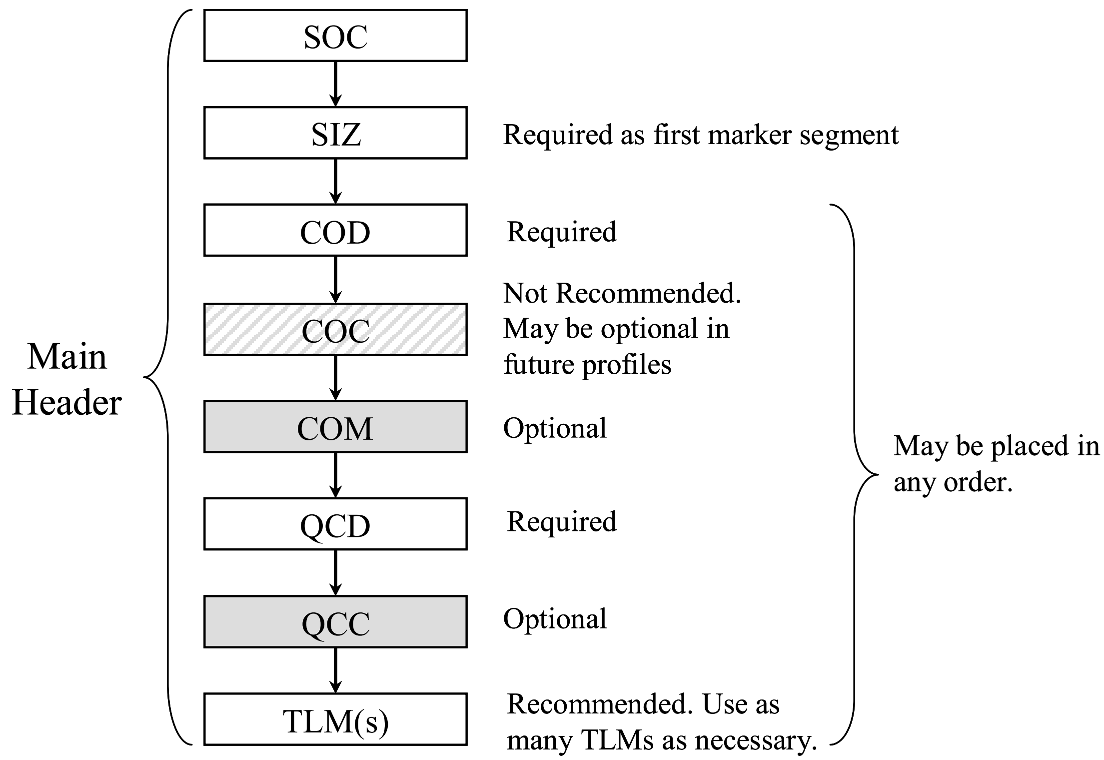

Figure E-5 Layout of JPEG 2000 main header for EPJE

The JPEG 2000 tile-part header for tile-part *i,* should contain markers and marker segments and be arranged as shown in Figure E-6. The JPEG 2000 bitstream in each tile should be arranged in Resolution- Layer-Component-Position (R-L-C-P) ordering as shown in Figure E-6. Tile specific COD and COC marker segments are not recommend for use. If they are necessary, they can be used as long as they do not change packet numbering/ordering in any way. The bitstream data is then separated into tile-parts at the resolution boundaries. Within each resolution (tile-part) the data is arranged in order of increasing quality layer. Within a given quality layer the data is arranged in component order. The JPEG 2000 codestream is terminated with an end of codestream (EOC) marker.

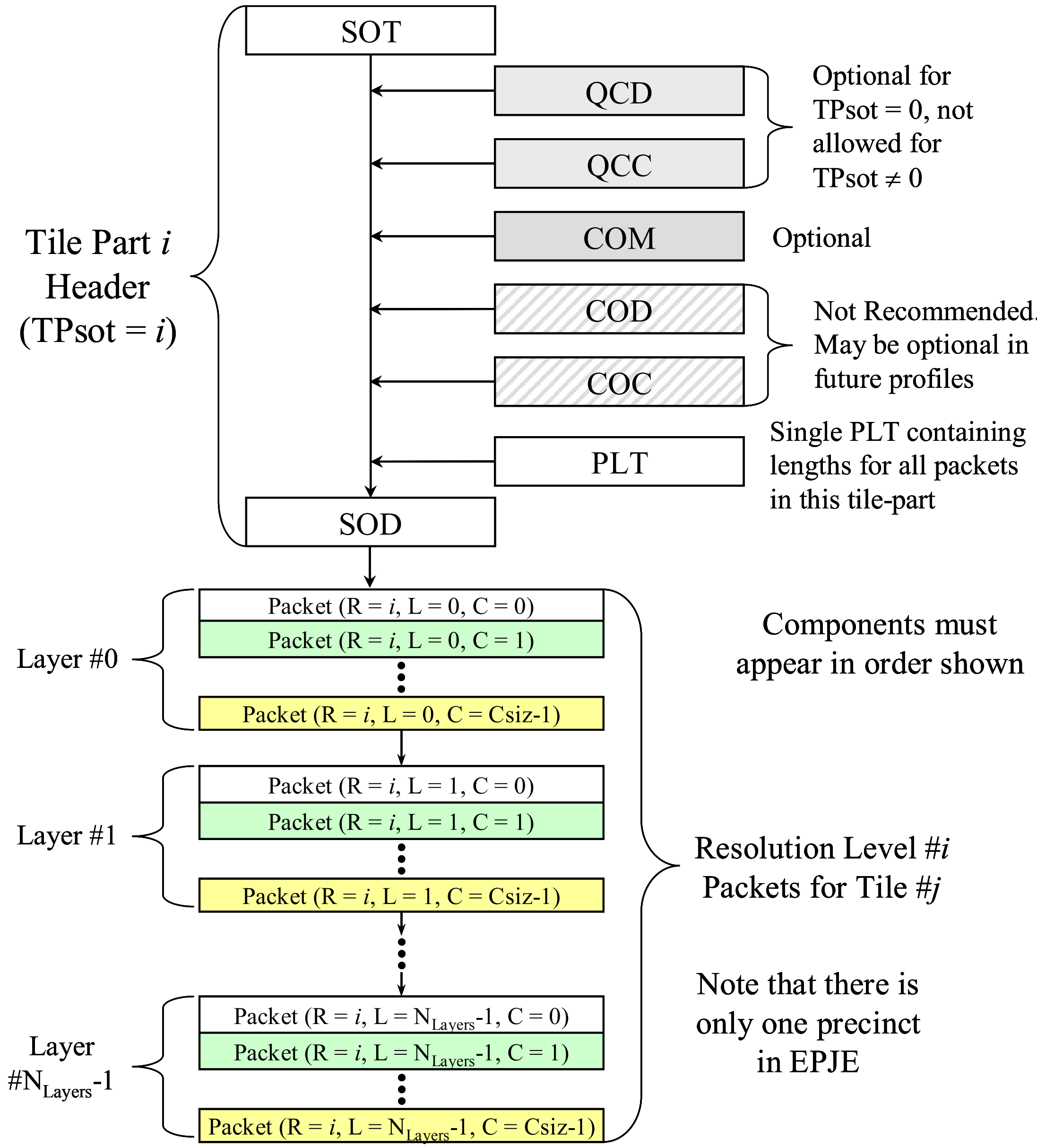

Figure E-6 Layout of a single JPEG 2000 tile-part header for EPJE

The values with which to populate the aforementioned markers and marker segments are described in Table E-1 through Table E-8 as well as tables from Appendix D called out in the text. Since EPJE is a repackaging of NPJE, the bulk of the contents of the marker segments are the same.

E.3 EPJE Header Information

Table E-1 EPJE JPEG 2000 Codestream structure (15444-1 Annex A.3)

Marker or Marker Segment	Size (bits)	Contents	Notes
Main Header	variable	Table E-2	Single header for the entire image at the start of the JPEG 2000 codestream.
Tile-Part Header	variable	Table E-3	Multiple tile-part headers per tile. Each tile-part header is followed by the data packets for that tile-part. Tile-parts of equal resolution appear contiguously in the codestream in raster index order.
EOC	16	0xFFD9	End of codestream.

Table E-2 EPJE JPEG 2000 Main header contents (15444-1 Annex A.3)

Marker or Marker Segment	Size (bits)	Contents	Notes
SOC	16	0xFF4F	Start of codestream marker.
SIZ	$320 + 24 \cdot Csiz$	Table D-9	Image and tile size marker segment.
COD	112	Table E-6	Coding style default marker segment (112 bits using max precincts).
QCD	9-7I: 296, or $56 + 48 \cdot N_{Levels}$ 5-3R: 168, or $48 + 24 \cdot N_{Levels}$	Table D-11	Quantization default marker segment. Computed numbers assume five levels of decomposition ($N_{Levels} = 5$), otherwise use formula. Note that "no quantization" is used with the 5-3R wavelet and "scalar quantization expounded" is used with the 9-7I wavelet.
TLM(s)	variable	Table E-7	Tile-part lengths main header. Describes length of every tile-part in the codestream. The values of each tile-part length are the same values given by the Psot parameter in the SOT marker segment.

Table E-2 EPJE JPEG 2000 Main header contents (15444-1 Annex A.3)			
Marker or Marker Segment	**Size (bits)**	**Contents**	**Notes**
QCC	9-7I (Csiz < 257): 304, or $64 + 48 \cdot N_{Levels}$ 9-7I (Csiz ≥ 257): 312, or $72 + 48 \cdot N_{Levels}$ 5-3R(Csiz < 257): 176, or $56 + 24 \cdot N_{Levels}$ 5-3R(Csiz ≥ 257): 184, or $64 + 24 \cdot N_{Levels}$	Table D-12	Quantization component marker segment, optional. Computed numbers assume five levels of decomposition ($N_{Levels} = 5$), otherwise use formula. Note that "no quantization" is used with the 5-3R wavelet and "scalar quantization expounded" is used with the 9-7I wavelet.
COM	variable	Table D-16	Comment, optional marker segment. Shall not contain information necessary to properly interpret or decode the codestream.

Table E-3 EPJE Tile header contents (15444-1 Annex A.3)			
Marker or Marker Segment	**Size (bits)**	**Contents**	**Notes**
SOT	96	Table E-5	Start of tile-part marker segment.
QCD	9-7I: 296, or $56 + 48 \cdot N_{Levels}$ 5-3R: 168, or $48 + 24 \cdot N_{Levels}$	Table D-11	Quantization default marker segment, optional. Only allowed in the first tile-part of any tile (TPsot=0). If present, this marker segment overrides the QCD specified in the main header. See notes in Table D-2.

Table E-3 EPJE Tile header contents (15444-1 Annex A.3)			
Marker or Marker Segment	**Size (bits)**	**Contents**	**Notes**
QCC	9-7I (Csiz < 257): 304, or $64 + 48 \cdot N_{Levels}$ 9-7I (Csiz ≥ 257): 312, or $72 + 48 \cdot N_{Levels}$ 5-3R(Csiz < 257): 176, or $56 + 24 \cdot N_{Levels}$ 5-3R(Csiz ≥ 257): 184, or $64 + 24 \cdot N_{Levels}$	Table D-12	Quantization component marker segment, optional. Only allowed in the first tile-part of any tile (TPsot=0). If present, this marker segment overrides the QCD specified in the main header. See notes in Table D-2.
PLT	variable	Table E-8	Packet length marker segment, tile-part header. The inclusion of this marker segment allows individual packets to be identified in the codestream without requiring that all packet headers be read first. There will be only one PLT marker segment per tile-part.
COM	variable	Table D-16	Comment, optional marker segment. See above.
SOD	16	0xFF93	Start of data marker.

E.4 Markers and Marker Segments Limits and EPJE

The following markers and marker segments are defined in Table 7-1. Table E-4 describes the recommended usage of each marker, where it is required (Req.), not allowed (NA), optional (Opt.), recommended (Rec.) or not recommended (NR) and if there are any restrictions or dependencies. There are only three places that a marker can be present, the main header, tile header, or the bitstream. The values column in the following tables give the full ISO/IEC 15444-1 ranges for comparison.

Table E-4 Marker and marker segment requirements within an EPJE codestream					
Marker	**Value**	**Main header**	**Tile-part Header**	**bitstream**	**Restriction/Dependencies**
SOC	0xFF4F	Req.	NA	NA	Required as first marker in main header and therefore the codestream.
SOT	0xFF90	NA	Req.	NA	Required as the first marker in each tile part.
SOD	0xFF93	NA	Req.	NA	Last marker of each tile part header.
EOC	0xFFD9	NA	NA	Req.	Required as last marker in the codestream.
SIZ	0xFF51	Req.	NA	NA	Required as second marker segment in the main header.
COD	0xFF52	Req.	NR	NA	Required in main header, and no more than one in the first tile-part header of a given tile. Indicates the usage of SOP and EPH. If used in a tile-part header only the filter type may change.
COC	0xFF53	NR	NR	NA	No more than one COC per any given component within the main header or in the first tile-part header of a given tile. If used in a tile-part header only the filter type may change.
RGN	0xFF5E	Opt.	Opt.	NA	May appear in the main header or first tile-part header of a given tile. When used in the main header it applies to one component across all tiles except those with an RGN marker. In a tile-part header it applies to one component in that tile.
QCD	0xFF5C	Req.	Opt.	NA	One and only one required in the main header. May be at most one in the first tile-part header of a given tile.
QCC	0xFF5D	Opt.	Opt.	NA	No more than one per any given component in the main header or first tile-part header of a given tile.
POC	0xFF5F	NA	NA	NA	POC marker segment is not allowed within EPJE.
TLM	0xFF55	Rec.	NA	NA	Required, there may be multiple TLM marker

Table E-4 Marker and marker segment requirements within an EPJE codestream					
Marker	**Value**	**Main header**	**Tile-part Header**	**bitstream**	**Restriction/Dependencies**
					segments in the main header.
PLM	0xFF57	NA	NA	NA	The PLM marker segment is not allowed within EPJE.
PLT	0xFF58	NA	Rec.	NA	Optional, there may be multiple PLT marker segments per tile. Must appear in any tile-part header before the packets whose lengths they describe.
PPM	0xFF60	NA	NA	NA	The PPM marker segment is not allowed within EPJE.
PPT	0xFF61	NA	NA	NA	PPT marker segment is not allowed within EPJE.
SOP	0xFF91	NA	NA	NR	May be used in front of each packet, shall not be used unless indicated in the proper COD marker segment. Whether or not an SOP marker segment is used for a given packet, Nsop must be incremented for each packet in the codestream.
EPH	0xFF92	NR	NR	NR	Shall not be present unless indicated in the proper COD marker segment. If EPH marker segments are signaled, they must appear for every packet header.
CRG	0xFF63	NR	NA	NA	Only one CRG may appear in the main header and it applies for all tiles.
COM	0xFF64	Opt.	Opt.	NA	Repeat as many times as desired in the main or tile-part headers. This marker segment shall have no effect on decoding the bitstream.

E.4.1 Delimiting Markers and Marker Segments

The delimiting markers shall be present in all JPEG 2000 compressed imagery. Each delimiting marker must be present in a compliant JPEG 2000 codestream. A codestream shall have only one SOC and EOC marker, and at least one tile-part for each resolution level in a tile. Each tile-part has one SOT (see Table E-5) and one SOD marker. The SOC marker in EPJE is the same as the SOC in NPJE (see Table D-5). The SOD and EOC markers are formatted as for NPJE (see Table D-7 and Table D-8).

Table E-5 Start of tile-part (15444-1 Annex A.4.2)

Parameter	Size (bits)	Values	EPJE	Notes
SOT	16	0xFF90	0xFF90	Start of tile part marker segment.
Lsot	16	10	10	Length of this marker segment in bytes.
Isot	16	0 – 65,534	Tile index	Tile index in raster order starting at index 0.
Psot	32	0, 14 – (2^{32} –1)	Length of tile-part 0, 14 – (2^{32} –1)	The length in bytes from the beginning of SOT marker segment of the tile-part to the end of the data of that tile-part. It is recommended a Psot of 0 be replaced by the actual tile length when a JPEG 2000 codestream is incorporated into NSIF. If Psot = 0 is maintained in an NSIF file, the current tile part will be interpreted to extend to the end of the current NSIF image segment.
TPsot	8	0 – 254	0 – N_{Levels}	Tile-Part index.
TNsot	8	0 – 255	N_{Levels}+1	0 = Number of tile-parts of this tile in the codestream is not defined in this header 1 – 255 number of tile-parts of this tile in the codestream

E.4.2 Fixed Information Marker Segment

This marker segment includes information required to properly decode the image. There shall be a SIZ marker segment in the main header immediately after the SOC marker segment. The EPJE SIZ marker segment is identical to the NPJE SIZ marker segment shown in Table D-9.

E.4.3 Functional Marker Segments

The functional marker segments define what parameters were used in the compression of a given tile or an image. These marker segments apply to the entire tile when in the first tile-part header for the given tile and to the image when in the main header. Markers in the first tile-part header for a given tile supersede markers in the main header. Markers which override the main header defaults can only appear in the first tile-part for a tile, not in later tile-parts.

Table E-6 Coding style default (15444-1 Annex A.6.1)

Parameter	Size (bits)	Values	EPJE	Notes
COD	16	0xFF52	0xFF52	Coding style default marker segment.
Lcod	16	Maximal precincts: Lcod = 12 User-defined precincts: Lcod = 13 + N_{Levels}	12	Length of this marker segment in bytes.
Scod	8	0000 0000 – 0000 0111 (Defined in Table 7-8)	0000 0000	Entropy coder with maximum precinct size. No SOP marker segments shall be used. EPH marker shall not be used.
SGcod	32	Defined below		
Progression order	8	0000 0000 – 0000 0100	0000 0001 (Defined in Table 7-9)	R-L-C-P progression order.
Number of layers (N_{Layers})	16	1 – 65,535	19 for visually lossless (9-7I) 20 for numerically lossless (5-3R)	The producer of an EPJE file should include all of the recommended layers to meet the system's quality or bit rate requirements within the specified 20 layers, including the non-truncated highest bit rate layer. More layers may be added as required to meet specific bit rate or quality requirements.
Multiple component transform	8	0000 0000 – 0000 0001	0000 0000 or 0000 0001	0000 0000 = No component transform used. 0000 0001 = Component transform used.
SPcod	Variable	Defined below		
Number of decomposition levels (N_{Levels})	8	0 – 32	5	There should be 5 decomposition levels.
Code-block width	8	0000 0000 – 0000 1000	0000 0100	The width of a code-block should be 64, xcb = value + _2.
Code-block height	8	0000 0000 – 0000 1000	0000 0100	The height of a code-block should be 64, ycb = value + 2.

Table E-6 Coding style default (15444-1 Annex A.6.1)				
Parameter	**Size (bits)**	**Values**	**EPJE**	**Notes**
Code-block style	8	0000 0000 – 0011 1111 (Defined in Table 7-10)	0000 0000 baseline 0000 0001 if arithmetic coder bypass is used	No selective arithmetic coding bypass in baseline. No reset of context probabilities on coding pass boundaries. No termination on each coding pass. No vertically causal context. No predictable termination. No segmentation symbols are used.
Transformation	8	0000 0000 – 0000 0001	0000 0001 0000 0000	5-3 reversible filter for numerically lossless applications (i.e., radiometric data). 9-7 irreversible filter for applications that do not require lossless data.
Precinct size	variable	0000 0000 – 1111 1111	NA	Not present. The precincts have been defined as maximum size by Scod.

The QCD marker is used in the main header to indicate quantization step-sizes valid for all tile-parts. The QCD marker is required in the main header. The values in this marker segment in the main header are used for all tiles that do not override these values via a tile-specific QCD in that tile's first tile-part header. The QCD and QCC marker segments for EPJE are identical to those for NPJE (see Table D-11 and Table D-12).

E.4.4 Pointer Marker Segments

The pointer markers segments are used to gain quick access to desired data for parsing, chipping, and decoding. The marker segments define either lengths of a data set or pointers to the start of a data set. The tile-part length marker (TLM) segment has the same length information as the start of tile marker segments in each tile-part, but this information is collected up front in the main header. This marker segment can be used to quickly access and chip a given tile or set of tiles in a compressed image. Within each tile header is recommended the packet length marker segment (PLT) which will allow quick access to a given packet or group of packets for parsing or decoding data at a given quality or resolution level. Each tile-part PLT will contain the packet lengths for those packets within the tile-part.

Table E-7 Tile-part lengths, main header (15444-1 Annex A.7.1)

Parameter	Size (bits)	Values	EPJE	Notes
TLM	16	0xFF55	0xFF55	Tile-part lengths marker segment.
Ltlm	16	$Ltlm = \left\{ \begin{array}{lcc} & ST & SP \\ 4+2\cdot N_{tpm} & 0 & 0 \\ 4+3\cdot N_{tpm} & 1 & 0 \\ 4+4\cdot N_{tpm} & 2 & 0 \\ 4+4\cdot N_{tpm} & 0 & 1 \\ 4+5\cdot N_{tpm} & 1 & 1 \\ 4+6\cdot N_{tpm} & 2 & 1 \end{array} \right.$ N_{tpm} = number of tile-parts in this TLM marker segment	ST = 2, SP = 1 $Ltlm = 4 + 6 \cdot N_{tpm}$ (8 – 65535)	Length of this marker segment in bytes. For this encoding, the number of tile parts in the marker segment may be restricted based upon the TLM marker length restrictions. It will be at most $(1+ N_{Levels})$ times the number of tiles.
Ztlm	8	0 – 255	0 – 255	Index of this TLM relative to other TLMs which appear in the main header. There may be multiple TLMs
Stlm	8	0000 0000 – 0110 0000	0110 0000	ST = 2; multiple tile-parts per tile, so each tile index must be listed; use 2 bytes. SP = 1; Ptlm parameter has 32 bits.
$Ttlm^i$	0 if ST = 0 8 if ST = 1 16 if ST = 2	Tiles in order 0 – 254 0 – 65,534	0 – 65,534	Tile index, for every tile-part in order as they appear in the file.
$Ptlm^i$	16 if SP = 0 32 if SP = 1	14 – 65,535 $14 – (2^{32} – 1)$	$14 – (2^{32}-1)$	The length, in bytes, from the beginning of the SOT marker of the i^{th} tile-part to the end of the codestream data for that tile-part. There should be one Ptlm for every tile-part.

Table E-8 Packet lengths, tile-part header (15444-1 Annex A.7.3)

Parameter	Size (bits)	Values	EPJE	Notes
PLT	16	0xFF58	0xFF58	Packet length, tile-part header, marker.
Lplt	16	4 – 65,535	4 – 65,535	Length of this marker segment in bytes (not including the marker).
Zplt	8	0 – 255	0	Index of this marker segment relative to all other PLT marker segments in the current header. There shall be one PLT per tile-part.
$Iplt^i$	8 bits repeated as necessary See ISO/IEC		0xxx xxxx	Signals that the next seven bits are the last bits indicating the length of the i^{th} packet.

Table E-8 Packet lengths, tile-part header (15444-1 Annex A.7.3)				
Parameter	**Size (bits)**	**Values**	**EPJE**	**Notes**
	15444-1:2004 Table A-36	0000 0000 – 1111 1111	1xxx xxxx	Signals that there are further bits to be included after these next seven bits are included as part of the packet length.
			x000 0000 – x111 1111	7 bits of packet length. All bits associated with the length of the ith packet are concatenated and right justified in the order in which they appear. The packet length shall include the packet header.

E.4.5 Informational Marker Segments

The CRG (component registration) marker segment is not recommended in EPJE. The COM (comment) marker segment is optional. The same caveats placed on the COM marker segment in section D.4.5 apply to the EPJE preferred encoding as well.

E.4.6 Recommended Compression Rate Control

EPJE levies no requirements on rate control beyond what applies to NPJE. Extra tile-part headers introduced by EPJE do not contribute to the rate computations.

E.4.7 Recommended Layers

EPJE uses the same recommended layering as NPJE (see Table D-17). If an EPJE file is being created directly from uncompressed data, then apply the NPJE layering recommendations. Otherwise, when converting data from NPJE to EPJE format, the NPJE layering will be maintained.

E.4.8 J2KLRA TRE Contents

When an EPJE codestream is included in an NSIF file, a J2KLRA TRE must also be included (see Table 8-3). The ORIG field of this TRE is used to indicate the type of formatting used in the codestream. EPJE codestreams are marked as either 2 (original encoding) or 3 (parsed data). Codestreams not strictly conforming to the EPJE format shall not use the values 2 or 3 in the ORIG field of the J2KLRA TRE.

Appendix F Tactical Preferred JPEG 2000 Encoding (TPJE)

The Tactical Preferred JPEG 2000 Encoding (TPJE) was created to facilitate all areas of airborne imagery compression ranging from future hardware-based, on-board, JPEG 2000 compression capabilities to the current software-based, ground-processing operations found in many of today's imaging architectures.

The TPJE addresses the specific JPEG 2000 compression parameters required to create an TPJE compliant codestream. This section also defines how an TPJE-compliant codestream is to be incorporated into a Basic Image Interchange Format (BIIF per ISO/IEC 12087-5:1998(E)), NATO Secondary Imagery Format (NSIF version 1.0 per STANAG 4545), or National Imagery Transmission Format (NITF version 2.1 per MIL-STD-2500C) file, if desired.

Like the other JPEG 2000 preferred encodings currently used by the imaging community (NPJE, EPJE, etc.), TPJE has been developed to be able to correctly decode any ISO/IEC 15444-1 Profile-1 compressed codestream contained in a BIIF, NSIF, or NITF file that conforms to the complexity level (CLEVEL) constraints of the image processing system being used.

F.1 TPJE Overview

The following discussion provides an overview of the JPEG 2000 compression parameter choices recommended for use by airborne imagery providers when performing the initial first-stage compression of collected imagery. Certain sensor types and their concept of operations might require or elect to use other preferred encodings defined in other appendices within this standard. TPJE, however, *shall* be understood and decoded by all decoders compliant to TPJE. Encoders compliant to other encodings contained within this standard are *not* required to generate TPJE compliant codestreams, though by default all NPJE and EPJE compliant codestreams will also adhere to the TPJE requirements. This is because TPJE is a superset of NPJE and EPJE. Note that, not all instantiations of TPJE are necessarily themselves NPJE or EPJE compliant. It is hoped, however, that all TPJE encoders will be able to create either an EPJE or NPJE version of their data, perhaps at a reduced quality or resolution.

F.1.1 TPJE Encoding Choices

The following JPEG 2000 parameter choices are recommended for TPJE.

<u>Support to Scalability</u>

- TPJE supports both the Layer-Resolution-Component-Position (L-R-C-P) and Resolution-Layer-Component-Position (R-L-C-P) progression orders
- Permits 1 to 20 Quality Layers for both lossy (includes visually lossless) and lossless compression

<u>Resolution Scalability</u>

- Permits 5 to 9 wavelet decomposition levels (produces 6 to 10 Resolution Levels)

<u>Support to Chipping and Parsing</u>

- Tile sizes can be 256x256, 512x512, or 1024x1024
- Each tile can be self-contained (i.e. one tile-part per tile) or contain up to 10 tile-parts per tile, as progression order permits
- Multiple TLM markers are permitted
- PLT marker segments are required, one or more are allowed. Applications shall use no more than one PLT per layer (they might use less) unless marker segment length constraints require more.

<u>JPEG 2000 Processing</u>

- JPEG 2000 code block sizes can be 32x32, 32x64, 64x32 or 64x64
- Maximal precincts (i.e., no spatial segmentation of subbands) are used
- Image offsets (XOsiz, YOsiz) and tile offsets (XTOsiz, YTOsiz) are always set to zero, even for spatially parsed (chipped) datasets. TPJE only supports tile-based chipping.

F.1.2 JPEG 2000 Repackaging

Repackaging, or parsing, allows a JPEG 2000-compressed image to be transmitted with lower image quality, coarser spatial resolution, fewer components (bands), and/or reduced spatial area (chipping), without expansion and recompression of the pixel data. Repackaging a TPJE codestream will maintain most of the compression recommendations presented in this section, however, several of these compression recommendations might be altered during dataset repackaging.

For example, repackaging may eliminate higher Quality Layers and/or eliminate finer spatial Resolution Levels from the parsed dataset. Furthermore, the parsing of spatial extent (chipping) is allowed, possibly reducing the number of pixels present in the chipped codestream. Once such parsing occurs, not all of the originally recommended Quality Layers, Resolution Levels, components, and/or pixels will be available in the repackaged file. Aside from these obvious differences, the repackaged codestream shall stay true to the compression recommendations provided in this TPJE definition.

When a TPJE-compliant JPEG 2000 codestream is created via the repackaging of an existing codestream, a few of the parameters shown in Table F-4 through Table F-15 shall have different ranges than those shown for the TPJE-compliant encoding of source imagery. The TPJE compression parameters potentially having altered value ranges are: Xsiz, Ysiz, XRsiz, YRsiz, Ltlm, Ztlm, N_{Levels}, N_{Layers} and TNsot.

Note that when a spatial chipping operation is performed, the XOsiz, YOsiz, XTOsiz, and YTOsiz parameters of a TPJE-compliant codestream are not altered to reflect the offset of the chipped scene from the origin of the original scene; these values are to remain set to zero. *This is possible because TPJE restricts tile sizes to powers of two and only tile-based chipping only is allowed*. Chipped scene offsets are instead recorded in the ICHIPB tagged record extension (TRE), as discussed in section F.8.2, when embedding the TPJE codestream in a BIIF, NSIF, or NITF file. Unfortunately, this rule means that spatially parsed, stand-alone, (not embedded in BIIF, NSIF, or NITF) TPJE-compliant codestreams will not retain any offset information.

TPJE-compliant codestreams may also be formed by JPEG 2000 transcoding operations. Transcoding refers to the reordering of the packets in an existing JPEG 2000 codestream. Traditionally, transcoding operations have been used to convert NPJE-compliant codestreams, which were based on the L-R-C-P progression order, to EPJE-compliant codestreams, which were based on the R-L-C-P progression order. Transcoding can also be used to convert TPJE-compliant codestreams into either NPJE- or EPJE-compliant codestreams, though care should be taken to ensure that the conversion process is possible. Implementations of specific transcoder logic are left to the implementer. For example, there may be reasons why a particular TPJE-compliant codestream could not be readily transcoded to a NPJE-compliant form. Careful study of this TPJE specification document will elucidate such potential pitfalls for the would-be developer.

F.2 TPJE Codestream Structure

Since TPJE supports NPJE, EPJE, and hybrids of these two encodings, Figure F-1 provides two representations of TPJE codestreams. Both files are shown in raster index order and represent a single-component image for the sake of simplicity.

F.3 TPJE Main Header

The following discussion describes the Main Header for TPJE-compliant codestreams.

F.3.1 TPJE Header Overview

The TPJE main header contains markers and marker segments as shown in Figure F-2 and enumerated below:

1. SOC (start of code stream)
2. SIZ (image and tile size) maker
3. COD (coding style default) marker
4. QCD (quantization default) marker for both lossy (including visually lossless) and numerically lossless cases
5. QCC (quantization component) marker is required if…
    a. The file is a multi-component data set

    -and-

    b. One or more of the components are handled (e.g. quantized) differently than the default QCD value (*For an overview of Main Header and Tile Header QCD and QCC Values, see section F.4.2.4*)
6. TLM (tile length) maker(s)
7. COM (optional information), optional

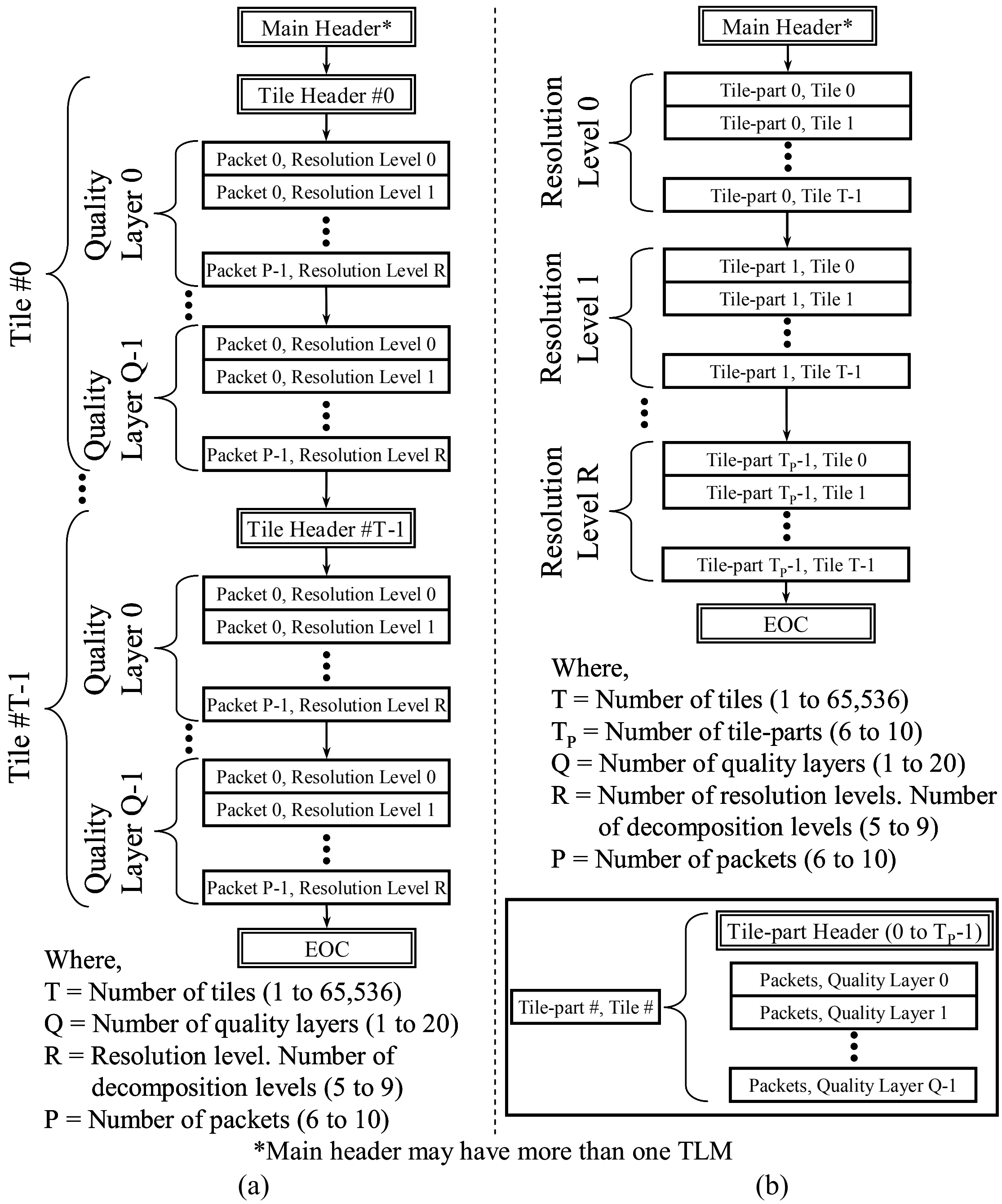

Figure F-1 High-Level View of the TPJE File. (a) Represents the appearance of the file when there is one tile-part per tile. (b) Represents the appearance of the file when there are multiple tile-parts per tile.

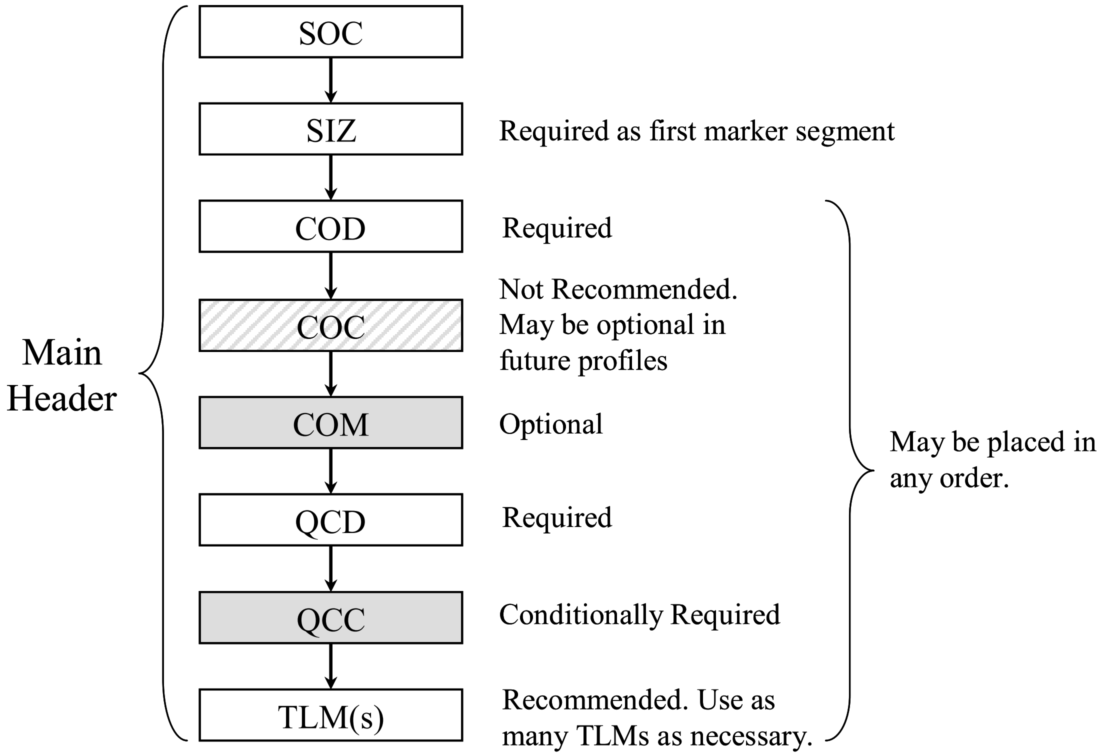

Figure F-2 Overview of the TPJE Main Header.

Note that, as shown in Figure F-2 above, the markers COD, QCD, QCC, TLM, and COM can appear in any order in the header; the only requirement is that the SIZ marker segment be the first marker segment present in the header.

F.3.2 TPJE Header Parameters

The following section describes the parameters used to populate the TPJE header. Table F-1 provides a summary of header parameters followed by a more detailed description of each marker and marker segment. The "Notes" column found on the table's far right is not to be construed as direct quotes from reference standards. Instead they are provided as helpful information to JPEG 2000 code implementers.

Table F-1 TPJE Marker Requirements within JPEG 2000 Main Header.

Marker	Size (bits)	Contents	Notes
SOC	16	0xFF4F	Required as first marker in main header.
SIZ	Lsiz·8 + 16	See Table F-4	Required as second marker (first marker segment) in the main header. Denotes the image and tile size
COD	Lcod·8 + 16	See Table F-5	Coding style default. Required in main header, and no more than one in the first tile-part header of a given tile. Indicates the usage of SOP and EPH.
QCD	Lqcd·8 + 16	See Table F-6 and Table F-7	Quantization default. One and only one required in the main header. May be at most one in the first tile-part header of a

Table F-1 TPJE Marker Requirements within JPEG 2000 Main Header.			
			given tile.
QCC	Lqcc·8 + 16	See Table F-8 and Table F-9	Quantization component. If present this marker segment overrides the QCD specified in the main header for specific individual components. No more than one per any given component in the main header or first tile-part header of a given tile.
TLM	Ltlm·8 + 16 bits per TLM	See Table F-10	Tile-part lengths. The inclusion of this marker segment allows individual tiles to be identified in the code stream without requiring all tile headers be read first. There may be multiple TLM marker segments (1-256) in the main header.
COM	5 – 65,636	See Table F-11	Optional. Allows the inclusion of unstructured data in the tile header. Multiple COM marker segments permitted.

F.3.2.1 Main Header Delimiting Markers and Marker Segments

The delimiting markers SOC and EOC shall be present in all TPJE headers (Table F-2 and Table F-3).

Table F-2 Start of Codestream (15444-1 Annex A.4.1).

Parameter	Size (bits)	TPJE	Notes
SOC	16	0xFF4F	Start of codestream.

Table F-3 End of Codestream (15444-1 Annex A.4.4).

Parameter	Size (bits)	TPJE	Notes
EOC	16	0xFFD9	End of codestream. Note: Though not technically considered to be part of the Main Header, this marker segment is required to terminate all valid JPEG 2000 codestreams and included here because it is often shown in close proximity to the SOC marker in the JPEG 2000 literature.

F.3.2.2 Main Header Fixed Information Marker Segments

Fixed marker segments provide required information about the image. TPJE has one fixed maker segment, SIZ. The SIZ marker segment follows the SOC marker in the codestream main header and identifies image and tile dimensions with respect to a reference grid. The contents of the SIZ marker segment are shown in Table F-4.

Table F-4 Image and tile size (15444-1 Annex A.5.1).

Parameter	Size (bits)	TPJE	Notes
SIZ	16	0xFF51	Image and tile size marker.
Lsiz	16	38 + 3·Csiz	Length of this marker segment in bytes (not including the marker).
Rsiz	16	0000 0000 0000 0010 (Profile-1)	Denotes capabilities that a decoder needs to properly decode the code stream. Note Rsiz = 2 indicates a 15444-1 Profile-1 compliant codestream.
Xsiz	32	1 to $(2^{31}-1)$	Width of Reference grid. This value should always be just the image width.
Ysiz	32	1 to $(2^{31}-1)$	Height of Reference grid. This value should always be just the image width.
XOsiz	32	0	Horizontal offset from the origin of the reference grid to the left side of the image area. Always set to 0 for TPJE.
YOsiz	32	0	Vertical offset from the origin of the reference grid to the top of the image area. Always set to 0 for TPJE.
XTsiz	32	256, 512, or 1024	Width of one reference tile with respect to the reference grid.
YTsiz	32	256, 512, or 1024	Height of one reference tile with respect to the reference grid.
XTOsiz	32	0	Horizontal offset from the origin of the reference grid to the left edge of the first tile. Always set to 0 for TPJE.
YTOsiz	32	0	Vertical offset from the origin of the reference grid to the top edge of the first tile. Always set to 0 for TPJE.
Csiz	16	1 to 16,384	The number of components (e.g. bands) in the image.
$Ssiz^i$ (i = 0,…Csiz-1)	8	0000 0111 (for 8 unsigned bit data) to 00001111 (for 16 unsigned bit data)	Value is one (1) less than bit depth of each component sample. First bit is 0 to indicate unsigned data. Repeat Csiz times.
$XRsiz^i$	8	1	Horizontal separation of a sample of

Table F-4 Image and tile size (15444-1 Annex A.5.1).			
Parameter	**Size (bits)**	**TPJE**	**Notes**
(i = 0,…Csiz-1)			the i^{th} component with respect to the reference grid. Repeat Csiz times.
$YRsiz^i$ (i = 0,…Csiz-1)	8	1	Vertical separation of a sample of the i^{th} component with respect to the reference grid. Repeat Csiz times.

F.3.2.3 Main Header Functional Marker Segments

Functional marker segments define what parameters were used in the compression of the whole image or individual component. These marker segments apply to the image when in the main header or the entire tile-part when in the tile-part header. Marker segments in the tile-part header supersede marker segments in the main header. For TPJE, COD (Table F-5), QCD (Table F-6 and Table F-7), and QCC (Table F-8 and Table F-9) are functional markers contained within the header.

Table F-5 Coding style default (15444-1 Annex A.6.1).			
Parameter	**Size (bits)**	**TPJE**	**Notes**
COD	16	0xFF52	Coding style default marker.
Lcod	16	12	Length of this marker segment in bytes (not including the marker).
Scod	8	0000 0000	Entropy coder with maximum precinct size. No SOP marker segments will be used. EPH marker will not be used.
SGcod	32	See next 3 rows	
Progression order	8	0 (L-R-C-P) 1 (R-L-C-P)	Layer-Resolution Level-Component-Position progression will be used for SNR progression. Resolution-Layer-Component- Position will be used for resolution progression.
Number of layers (N_{Layers})	16	1 to 20 for lossy 1 to 20 for lossless	Number of code stream layers including the non-truncated highest bit rate layer.
Multiple component transform	8	0	Currently, multiple component transforms are not allowed in TPJE, though they may be allowed in future versions.
SPcod	40	See next 6 rows	
Number of decomposition levels (N_{Levels})	8	5 to 9	There will be 5 to 9 wavelet decompositions, which form 6 to 10 resolution levels in the codestream.
Code-block width	8	3 or 4	The width of a code-block will be 32 or 64. The value of this parameter is such that it is two less than the base two exponent of the width.

Table F-5 Coding style default (15444-1 Annex A.6.1).			
Parameter	**Size (bits)**	**TPJE**	**Notes**
Code-block height	8	3 or 4	The height of a code-block will be 32 or 64. The value of this parameter is such that it is two less than the base two exponent of the height.
Code-block style	8	0000 0000 default	No reset of context probabilities on coding pass boundaries. No termination on each coding pass. No vertically causal context. No predictable termination. No segmentation symbols are used.
Transformation	8	0000 0000 for 9-7 irreversible (lossy) 0000 0001 for 5-3 reversible (lossless, lossy)	Wavelet filter used. Note: Under TPJE, the 5-3R filter can also be used for lossy compression. This is accomplished by truncating the final target bit rate such that numerically lossless compression is not achieved.
Precinct size	0	N/A	Not present. The precincts have been defined to be maximum size by Scod.

The QCD marker segment indicates the default quantization step size for all tile-parts. Table F-6 shows the values that appear in the QCD marker segment fields for JPEG 2000 lossy (including visually lossless) compression while Table F-7 provides the values for lossless compression.

Table F-6 Quantization default, 9-7 irreversible wavelet (15444-1 Annex A.6.4).			
Parameter	**Size (bits)**	**TPJE**	**Notes**
QCD	16	0xFF5C	Quantization default marker.
Lqcd	16	$5 + (6 \cdot N_{Levels})$	Length of this marker segment in bytes (not including the marker).
Sqcd	8	0100 0010	Number of guard bits = 2. Scalar expounded values used.
SPqcd[i]	16	Mantissa (μ_b) xxxx x000 0000 0000 to xxxx x111 1111 1111 Exponent (ε_b) 0000 0xxx to 1111 1xxxx	Exponent and mantissa values must appear explicitly for each of the subbands. See Table 7-18 or ISO/IEC 15444-1: Table A-30.

Table F-7 Quantization default, 5-3 reversible wavelet (15444-1 Annex A.6.4).			
Parameter	**Size (bits)**	**TPJE**	**Notes**
QCD	16	0xFF5D	Quantization default marker.
Lqcd	16	$4 + (3 \cdot N_{Levels})$	Length of this marker segment in bytes (not including the marker).
Sqcd	8	0100 0000	Number of guard bits = 2. No quantization.
$SPqcd^i$	8	Exponent (ε_b) 0000 0xxx to 1111 1xxxx	Reversible dynamic range exponent values must appear explicitly for each of the subbands. See Table 7-17 or ISO/IEC 15444-1: Table A-29.

The main header is also required to contain one QCC marker segment for each component of a multi-band image that has been quantized differently than the default value defined in the QCD parameter. If quantization step sizes are consistent across all components of the tiles, no QCC markers are required and the default value from the QCD marker segment is applied. Table F-8 shows the values that appear in the QCC marker segment fields for JPEG 2000 lossy (including visually lossless) compression while Table F-9 provides the values for lossless compression.

Table F-8 Quantization component, 9-7 irreversible wavelet (15444-1 Annex A.6.5).			
Parameter	**Size (bits)**	**TPJE**	**Notes**
QCC	16	0xFF5D	Quantization component marker.
Lqcc	16	$6 + (6 \cdot N_{Levels})$	Length of this marker segment in bytes (not including the marker).
Cqcc	8 (Csiz < 257) 16 (Csiz ≥ 257)	0 to (Csiz – 1)	Index of the component to which this marker segment relates.
Sqcc	8	0100 0010	Number of guard bits = 2. Scalar expounded values used.
$SPqcc^i$	16	Mantissa (μ_b) xxxx x000 0000 0000 to xxxx x111 1111 1111 Exponent (ε_b) 0000 0xxx to 1111 1xxxx	Exponent and mantissa values must appear explicitly for each of the subbands. See Table 7-18 or ISO/IEC 15444-1: Table A-30.

Table F-9 Quantization component, 5-3 reversible wavelet (15444-1 Annex A.6.5).

Parameter	Size (bits)	TPJE	Notes
QCC	16	0xFF5D	Quantization component marker.
Lqcc	16	$5 + (3 \cdot N_{Levels})$	Length of this marker segment in bytes (not including the marker).
Cqcc	8 (Csiz < 257) 16 (Csiz ≥ 257)	0 to (Csiz – 1)	Index of the component to which this marker segment relates.
Sqcc	8	0100 0000	Number of guard bits = 2. No quantization.
SPqcc[i]	8	Exponent (ε_b) 0000 0xxx to 1111 1xxxx	Reversible dynamic range exponent values must appear explicitly for each of the subbands. See Table 7-17 or ISO/IEC 15444-1: Table A-29.

F.3.2.4 Main Header Pointer Marker Segments

The pointer markers segments are used to gain quick access to desired data for parsing and decoding. The marker segments define either lengths of a dataset or pointers to the start of a dataset. The tile-part length (TLM) marker segment has the same length information as the start of tile marker segments in each tile-part, but this information is collected up front in the main header. This marker segment can be used to quickly access and spatially parse (chip out) a given tile, or set of tiles, in a compressed image. TPJE can contain up to 256 TLM marker segments defined as presented in Table F-10.

Table F-10 Tile-part lengths, main header (15444-1 Annex A.7.1).

Parameter	Size (bits)	TPJE	Notes
TLM	16	0xFF55	Tile-part lengths marker.
Ltlm	16	For codestreams containing a single TLM: 4 + (4·NTPMS) (8 to 65532) For codestreams containing multiple TLMs: 4 + (6·NTPMS) (10 to 65530) Note: NTPMS stands for number of tile parts in marker segment	Length of this marker segment in bytes (not including the marker).
Ztlm	8	0 to 255	Index of this marker segment relative to all other TLM marker segments present in the current header.

Parameter	Size (bits)	TPJE	Notes
			Note: There may be multiple TLM markers segments in the main header.
Stlm	8	Codestreams containing a single TLM: Stlm = 0100 0000 Codestreams containing multiple TLMs: Stlm = 0110 0000	Codestreams containing a single TLM: ST = 0, SP = 1; only one tile-part per tile and the tiles are in index order without omission or repetition, Ptlm offsets are 32 bits. Codestreams containing multiple TLMs: ST = 2 and SP = 1; Ttlm indices are 16 bits; Ptlm offsets are 32 bits.
$Ttlm^i$	0 16	Codestreams containing a single TLM: N/A Codestreams containing multiple TLMs: 0 to 65,534	For codestreams containing a single TLM, the tiles appear in order, so no Ttlm indices are necessary.
$Ptlm^i$	32	14 to $(2^{32}$-1)	The length, in bytes, from the beginning of the SOT marker of the i^{th} tile-part to the end of the code stream data for that tile-part. There will be one Ptlm for every tile-part.

F.3.2.5 Main Header Informational Marker Segments

Informational marker segments are strictly for providing additional information about the file and are not required for decoding. TPJE codestreams have the option to contain the COM marker segment to permit file producers to add additional informative data if desired (Table F-11). Note however, *no data shall be placed in a comment marker segment that is necessary to decode or properly interpret a JPEG 2000 codestream.* Neither metadata that might be used to properly interpret and exploit the compressed imagery nor any information that might be used to improve JPEG 2000 decoder performance shall be placed within a COM marker segment. An acceptable use of the COM marker segment might be to put the version number of the software used in creating the file.

Table F-11 Comment (15444-1 Annex A.9.2).

Parameter	Size (bits)	TPJE	Notes
COM	16	0xFF54	Optional information.
Lcom	16	5 to 65,535	Length of this marker segment in bytes (not including the marker).
Rcom	16	0 or 1	Registration Values: 0 = any binary data is allowed.

			1 = bytes are text encoded with the "Latin" character set defined in IS 8859-15:1999 (an extension of ASCII). All other values are reserved.
$Ccom^i$	8	0 to 255	Byte of unstructured data.

F.4 TPJE Tile-Part Header

The following discussion describes the tile-part header in TPJE-compliant codestreams.

F.4.1 TPJE Tile-Part Header Overview

The following section describes the TPJE tile-part header. TPJE permits one to ten tile-parts per tile. The tile-part header contains the following:

1) SOT (start of tile-part) marker
2) A QCD (quantization default) marker segment is required for both lossy (including visually lossless) and numerically lossless cases in the event that certain tiles are handled (e.g. quantized) differently than the processing specified by the main header's QCD or QCC values
3) QCC (quantization component) marker segment is required for lossy (including visually lossless) and lossless compression if…
    a. The file is a multi-component data set

    -and-

    b. One or more of the tile components is to be handled (e.g. quantized) differently from the processing specified in the main header QCD, QCC or Tile Header QCD (*For an overview of Main Header and Tile Header QCD and QCC Values, see section F.4.2.4*)
4) PLT (packet-length, tile-part) marker segment
5) COM (optional information), marker segment, optional
6) SOD (start of data) marker

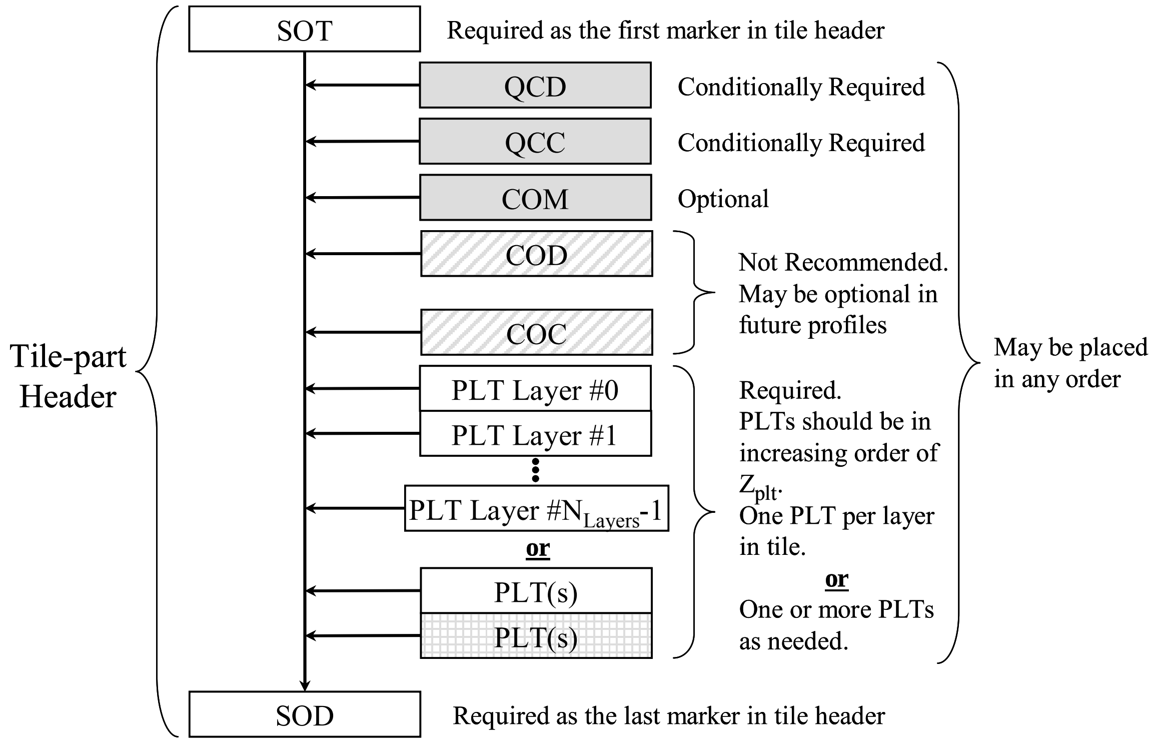

Figure F-3 Overview of the First Tile-Part Header for TPJE. First Tile-Part Header for Multiple PLT Option (see Section F.1).

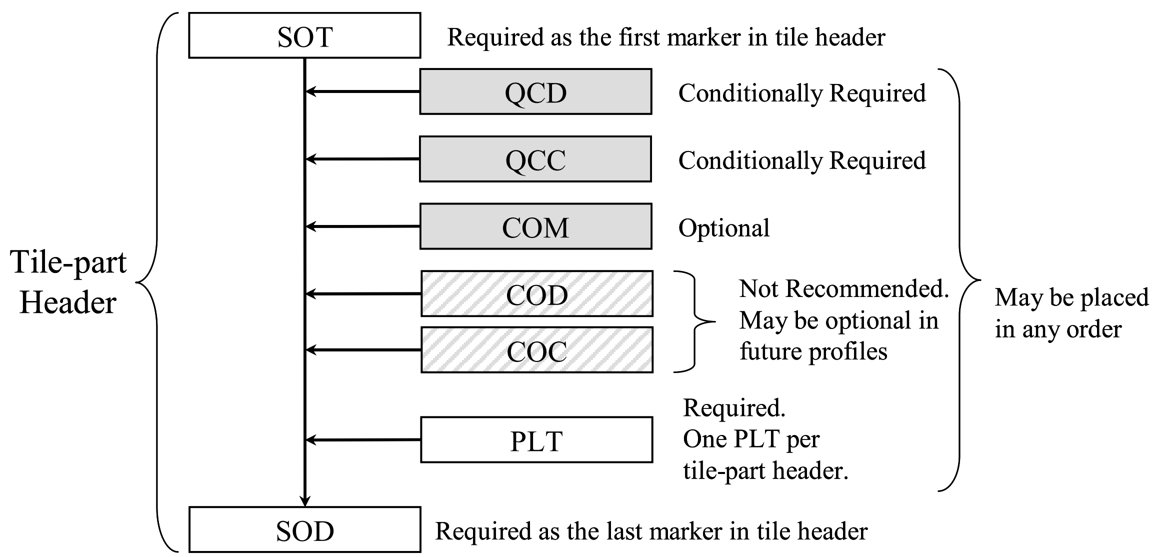

Figure F-4 Overview of the First Tile-Part Header for TPJE. First Tile-Part Header for Single PLT Option (see Section F.1).

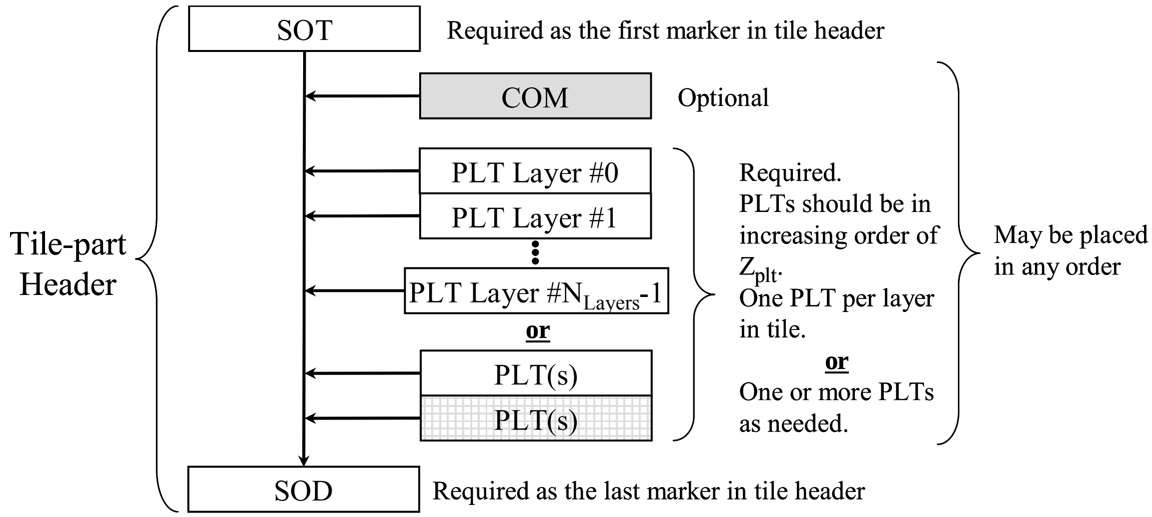

Figure F-5 Overview of the Non-First Tile-Part Header for TPJE. Non-First Tile-Part Header for Multiple PLT Option (see Section F.1)

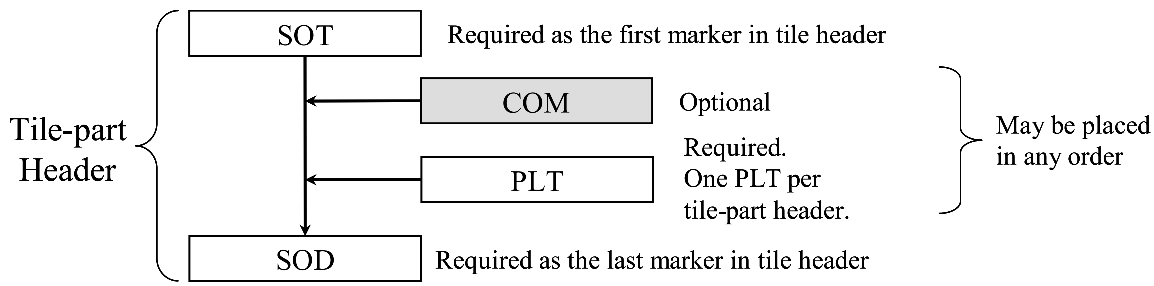

Figure F-6 Overview of the Non-First Tile-Part Header for TPJE. Non-First Tile-Part Header for Single PLT Option (see Section F.1).

Note that, as shown in Figure F-3, Figure F-4, Figure F-5 and Figure F-6 above, the markers, QCD (present in the first tile-part only), QCC (present in the first tile-part only), PLT, and COM can appear in any order in the tile-part header.

F.4.2 TPJE Tile-Part Parameters

Table F-12 provides an overview of the tile-part header markers. The “Notes” column found on the table’s far right is not to be construed as direct quotes from reference standards. Instead they are provided as helpful information to JPEG 2000 code implementers.

Table F-12 TPJE Marker Requirements within JPEG 2000 Tile-Part Header.

Marker	Size (bits)	TPJE	Notes
SOT	96	Table F-13	Required as first marker in main header and therefore the codestream.
QCD	Lqcd·8 + 16	Table F-6 and Table F-7	Quantization default. If present in the tile header, this marker segment overrides the QCD specified in the main header. Note: When using multiple tile-parts, the tile-

Marker	Size (bits)	TPJE	Notes
			part header QCD is only allowed in the first tile-part header.
QCC	Lqcc·8 + 16	Table F-8 and Table F-9	Quantization component. If present, this marker specifies coefficient quantization values for individual components in this tile. The Tile QCC overrides both the QCD and QCC parameters specified in the main header and tile header QCD value. Note: When using multiple tile-parts, the tile-part header QCC is only allowed in the first tile-part header.
PLT	Lppt·8 + 16 per PLT	Table F-15	Packet length, tile-part header. The inclusion of this marker segment allows individual packets to be identified in the code stream without requiring that all packet headers be read first.
COM	5 to 65,636	Table F-11	Optional. Allows the inclusion of unstructured data in the tile header. Multiple COM marker segments permitted.
SOD	16	Table F-14	Start of data marker.

F.4.2.1 Tile-Part Header Delimiting Markers and Marker Segments

The delimiting markers SOT and SOD and their marker segments shall be present in all TPJE tile-parts. SOT (Table F-13) denotes the start of the tile-part. It cannot be completed until the length of the tile-part is known.

Table F-13 Start of tile-part (15444-1 Annex A.4.2).

Parameter	Size (bits)	TPJE	Notes
SOT	16	0xFF90	Start of tile-part marker.
Lsot	16	10	Length of this marker segment in bytes (not including the marker).
Isot	16	0 to 65,534	Tile index. This number refers to the tiles in raster-scan order starting at zero.
Psot	32	14 to (2^{32}-1)	Length, in bytes, from the beginning of the first byte of the SOT marker segment of the tile-part to the end of the data of the tile-part.
TPsot	8	0 to 9	Tile-part index. Because there is only one tile-part per tile, this value will always be zero for the L-R-C-P progression order. For the R-L-C-P progression order, there may be up to 10 tile-parts per tile, resulting in an index from 0 to 9.

Parameter	Size (bits)	TPJE	Notes
TNsot	8	0, 1 or 6 to 10	0 = Number of tile-parts of this tile in the codestream is not defined in this header. 1 or 6 to 10 = The number of tile-parts of a tile in the codestream. If there is only one tile-part per tile, the value is 1. If there are multiple tile-parts per tile, there will be 1 tile-part per resolution level per tile.
Note: The SOT marker cannot be completed until the length of the tile-part data is known.			

Table F-14 Start of data (15444-1 Annex A.4.3).

Parameter	Size (bits)	TPJE	Notes
SOD	16	0xFF93	Start of data marker.

Following the SOT marker information, the tile-part header may contain a QCD marker segment if the data has been coded with different quantization values. If the quantization step sizes are constant across tiles, the QCD marker segment is used only in the main header and those values apply to all tiles in the file. The format of the QCD marker in a tile-header is the same as that described in Table F-6 for lossy data and Table F-7 for lossless data.

F.4.2.2 Tile-Part Header Functional Marker Segments

Functional marker segments define what parameters were used in the compression of the whole image or individual component. In some cases, it is useful or necessary to vary these parameters on a tile basis. This might happen when portions of the image area exhibit very different image properties or statistics. A good example would be when a coastline is imaged. The land and water portions of the image might display very different properties. In these cases, TPJE allows the use of tile-specific QCD (Table F-6 and Table F-7) and QCC (Table F-8 and Table F-9) marker segments. Usage of the COD and COC marker segments in a tile-part header is not recommended in TPJE. See section F.4.2.4 for further details regarding the use of QCD and QCC marker segments.

F.4.2.3 Tile-Part Header Pointer Marker Segments

Within each tile-part header is recommended the packet length (PLT) marker segment which allows quick access to a given packet or group of packets for parsing or decoding data at a given quality or resolution level.

PLT marker segments indicate the length of each packet in the tile-part. If the codestream uses the L-R-C-P progression order, it is recommended that there be one PLT marker segment per quality layer in the codestream or that there be one PLT for all quality layers (unless more are required due to marker segment length restrictions). This behavior follows that of NPJE. If the codestream uses the R-L-C-P progression order, there are two possible uses of PLT marker segments within TPJE. First, there shall be one

PLT marker segment for all quality layers within a tile-part as defined within EPJE **or** there shall be one PLT marker segment per quality layer within a tile-part.

In the case of one tile-part per tile, the second option follows the behavior of NPJE. This allows creation of "NPJE-like" codestreams that utilize the R-L-C-P progression order. If the second option is used in the case of multiple tile-parts per tile, the result is an EPJE/NPJE mix. The PLT marker segments within a single tile-part header are inserted in the code stream in the same order as the corresponding layers. From lowest (Zplt = 0) to highest (Zplt = N_{layers} – 1) bit rate. The PLT marker segment fields are populated according to Table F-15.

Table F-15 Packet-lengths tile-part header (15444-1 Annex A.7.3).

Parameter	Size (bits)	TPJE	Notes
PLT	16	0xFF58	Packet length, tile-part header marker
Lplt	16	4 to 65535	Length of this marker segment in bytes (not including the marker).
Zplt	8	0 to (N_{Layers} – 1)	Index of this marker segment relative to all other PLT marker segments in the current header.
$Iplt^i$	8 bits repeated as necessary See Table 7-24 or ISO/IEC 15444-1: Table A-36.	0xxx xxxx	Signals that the next seven bits are the last bits indicating the length of the i^{th} packet.

F.4.2.4 Tile-Part Header Informational Marker Segments

The COM marker segment (Table F-11) may be used within tile-part headers to convey informational data. *No data shall be placed in a comment marker segment that is necessary to decode or properly interpret a JPEG 2000 codestream.* Neither metadata that might be used to properly interpret and exploit the compressed imagery nor any information that might be used to improve JPEG 2000 decoder performance shall be placed within a COM marker segment.

F.5 Overview of QCD and QCC in the Main and Tile Header

Coefficient quantization is controlled by four marker segments in JPEG 2000. They are the QCD and QCC values found in the main header and the QCD and QCC values found in the Tile header. The following is a short summary of each parameter and how it can be superseded.

- Main Header QCD: Provides the default coefficient quantization value for the entire image. These values can only be superseded by the Main header's QCC value and the Tile header's QCD and QCC values.

- Main Header QCC: Used in instances when one of the components of a multi-component image has a different quantization value than the remaining components. This value can only be superseded by the Tile header's QCD and QCC values.

- Tile Header QCD: This parameter gives the default value for the entire tile. It can be only superseded by the Tile QCC value.

- Tile Header QCC: This parameter gives the default value for a single tile-component. No parameter can supersede this value.

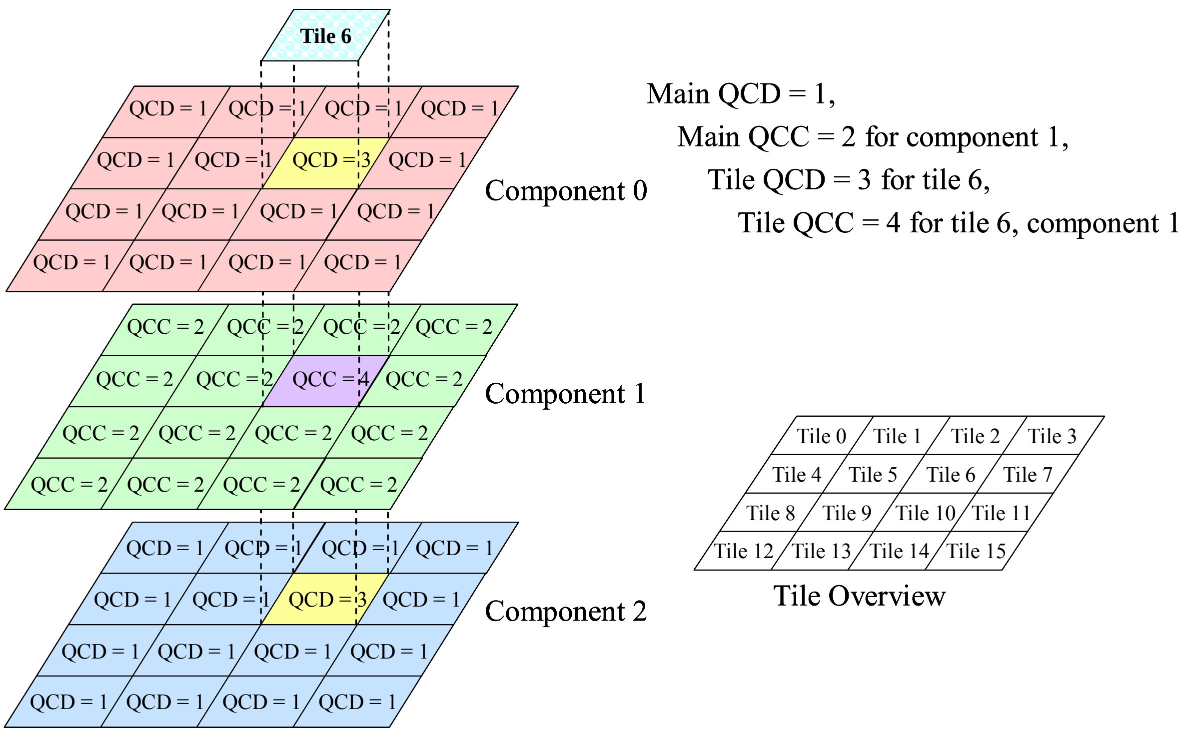

Figure F-7 Overview of Main Header and Tile Header QCD and QCC Values.

Figure F-7 gives an example where all four types of marker segments are used (main and tile-part header, QCD and QCC). This figure illustrates the proper way to signal the various component, tile and tile-component changes with a minimal number of marker segments. The precedence of the tile-part QCD and QCC over the main header QCD and QCC, as well as the precedence of the QCC marker segment over the QCD marker segment are shown.

F.6 Recommended Compression Rate Control

Rate control is intimately tied with layer formation. The rate control mechanism decides how many and which bytes are allocated per layer in the compressed file. The JPEG 2000 standard leaves this matter up to the implementer, but an example of a detailed rate control procedure may be found in NIMA N0106-97. Three common forms of rate control used with tiled imagery are:

- Full image rate control
- Individual tile rate control
- Individual tile-component rate control

Full image rate control produces a more uniform image quality by allowing busy tiles (those with fine detail) to contribute more bytes to the compressed file at the expense of less busy tiles. The aggregate bit rate for the entire image is still maintained, but the tile-by-tile bit rate can change. In general, full image bit rate control consumes more memory during compression since compressed data from each tile must be retained to form the final layers. Full image rate control is not conducive to independent tile processing for this same reason.

Individual tile rate control allocates the same number of bits to each tile regardless of that tile's scene content. This allows every tile to be layered and processed independently. If the scene variation between neighboring tiles varies dramatically, then there may be noticeable image quality differences between tiles at the same layer. For example, tile boundaries might become visible to the user even at higher bit rates (visible tile boundaries often occur at very low bit rates regardless of rate control paradigm). However, this problem will not manifest itself at visually lossless bit rates.

Note that with multi-component (multi-band) imagery individual tile rate control allows the sharing of bits across tile-components. This is analogous to the full image rate control process of sharing bits across multiple tiles, but in this case only the components (bands) within a given tile can share bits. This means that one component in a given tile can have a different number of compressed bits than the other components in that tile. The compressed bits allocated to a given component can also exceed the desired target bit rate as long as the total bit rate across all components in the tile does not exceed the target bit rate for the tile.

Individual tile-component rate control allocates the same number of bits to each tile-component regardless of that tile's scene content. This allows every tile-component to be layered and processed independently. This rate control method has the same implications for image quality as the individual tile rate control method, except that possible image quality issues may now arise between the various color-channels. Again, these problems will not manifest themselves at visually lossless bit rates – though the definition of a visually lossless bit rate for the color display of multi-component imagery may be difficult to define for some multi-spectral imaging systems. For this reason, many remote sensing multi-spectral systems rely on numerically lossless JPEG 2000 compression for the formation of their individual tile-component rate controlled codestreams. It should be pointed out that, for single band imagery, the individual tile-component form of rate control and the individual tile form of rate control are identical in implementation.

The TPJE preferred encoding makes no recommendation as to the choice of rate control that should be used by a given remote sensing system. All three options have their own merits and drawbacks. It is left to the system designer to determine which form of rate control best meets the needs of their imaging system and their potential imagery product end-users.

F.7 JPEG 2000 Recommended Layers

The TPJE preferred encoding permits encoders to generate anywhere between 1 and 20 Quality Layers. This section provides several examples of how the bits per pixel per band (bpppb) could be divided up between the Quality Layers chosen for the TPJE codestream.

Table D-17 shows the common break-down of Quality Layer values used by many community data providers (NPJE). Table F-16 shows an example of a TPJE codestream using fewer Quality Layers. Section F.8.1 demonstrates how each of these tables' corresponding JPEG 2000 Layer Target Bit Rates tagged record extensions (J2KLRA TRE) would be populated if the codestreams were embedded in a BIIF, NSIF, or NITF file.

Note that the actual target bit rates and numbers of Quality Layers used for a specific sensor would need to be optimized to accommodate any unique properties of that particular imagery collection system. Likewise, the number of bits per pixel per band that are needed to define a visually lossless target bit rate would need to be determined for each individual collection system.

Table F-16 TPJE Example Layers and Bits per Pixel per Band.

Layer	Bits Per Pixel Per Band (bpppb)
Layer 6	2.0 bpppb (lossy compression)
Layer 5	1.0 bpppb
Layer 4	0.5 bpppb
Layer 3	0.25 bpppb
Layer 2	0.125 bpppb
Layer 1	0.0625 bpppb
Layer 0	0.03125 bpppb

F.8 TPJE Use within NSIF Formatted Files

The following sections describe TPJE specific recommendations for the use of TPJE within the NSIF file format. The recommendations from section 8 of this document apply to TPJE compliant codestreams within NSIF as well and must be followed unless modified by the following text. These recommendations (as well as those of section 8) shall apply to the use of TPJE within the BIIF and NITF file formats as well.

F.8.1 Recommended J2KLRA TRE Population for TPJE

The general form of the JPEG 2000 Layer Target Bit-Rates tagged record extension (J2KLRA TRE) was primarily developed to allow providers and users of NPJE-encoded and EPJE-encoded image data embedded in BIIF, NSIF, or NITF files to quickly access the compressed data, but the TRE is also available to be used by other encodings. The TRE provides users information about the number of Resolution Levels, number of Quality Layers, and number of components (bands) in both the original dataset and derived (parsed and/or transcoded) products. This information can be critical in the selection and ordering of data from an image library.

The J2KLRA TRE is recommended to be included with any original compressed data and compliant derived compressed products (i.e. parsing and repackaging) produced under TPJE and embedded in a BIIF, NSIF, or NITF file. Table F-17 provides the specific J2KLRA TRE population ranges for TPJE-compliant NSIF datasets. Note that the value ranges presented in this table are in some cases more restrictive than those provided in the standard definition of this TRE, and in other cases the value ranges augment the ranges provided in the standard definition of this TRE.

Table F-17 JPEG 2000 Layer Target Bit-Rates (J2KLRA TRE) Format for TPJE.

Field	Name/Description	Size Bytes	Format	Value Range	Type
CETAG	Unique Extension Type Identifier	6	BCS-A	J2KLRA	R
CEL	Length of User-Defined Data	5	BCS-N	00023-00261	R
ORIG	Original Compressed Data Note that the values 0, 1, 2, and 3 are provided since it is recommended that if an TPJE-compliant codestream is also either NPJE- or EPJE-compliant, then the TRE population shall reflect the most restrictive preferred encoding compliance for the codestream in question. This serves to promote wider interoperability with decoders found in the user community. Other values 6, 7, 8, 9 shall not be used with TPJE (see Table 8-3).	1	BCS-N	0 – Original NPJE 1 – Parsed NPJE 2 – Original EPJE 3 – Parsed EPJE 4 – Original TPJE 5 – Parsed TPJE	R
Original compressed image information (the first JPEG 2000 compression).					
NLEVELS_O	Number of Wavelet Levels in Original Image	2	BCS-N	05-09	R
NBANDS_O	Number of Bands in Original Image	5	BCS-N	00001-16384	R
NLAYERS_O	Number of Layers in Original Image	3	BCS-N	001-020	R
Start of layer information loop. Field repeats for n = 0 to NLAYERS_O - 1 times.					
LAYER_IDn	Layer ID Number	3	BCS-N	000-019	R
BITRATEn	Bitrate	9	BCS-A	00.000000-37.000000*	R
End of layer information loop.					

Field	Name/Description	Size Bytes	Format	Value Range	Type
Start of conditional fields for parsed data (ORIG = 1, 3, or 5).					
NLEVELS_I	Number of Wavelet Levels in This Image	2	BCS-N	00-09	C
NBANDS_I	Number of Bands in This Image	5	BCS-N	00001-16384	C
NLAYERS_I	Number of Layers in This Image	3	BCS-N	001-020	C
* The component sample precision is limited by the number of guard bits, quantization, growth of coefficients at each decomposition level, and the number of coding passes that can be signaled. Not all combinations of coding styles will allow the coding of 37 bit samples per band.					

During primary compression, all of the TRE fields, except for the conditional NLEVELS_I, NLAYERS_I, and NBANDS_I fields, are populated. When an image is repackaged, the CEL and ORIG fields are updated and NLEVELS_I, NLAYERS_I, and NBANDS_I are added or replaced (if they already exist).

The following J2KLRA TRE population example corresponds to the quality layers versus target bit rates per pixel per band as shown in the visually-lossless case provided in Table D-17.

Table F-18 J2KLRA TRE Metadata Population Example (Matches Table D-17).

Field	Name/Description	Size Bytes	Format	Value Range	Type
CETAG	Unique Extension Type Identifier	6	BCS-A	J2KLRA	R
CEL	Length of User-Defined Data	5	BCS-N	00239	R
ORIG	Original Compressed Data	1	BCS-N	4 – Original APJE	R
Original compressed image information (the first JPEG 2000 compression).					
NLEVELS_O	Number of Wavelet Levels in Original Image	2	BCS-N	05	R
NBANDS_O	Number of Bands in Original Image	5	BCS-N	00001	R
NLAYERS_O	Number of Layers in Original Image	3	BCS-N	019	R
Start of layer information loop. Field repeats for n = 0 to NLAYERS_O - 1 times.					
$LAYER\_ID_0$	0^{th} Layer ID Number	3	BCS-N	000	R
$BITRATE_0$	0^{th} Bit Rate	9	BCS-A	00.031250	R
$LAYER\_ID_1$	1^{st} Layer ID Number	3	BCS-N	001	R
$BITRATE_1$	1^{st} Bit Rate	9	BCS-A	00.062500	R
$LAYER\_ID_2$	2^{nd} Layer ID Number	3	BCS-N	002	R
$BITRATE_2$	2^{nd} Bit Rate	9	BCS-A	00.125000	R
$LAYER\_ID_3$	3^{rd} Layer ID Number	3	BCS-N	003	R
$BITRATE_3$	3^{rd} Bit Rate	9	BCS-A	00.250000	R
$LAYER\_ID_4$	4^{th} Layer ID Number	3	BCS-N	004	R
$BITRATE_4$	4^{th} Bit Rate	9	BCS-A	00.500000	R
$LAYER\_ID_5$	5^{th} Layer ID Number	3	BCS-N	005	R

Field	Name/Description	Size Bytes	Format	Value Range	Type
$BITRATE_5$	5^{th} Bit Rate	9	BCS-A	00.600000	R
$LAYER\_ID_6$	6^{th} Layer ID Number	3	BCS-N	006	R
$BITRATE_6$	6^{th} Bit Rate	9	BCS-A	00.700000	R
$LAYER\_ID_7$	7^{th} Layer ID Number	3	BCS-N	007	R
$BITRATE_7$	7^{th} Bit Rate	9	BCS-A	00.800000	R
$LAYER\_ID_8$	8^{th} Layer ID Number	3	BCS-N	008	R
$BITRATE_8$	8^{th} Bit Rate	9	BCS-A	00.900000	R
$LAYER\_ID_9$	9^{th} Layer ID Number	3	BCS-N	009	R
$BITRATE_9$	9^{th} Bit Rate	9	BCS-A	01.000000	R
$LAYER\_ID_{10}$	10^{th} Layer ID Number	3	BCS-N	010	R
$BITRATE_{10}$	10^{th} Bit Rate	9	BCS-A	01.100000	R
$LAYER\_ID_{11}$	11^{th} Layer ID Number	3	BCS-N	011	R
$BITRATE_{11}$	11^{th} Bit Rate	9	BCS-A	01.200000	R
$LAYER\_ID_{12}$	12^{th} Layer ID Number	3	BCS-N	012	R
$BITRATE_{12}$	12^{th} Bit Rate	9	BCS-A	01.300000	R
$LAYER\_ID_{13}$	13^{th} Layer ID Number	3	BCS-N	013	R
$BITRATE_{13}$	13^{th} Bit Rate	9	BCS-A	01.500000	R
$LAYER\_ID_{14}$	14^{th} Layer ID Number	3	BCS-N	014	R
$BITRATE_{14}$	14^{th} Bit Rate	9	BCS-A	01.700000	R
$LAYER\_ID_{15}$	15^{th} Layer ID Number	3	BCS-N	015	R
$BITRATE_{15}$	15^{th} Bit Rate	9	BCS-A	02.000000	R
$LAYER\_ID_{16}$	16^{th} Layer ID Number	3	BCS-N	016	R
$BITRATE_{16}$	16^{th} Bit Rate	9	BCS-A	02.300000	R
$LAYER\_ID_{17}$	17^{th} Layer ID Number	3	BCS-N	017	R
$BITRATE_{17}$	17^{th} Bit Rate	9	BCS-A	02.800000	R
$LAYER\_ID_{18}$	18^{th} Layer ID Number	3	BCS-N	018	R
$BITRATE_{18}$	18^{th} Bit Rate	9	BCS-A	03.500000	R
End of layer information loop.					

The following J2KLRA TRE population example corresponds to the quality layers versus target bit rates per pixel per band as shown in the lossy case provided in Table F-16. Note that this is defined as a lossy original compression under TPJE. Under TPJE, lossy compression is allowed to any arbitrary final target bit rate desired by a user. Furthermore, lossy compression may use any number of target bit rate layers, between 1 and 20, with arbitrary target bit rates set at those layers.

Under this paradigm, it is essential that all NSIF handling software tools be able to interpret the J2KLRA TRE. Hard-coding of target bit rates and their associated layers is no longer an acceptable practice for the user community. An image processing system handling an image compressed under TPJE might encounter any number of resolution levels (6 to 10, since 5 to 9 wavelet decompositions are allowed), quality layers (1 to 20), target bit rates (00.000000 to 37.000000) at the given quality layers, choice of wavelet filter (9-7I or 5-3R), progression order, and rate control paradigm. Note that the first three items listed in the previous sentence are discoverable in the J2KLRA TRE; the remaining items need to be obtained from the JPEG 2000 codestream itself.

Table F-19 J2KLRA TRE Metadata Population Example (Matches Table F-16).

Field	Name/Description	Size Bytes	Format	Value Range	Type
CETAG	Unique Extension Type Identifier	6	BCS-A	J2KLRA	R
CEL	Length of User-Defined Data	5	BCS-N	00095	R
ORIG	Original Compressed Data	1	BCS-N	4 – Original APJE	R
Original compressed image information (the first JPEG 2000 compression).					
NLEVELS_O	Number of Wavelet Levels in Original Image	2	BCS-N	06	R
NBANDS_O	Number of Bands in Original Image	5	BCS-N	00001	R
NLAYERS_O	Number of Layers in Original Image	3	BCS-N	007	R
Start of layer information loop. Field repeats for n = 0 to NLAYERS_O - 1 times.					
$LAYER\_ID_0$	0^{th} Layer ID Number	3	BCS-N	000	R
$BITRATE_0$	0^{th} Bit Rate	9	BCS-A	00.031250	R
$LAYER\_ID_1$	1^{st} Layer ID Number	3	BCS-N	001	R
$BITRATE_1$	1^{st} Bit Rate	9	BCS-A	00.062500	R
$LAYER\_ID_2$	2^{nd} Layer ID Number	3	BCS-N	002	R
$BITRATE_2$	2^{nd} Bit Rate	9	BCS-A	00.125000	R
$LAYER\_ID_3$	3^{rd} Layer ID Number	3	BCS-N	003	R
$BITRATE_3$	3^{rd} Bit Rate	9	BCS-A	00.250000	R
$LAYER\_ID_4$	4^{th} Layer ID Number	3	BCS-N	004	R
$BITRATE_4$	4^{th} Bit Rate	9	BCS-A	00.500000	R
$LAYER\_ID_5$	5^{th} Layer ID Number	3	BCS-N	005	R
$BITRATE_5$	5^{th} Bit Rate	9	BCS-A	01.000000	R
$LAYER\_ID_6$	6^{th} Layer ID Number	3	BCS-N	006	R
$BITRATE_6$	6^{th} Bit Rate	9	BCS-A	02.000000	R
End of layer information loop.					

F.8.2 Recommended ICHIPB TRE Population for TPJE

The definition of the Image Chip tagged record extension (ICHIPB TRE) is provided in "The Compendium of Controlled Extensions (CE) for the National Imagery Transmission Format (NITF) Version 2.1" (STDI-0002) standards document. It is strongly recommended that this TRE be included in any NSIF file containing a TPJE-compliant, JPEG 2000 compressed codestream.

The ICHIPB TRE serves to record the row and column offset information required to relate the current JPEG 2000 codestream to the BIIF CCS coordinates of the original JPEG 2000 codestream. This information is important, since it allows a spatially-parsed, re-sampled (e.g. asymmetric pixel spacing correction), and/or resolution parsed codestream to have its associated NSIF support data translated into the image space of the current codestream.

TPJE requires that any positional offset information that would normally be stored in a JPEG 2000 codestream's SIZ marker segment (fields XOsiz, YOsiz, XTOsiz, and YTOsiz) always be set to a value of zero. Since this requirement effectively "hides" the act of spatial chipping from the JPEG 2000 codestream metadata, there is no way for an end-user to know if a received codestream has been spatially parsed or not. Due to this requirement, the ICHIPB TRE must be used to record this information for codestreams embedded in NSIF files. It is strongly recommended that the ICHIPB TRE be included in all NSIF files containing original JPEG 2000 compressed codestreams, as well as parsed codestreams.

F.8.3 Recommended HISTOA TRE Population for TPJE

The definition of the Softcopy History tagged record extension (HISTOA TRE) is provided in "The Compendium of Controlled Extensions (CE) for the National Imagery Transmission Format (NITF) Version 2.1" (STDI-0002) standards document. It is strongly recommended that this TRE be included in any BIIF, NSIF, or NITF file containing a TPJE-compliant, JPEG 2000 compressed codestream.

The HISTOA TRE is used to record image processing events that have taken place for a given BIIF, NSIF, or NITF dataset. When a JPEG 2000 compressed BIIF, NSIF, or NITF dataset is originally formed, various fields in the HISTOA TRE may need to be populated based on the formation process of the JPEG 2000 codestream. These fields include:

- Prior Compression (PC)
- Number of Image Processing Comments (NIPCOM)
  - n^{th} Image Processing Comment (IPCOMn)
- Input Bandwidth Compression (INBWC)
- Asymmetric Correction (ASYM_FLAG)
  - Magnification in Line Direction (ZOOMROW)
  - Magnification in Element Direction (ZOOMCOL)
- Symmetrical Magnification (MAG_FLAG)
  - Level of Relative Magnification (MAG_LEVEL)
- Output Bandwidth Compression (OUTBWC)

Second-tier bullets in the above list denote HISTOA TRE fields that are conditional upon the presence of the first-tier bullets they follow. Note that in some cases, some of these fields will not be affected by original JPEG 2000 compression. The potential to affect these listed fields is driven by TPJE, JPEG 2000 compression generally, or community-derived best practices for NSIF file support data population.

See sections 8.2.1.5 and 8.2.1.6 for a discussion of the use of the Asymmetric Correction field with the formation of original, as well as parsed, JPEG 2000 codestreams. Section

8.2.1.5 also addresses the effect that JPEG 2000 original compression and/or parsing might have on the Symmetrical Magnification field.

The HISTOA field values, “J2NL” and “J2VL”, have been added to the list of possible INBWC and OUTBWC field values to denote numerically lossless JPEG 2000 compression and visually lossless JPEG 2000 compression, respectively. Additionally the code value, “J2LO”, to denote original lossy compression that is below the visually lossless threshold for a given imaging system, was added to the HISTOA list of valid entries for the INBWC and OUTBWC fields.

Image Processing Comments should be recorded in the HISTOA TRE when JPEG 2000 compression, transcoding, and/or parsing operations are being used. Some example image processing comments employed by the BIIF, NSIF, and NITF-using communities today are found in Table F-20.

Table F-20 Example of Currently Used JPEG 2000-Related Image Processing Comments for the HISTOA TRE.

HISTOA Image Processing Comments	Notes
Spatial Chip	Indicates that spatial parsing of the codestream(s) in the BIIF, NSIF, or NITF dataset(s) has occurred.
Transcode to EPJE	Indicates that the current codestream was formed by transcoding a NPJE-compliant codestream to an EPJE-compliant codestream.
Tile nulling	Indicates either the removal (block masking) of some number of tiles (all tile-parts in each tile) from a codestream, or their replacement with compressed tiles that hold uniform-valued scenes (e.g. zero-filled replacement tiles).
Quality Layer Parsing	Indicates that quality parsing of the codestream(s) in the BIIF, NSIF, or NITF dataset(s) has occurred.
Uncompression	Indicates that the originally JPEG 2000-compressed BIIF, NSIF, or NITF dataset(s) has been expanded to an uncompressed format.
Rset extraction/generation	Indicates that resolution parsing of the codestream(s) in the BIIF, NSIF, or NITF dataset(s) has occurred to aid in Reduced Resolution Data Set production.
Note: Multiple qualifying event comments can be concatenated using a BCS-A "+" symbol.	

The following comments, listed in Table F-21, are recommended for community use with JPEG 2000-compressed BIIF, NSIF, and NITF datasets in general, and with TPJE encoded datasets specifically. This list is not meant to be exhaustive or limiting in any way. The HISTOA TRE specification lists the IPCOMn field as being an 80 character field suitable for the recording of free-text alphanumeric comments about the current image processing event. As such, implementers are able to record any form of comment that they deem to be suitable for their needs. The recommended comments provided in Table F-21 are offered as simple guidance for promoting some level of community uniformity with respect to the population of the HISTOA TRE with JPEG 2000-compressed datasets.

Table F-21 Recommended JPEG 2000-Related Image Processing Comments for the HISTOA TRE.

HISTOA Image Processing Comments	Notes
Spatial Chip	Indicates that spatial parsing of the codestream(s) in the BIIF, NSIF, or NITF dataset(s) has occurred.
Transcode to EPJE	Indicates that the current codestream was formed by transcoding a NPJE-compliant (or TPJE-compliant) codestream to an EPJE-compliant codestream.
Tile nulling	Indicates either the removal (block masking) of some number of tiles (all tile-parts in each tile) from a codestream, or their replacement with compressed tiles that hold uniform-valued scenes (e.g. zero-filled replacement tiles).
Quality Layer Parsing	Indicates that quality parsing of the codestream(s) in the BIIF, NSIF, or NITF dataset(s) has occurred.
Uncompression	Indicates that the originally JPEG 2000-compressed BIIF, NSIF, or NITF dataset(s) has been expanded to an uncompressed format.
Rset extraction/generation	Indicates that resolution parsing of the codestream(s) in the BIIF, NSIF, or NITF dataset(s) has occurred to aid in Reduced Resolution Data Set production.
Resolution Parsing	Indicates that resolution parsing of the codestream(s) in the BIIF, NSIF, or NITF dataset(s) has occurred.
Component Parsing	Indicates that component parsing of the codestream(s) in the BIIF, NSIF, or NITF dataset(s) has occurred.
Transcode to NPJE	Indicates that the current codestream was formed by transcoding an TPJE-compliant or EPJE-compliant codestream to an NPJE-compliant codestream.
Transcode to TPJE	Indicates that the current codestream was formed by transcoding a NPJE-compliant or EPJE-compliant codestream to an TPJE-compliant codestream.
L-R-C-P Progression	The codestream uses the L-R-C-P progression order.
R-L-C-P Progression	The codestream uses the R-L-C-P progression order.
5-3R Filter	The codestream uses the 5-3R Filter to compress to a lossy, visually lossless, or numerically lossless product.
9-7I Filter	The codestream uses the 9-7I Filter to compress to a lossy or visually lossless product.
Resampled Pixel Spacing	Indicates that the pixel grid spacing of the codestream has been re-sampled (usually used in conjunction with asymmetric pixel correction).
RGB Component Transformation	Indicates that the codestream has been transformed from some color space to the RGB color space (usually used to record when YCbCr color space imagery is converted to RGB color space).
YCbCr Component Transformation	Indicates that the codestream has been transformed from some color space to the YCbCr color space (usually used to record when RGB color space imagery is converted to YCbCr color space).
Principal Components Transformation	Indicates that the multi-component codestream has been transformed to a new multi-component space via the Principal Components algorithm.
Inverse Principal Components Transformation	Indicates that the previously Principal Component algorithm processed multi-component codestream has been inverse transformed to a new multi-component space via the Inverse Principal Components algorithm.

HISTOA Image Processing Comments	Notes
Num Lossless Comp	Indicates that the codestream has been compressed to a numerically lossless state using the 5-3R filter.
Vis Lossless Comp	Indicates that the codestream has been compressed to a visually lossless state using either the 9-7I or 5-3R filter.
Lossy Comp	Indicates that the codestream has been compressed to a lossy state using either the 9-7I or 5-3R filter.
Multiple PLT	Indicates that the codestream uses the multiple PLT marker segment method (see Section 1.1).
Single PLT	Indicates that the codestream uses the single PLT marker segment method (see Section 1.1).
Other PLT	Indicates that the codestream uses PLT marker segments differently from the methods described in Section 1.1.
Orig EPJE Cmp	Indicates that the original JPEG 2000 compression of the dataset is compliant to EPJE.
Orig NPJE Cmp	Indicates that the original JPEG 2000 compression of the dataset is compliant to NPJE.
Orig TPJE Cmp	Indicates that the original JPEG 2000 compression of the dataset is compliant to the TPJE preferred encoding. Note that technically speaking, an EPJE or NPJE codestream can also be said to be compliant to TPJE. To avoid confusion, the comment used should reflect the most restrictive encoding that accurately describes the codestream.
Orig Other Profile Cmp	Indicates that the original JPEG 2000 compression of the dataset is compliant to a preferred encoding other than the TPJE, EPJE, or NPJE encodings.
Mask Removed	Indicates that at the time of JPEG 2000 compression the block and/or pixel mask information present in the BIIF, NSIF, or NITF file was removed.
Mask Added	Indicates that at the time of JPEG 2000 compression the block and/or pixel mask information present in the BIIF, NSIF, or NITF file was retained, in whole or in part. Also, this comment is used to indicate when a block or pixel mask is added to a JPEG 2000 codestream that wasn't present in the original file. This comment is provided for future compatibility since currently, block and pixel masking are not allowed with JPEG 2000-compressed BIIF, NSIF, and NITF files.
Parsing Involves Multiple Codestreams	Indicates that any parsing performed on the original BIIF, NSIF, or NITF dataset(s) involved the handling of pixel data from multiple codestreams.
Note: Multiple qualifying event comments can be concatenated using a BCS-A "+" symbol. *Note: Highlighted fields are currently accepted comments repeated from Table F-20.*	

Appendix G LVSD Preferred JPEG 2000 Encoding (LPJE)

LPJE (LVSD Preferred JPEG 2000 Encoding) defines the preferred JPEG 2000 compression encoding for Large Volume Streaming Data sensors (LVSD). LVSD is a new NATO designation for sensors that collect large arrays of image samples. Typical LVSD image arrays may be comprised of 10 Mpixel per image frame upwards to 1 Gpixel per frame and larger. To create such large arrays, LVSD systems may utilize one or more large framing cameras that are mosaic together into one large image frame or they may keep each camera's imagery distinct and dynamically mosaic the data together during display.

Furthermore, LVSD systems are typically used in a persistent mode. Imagery is collected from the camera(s) at rates from one frame-per-second (1fps) and faster. Collections can last several hours which leads to a large volume of data being collected. JPEG 2000 has been selected as one method to compress LVSD data because it provides multi-resolution and region-of-interest compression/decompression and a companion interactive streaming protocol (JPEG 2000 Interactive Protocol (JPIP), ISO/IEC 15444-9:2005). This annex deals only with the compression of individual LVSD frames (mosaic or not) using JPEG 2000. For standards regarding other aspects of LVSD systems (other sensor types, compression and file formats, metadata formats, etc.) see the current edition of STANAG 4609 (Edition 3).

LVSD systems might require hardware-assisted compression of imagery frames to maintain desired throughput during collection. LVSD must therefore accommodate both hardware and software encoders and decoders. Oftentimes however, hardware-assisted compression and software-based decompression systems are designed to meet different requirements, and when used together may suffer performance issues. Designing an LVSD system that maintains desired collection throughput, while also providing smooth playback of compressed data for dissimilar codec's is a challenge.

LPJE supports the performance requirements of LVSD systems. LPJE is a superset of the ISO/IEC 15444-1 Profile-1, NPJE, EPJE, APJE and SPJE encodings, which means that any JPEG 2000 codestream compressed to any preferred encoding in BPJ2K01.10 is also an LPJE compressed image. LPJE, however, offers a wider range of compression options than any of the other preferred encodings. *It is possible to create LPJE compliant files that lie outside the boundaries defined within Profile-1 in JPEG 2000 Part 1 (see Appendix C of this document); therefore, strictly NSIF01.01 compliant systems will not be able to decode all possible LPJE files*.

It is recommended that LPJE be used to compress LVSD imagery and further that LVSD imagery be left as a raw JPEG 2000 codestream. The NSIF file format may be used with LVSD systems, but raw JPEG 2000 codestreams are preferred with metadata transport happening through the AAF/MXF file format. The AAF/MXF file format is recommended (see MISB RP 0301.3) as the means to directly (or indirectly) transport LVSD compressed imagery and its associated metadata. Nevertheless, some systems are using NSIF (or NITF)-wrapped JPEG 2000. If an LVSD implementation wishes to use the NSIF (NITF) file format to encapsulate JPEG 2000 codestreams, it may do so;

however, the implementation should still use MXF/AAF along with the proper KLV-based metadata structures to transport and store the imagery (either JPEG 2000 codestreams or NSIF (NITF) files) and associated metadata.

Note STANAG 4609 (Edition 3) points to the BIIF Profile for JPEG 2000 01.00 Amendment 1 for compression guidelines. This document replaces Amendment 1, which was not published. AEDP-8 (Edition 3) reproduces portions of this profile related to LPJE. This was done to get LPJE out as quickly as possible. In the event of any discrepancies between AEDP-8 and BPJ2K01.10, BPJ2K01.10 takes precedence. Future versions of STANAG 4609 and AEDP-8 will reference this document.

It is possible that an NSIF01.01 encoder might wish to create codestreams that do not fit any of the preferred encodings in BPJ2K01.10 except LPJE. This might be done to utilize JPEG 2000 encoding features or parameters that are not allowed under the other encodings. Let us assume that these codestreams are constrained, however, to be ISO/IEC 15444-1 Profile-1 compliant. Thus the specific implementation’s NSIF01.01 encoder can create an LPJE compliant codestreams, and the corresponding NSIF01.01 decoder need not support the full range of LPJE encodings. In this case, the implementation is not considered Appendix G compliant. The implementation’s NSIF01.01 decoder would be able to decode such codestreams, however, since they would be ISO/IEC 15444-1 Profile-1 compliant.

G.1 JPEG 2000 Profile and Limitations

Section 7 contains the codestream restrictions defined in ISO/IEC 15444-1, Profile-1. All compliant NSIF decoders are required to properly decode compressed data within the limits of that profile. The LPJE preferred encoding defined in this Annex allows for compressed data that does not fit within the limitations of ISO/IEC 15444-1, Profile-1. Therefore, BPJ2K01.10 decoders that are not compliant to LPJE might not be able to correctly interpret all LPJE codestreams. Any LPJE codestream, however, that conforms to the limitations of ISO/IEC 15444-1 Profile-1, will be correctly interpreted by all compliant BPJ2K01.10 decoders. Implementers are cautioned to exercise care when creating LPJE codestreams that do not fit within ISO/IEC 15444-1 Profile-1. Interoperability should always be a primary concern and LPJE codestreams should conform to Profile-1 of ISO/IEC 15444-1 wherever possible.

All LPJE compliant encoders must correctly set the capability parameter, Rsiz, within the SIZ marker segment (see Table G-2). If an LPJE codestream meets the Profile-1 restrictions of ISO/IEC 15444-1, it shall be so indicated through proper setting of the Rsiz parameter. This will allow all BPJ2K01.10 compliant decoders to decode that particular codestream. Codestreams that do not fit within Profile-1 of ISO/IEC 15444-1 will be marked to indicate that decoding resources beyond Profile-1 are required to properly interpret those codestreams. This will alert other BPJ2K01.10 compliant decoders that they may not be able to properly decode these codestreams.

All compliant LPJE decoders shall be able to properly decode compressed data within the limits of LPJE. Any LPJE compliant decoder will be capable of decoding all BPJ2K01.10 compliant codestreams. All LPJE compliant encoders must produce compressed data that is within the limits of the LPJE preferred encoding. It is recommended that encoders adhere to the recommendations in this Appendix. There is *no requirement* that LPJE compliant encoders be able to produce other BPJ2K01.10 compliant codestreams (NPJE, EPJE, TPJE or SPJE), although this functionality is desired and should be readily achieved with software-based encoders.

G.2 LPJE Overview

LPJE was developed to address the wide array of needs for persistent surveillance systems. To better understand the design choices behind LPJE we consider two specific applications that represent different requirements with regard to processing timelines, memory constraints and performance metrics. These applications employ on-board/embedded hardware-assisted compression and high performance software visualization and streaming. Throughout the following discussion these applications provide a reference that will indicate certain tradeoffs. Designing an LVSD system that serves both example application needs can be challenging.

G.2.1 LPJE Requirements

The necessary features for managing LPJE compressed data are: Resolution scalability, quality scalability, parsing and chipping, region-of-interest encoding, fast roam and zoom at reduced resolution, fast hardware-assisted compression, and small memory footprint for encoding and decoding. It is important to realize that the particular JPEG 2000 codec (encoder/decoder) implementation can be the dominating factor that affects system performance. The following discussion looks at some of the above features with regard to two application scenarios; hardware-assisted compression and high performance software visualization. It is meant only to illustrate some of the tradeoffs that implementers should consider for LVSD systems.

G.2.1.1 Resolution Scalability

Resolution scalability in JPEG 2000 is enabled through repeated application of the wavelet transform. This decomposes an image into a hierarchy of resolution levels. Given the large aggregate frame size of LVSD sensors it is important to include a sufficient number of wavelet transform levels to enable reduced resolution (thumbnail) viewing of an entire scene. Consider a 100 Mpixel aggregate frame size image (10,240 x 10,240 pixels) with eight levels of wavelet decomposition. In this case, the R1 (first) level image resolution is 5,120 x 5,120 pixels; the R2 (second) level image resolution is 2,560 x 2,560 pixels; the R3 (third) level image resolution is 1,280 x 1,280 pixels, etc. The R8 (subsampled by a factor of 256 in the row and columnar directions) version of this image is then 40 x 40 pixels. Here, R8 represents a reasonable thumbnail version of the image. If only five levels of wavelet transform are available—as prescribed by NPJE, the R5 level image resolution of 320 x 320 pixels is too large for thumbnail purposes and further sub-sampling would be needed.

When designing hardware, minimization of the amount of computation is desired. In general there is a diminishing return in compression efficiency as the number of wavelet transform levels is increased. This suggests that the number of wavelet transform levels should be kept below a point at which compression gains are no longer significant with additional wavelet processing. On the other hand, if a hardware encoder performs too few decomposition levels, subsequent software applications may be forced continually to subsample the lowest resolution level to meet thumbnail display constraints.

JPEG 2000 allows images to be broken into a regular rectangular array of sub-images called tiles. Tiles are encoded independently (compressed tile data can be broken into “tile-parts” that can be multiplexed together) and provide an easy way to break large images into manageable smaller pieces. Tiles are especially attractive for hardware implementations. Resolution scalability can be impaired by employing tiles that are too small. Small tiles put an upper bound on the number of wavelet transform levels since it is not possible to subsample a tile beyond a 1 x 1 sample. In the above example, if 64 x 64 tiles are used, we only have access to an R6 level image (160 x 160 pixels) if the maximum number of transform levels are used. In practice, this should not be done. Running the wavelet transform past the point where tiles are subsampled to approximately 32 x 32 or 64 x 64 pixels in size does not improve compression performance. Therefore, tile size should be large enough to allow a sufficient number of wavelet transform levels to generate an “appropriate” lowest resolution level. The tile size must also be balanced against memory and processing costs.

G.2.1.2 Quality Scalability

Arguments similar to that for resolution scalability can be made for quality scalability. Applications need a sufficient number of codestream layers for optimal bandwidth and memory management across such a wide range of resolutions. Layers provide an elegant means to control visual quality and manage channel capacity for streaming compressed imagery over low bandwidth links. Layer generation necessitates that an encoder maintains a measured amount of codestream statistics and process multiple passes through the entropy encoded data. This translates into computation and memory costs for the encoder. NPJE recommends 19 to 20 layers for sufficient bandwidth management across all resolution levels, which is a good starting point for implementers developing a layering scheme. LPJE does not require a specific layer structure since the goal of this preferred encoding is to accommodate hardware and software implementations.

Without layers, libraries that parse JPEG 2000 codestreams and servers that employ JPIP (JPEG 2000 Interactive Protocol, ISO/IEC 15444-9:2005) streaming of JPEG 2000 compressed imagery have more work to do. Decisions on which compressed data should be retained or streamed to meet a desired image quality for a given file size or channel bandwidth must be made. Furthermore, if lossy compression is employed, access to the image at its original quality or fidelity might not be possible. In this case only educated guesses can be made at how to parse the compressed data into new files or streams without ancillary information. Encoders need to create sufficient quality layers at appropriate bitrates to facilitate parsing across all resolution levels.

G.2.1.3 Parsing and Chipping

Parsing and chipping regions of interest within an image to create new files or during JPIP streaming sessions is a very common task. Design tradeoffs with regard to parsing and chipping will affect other performance metrics. In particular, reduced-resolution roam and zoom are sensitive to choices made here. JPEG 2000 offers two methods to enable region-of-interest parsing and chipping: Tiles and precincts. The interactions between resolutions, layers, tiles and precincts, various JPEG 2000 pointer marker segments (TLM, PLT, etc.), and computer disk access patterns is complex. These interactions, however, can determine application performance and memory usage. The NPJE and EPJE preferred encodings will serve to aid in understanding this.

EPJE was introduced to improve reduced-resolution roam and zoom relative to NPJE. The differences between EPJE and NPJE are small and it is possible to "repackage" (see below) the two with a minimal amount of effort. The difference between EPJE and NPJE is in the disk access patterns required to parse out a reduced-resolution version of an image. In EPJE the portions of the codestream needed to extract a reduced-resolution image tend to be located close to each other within a compressed file. Thus, when the computer operating system reads data off its disk drive the needed information is obtained through a minimum number of seeks. This is not true for NPJE. With NPJE, the operating system must seek to multiple non-contiguous locations within the file. This significantly slows reading the data into memory and impairs performance. Refer to Appendix E for further discussions on EPJE and its codestream design.

This problem remains in LVSD systems. In fact, with increased frame sizes, it becomes a serious concern. To improve application performance, LPJE allows the use of precincts. Precincts can be thought of as tiling within the wavelet transform subbands. They offer a good mechanism to collect spatially related codestream data together over multiple wavelet transform levels. Proper use of precincts and JPEG 2000 progression orders can greatly improve reduced-resolution roam and zoom performance. In fact, precincts typically perform better than tiles in this regard. Precincts do not help tile-based chipping operations; "transcoding" (see below) is required. Precincts also aid greatly in memory management during the compression of very large frame imagery.

From a software perspective, precincts can improve performance without the need for tiling in certain applications. They are a new concept, however, and it is quite easy for software implementations to "get it wrong". Precincts must be properly understood and implemented to realize the potential performance improvement. Tiles, on the other hand, are a simple concept and are easily implemented. No processes or codestream structures within JPEG 2000 Part 1 spans tile boundaries (this is not true for precinct boundaries). Tiles allow larger images to be cut into smaller subsections that are independently processed. Each tile can be processed by a separate software thread or hardware ASIC. Precincts can also be processed in an independent fashion in software, although, designing hardware to support precinct processing is difficult. Choosing to use tiles or precincts will depend on the application scenario. There are arguments one might make for using both (large tile size with precincts inside the tiles). It is likely that both precincts and tiles will be utilized within LVSD systems. LPJE will allow both of these constructs.

G.2.1.4 Region of Interest Encoding

LPJE allows the use of the RGN marker segment thereby giving encoders the capability to perform region of interest encoding. In JPEG 2000 it is possible to preferentially encode spatial regions so that this information appears first in the codestream layers. This has the effect that when the codestream is sent via JPIP to a client viewer application, the region of interest (ROI) is received first. Thus, ROI encoding can be used to preferentially allocate bandwidth to one or more ROIs over background image regions. This strategy can substantially reduce bandwidth provided the background image quality can be sacrificed.

For example, an image can be losslessly encoded with an ROI. The codestream data corresponding to the ROI is placed in the first layer of the codestream. This layer can be delivered at a lossless image quality to a client application without any (or very little) background image data. This can be achieved at a very high effective compression ratio (possibly 100:1 or higher). If the client application waits to receive all of the subsequent layers in the codestream, the entire image is recovered losslessly. Often, all that is required of the background is to put an ROI "in context". In these situations ROI encoding is a powerful tool. The choice to decide when enough background image quality has been received is left to the client application. Algorithms that autonomously define the ROIs for the JPEG 2000 encoder are of course outside the scope of this standard.

G.2.2 JPEG 2000 Repackaging and Transcoding

The concepts of JPEG 2000 "repackaging" and "transcoding" were discussed in section A.7. The difference between repackaging and transcoding is a matter of degree. Repackaging refers to simple codestream manipulations that do not require decoding of any entropy encoded wavelet coefficients (marker segment and packet header manipulation may be necessary). Transcoding refers to more sophisticated codestream manipulations where entropy coding of wavelet coefficients might be changed and possible recalculation of wavelet coefficients may occur.

LPJE allows a much broader range of compression options to be selected compared to NPJE and EPJE. As such, there are more opportunities for the need of sophisticated codestream manipulation. Adding tiling to a codestream that was not tiled requires transcoding. Wavelet coefficients must be recomputed at the tile boundaries. Adding precincts to a codestream that does not use precincts or altering the precincts in a codestream that uses them requires processing somewhere between repackaging and transcoding. There is *no* requirement for LPJE encoders, decoders, or JPIP servers to maintain compliance to this Appendix after repackaging or transcoding procedures.

G.3 LPJE Codestream Structure

The LPJE JPEG 2000 main header contains markers and marker segments as shown in Figure G-1. Note that only the SIZ marker segment has a fixed placement, all other marker segments can appear in any order within the main header. The SIZ, COD and QCD marker segments are required in the main header by ISO/IEC 15444-1. All other marker segments listed are allowed, but may not be recommended for use. See Table G-1 for a full listing of marker segments and where they may be used within LPJE.

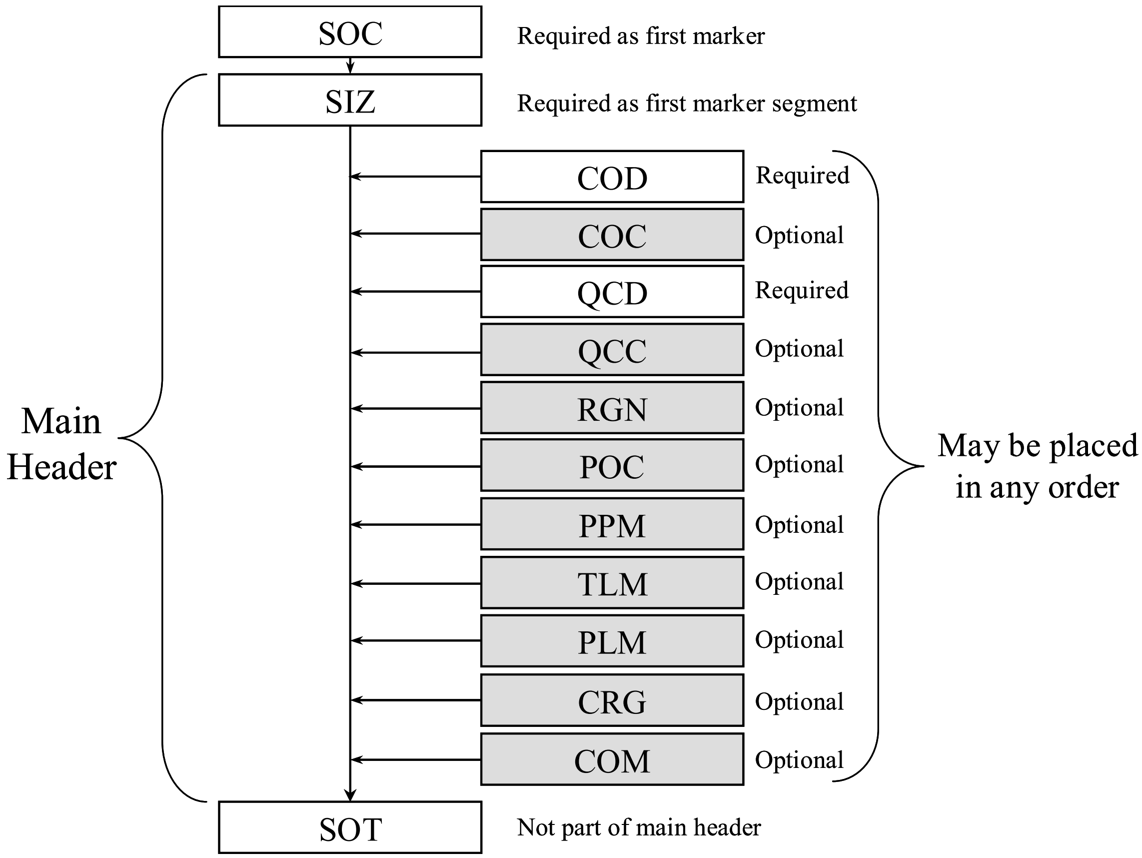

Figure G-1 Layout of LPJE JPEG 2000 main header

Certain main header marker segments can be overridden by other marker segments within the main and tile-part headers. For example, a tile-part header COD or COC marker segment or a main header COC marker segment will override a main header COD marker segment. This behavior allows specific tiles, components or tile-components to be encoded differently than all other components in the codestream. Improper usage of main header and tile-part header marker segments to override encoding defaults can lead to inefficient (but syntactically correct) signaling within the codestream. It is recommended that LPJE encoders properly maintain and minimize the number of override marker segments consistent with correctly describing the codestream.

As an example, consider an image with six tiles. In one of these tiles we override the main header COD and QCD with a tile-part header COD and QCD. Within this particular tile, the tile-part COD and QCD will dictate how the codestream is formed and the main header COD and QCD are not used. An efficient way (i.e. one that minimizes file overhead) to signal this in the compressed file is to have a main header COD and QCD that applies to the five other tiles and for the sixth tile a tile-part header COD and QCD are included. A less efficient way to signal this (but still syntactically correct and compliant to ISO/IEC 15444-1) is to place tile-part COD and QCDs in each first tile-part header. This places redundant information in the file that could be carried by the main header alone. One application where carrying the redundant markers makes sense would be transmission in a noisy environment. In this case repeating the COD and QCD marker segments in a tile-part header could help guard against bit errors in the main header. There are other error resiliency techniques that might be employed in noisy environments as well.

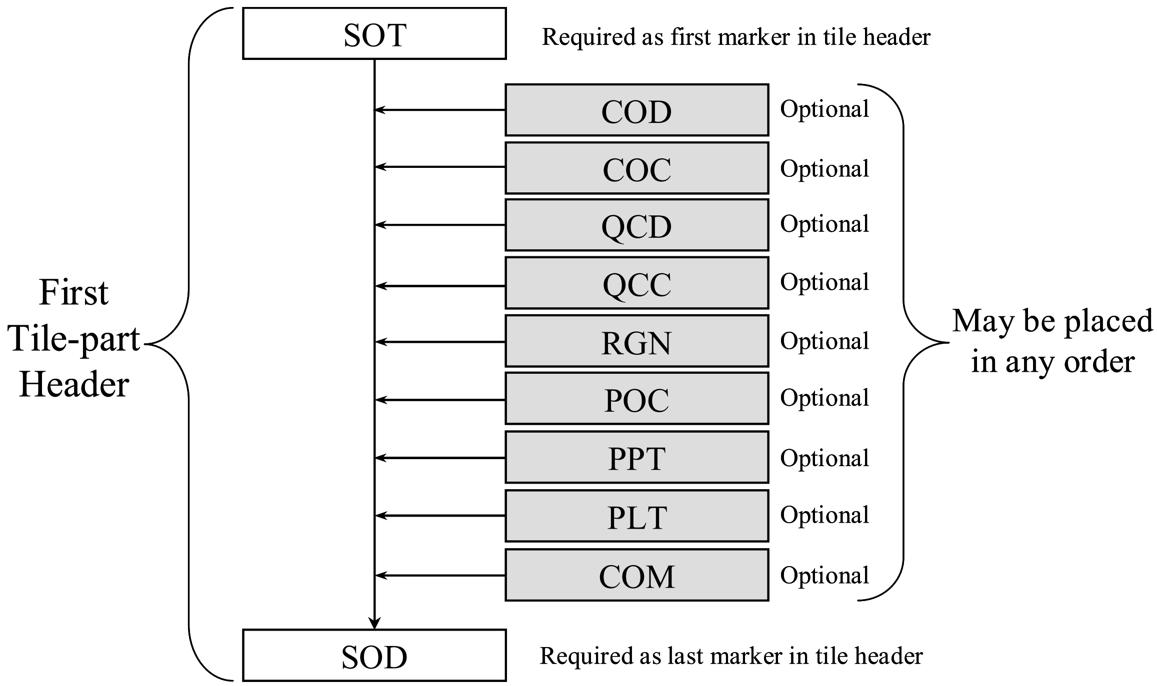

Figure G-2 Layout of a single JPEG 2000 tile-part header (first tile-part only)

Figure G-2 shows a JPEG 2000 tile-part header (first tile-part only) for LPJE. The structure of other tile-part headers (not first tile-part) is given in Figure G-3. All marker segments are optional. Refer to Table G-1 for a full listing of marker segments and where they may be used within LPJE as well as any additional applicable restrictions.

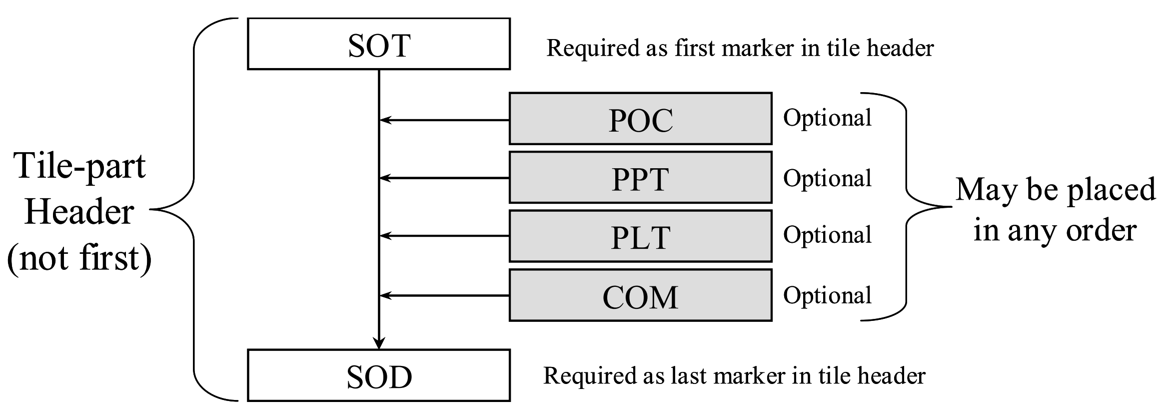

Figure G-3 Layout of a single JPEG 2000 tile-part header (not first tile-part)

G.4 Markers and Marker Segments Limits Within LPJE

Table G-1 describes the recommended usage of each marker, whether it is required (Req.), not allowed (NA), optional (Opt.), recommended (Rec.) or not recommended (NR) and if there are any restrictions or dependencies. Marker segments may occur in three places within the codestream; the main header, a tile-part header, or the bitstream. A marker segment might be required in one area, the main header for example, and not allowed in others (tile-part header and bitstream). The bitstream plus the main and tile-part headers comprise the JPEG 2000 codestream. The bitstream comprises those portions of the codestream that are not in either the main or tile-part header. It is comprised mainly of entropy encoded data with some markers present.

If a marker segment is Required (Req.) it *shall* be present as indicated. There may be additional restrictions placed on the marker segment. For example, the SIZ marker segment is required in the main header and it must be the first marker segment present after the SOC marker. If a marker segment is Optional (Opt.) it *might* be present. The decision to use the marker segment is up to the implementation. The marker segment may also be subject to restrictions (e.g. COM marker segment). Marker segments that are not recommended (NR) are *not recommended* for general use. This *does not* mean that the marker segments are *forbidden*. Instead the marker segment might be used in special circumstances. For example, the SOP and EPH marker segments can be useful in noisy communications scenarios. In this special circumstance the SOP and EPH marker segments are useful to identify and control codestream errors due to noisy channels. In general, these marker segments are not needed or recommended. Any marker segment marked Not Allowed (NA) *shall not* be used as indicated.

The encoding recommendations made in this Appendix should be compared against ISO/IEC 15444-1 Profile-1. Readers should compare Table 7-1 and Table G-1 to see the differences between the two encodings. The values column in the following tables give the full ISO/IEC 15444-1 ranges for comparison.

Table G-1 Marker and marker segment requirements within an LPJE codestream

Marker	Value	Main Header	Tile-part Header	Bitstream	Restrictions/Remarks
SOC	0xFF4F	Req.	NA	NA	Required as first marker in the file. Main headers starts immediately after SOC.
SOT	0xFF90	NA	Req.	NA	Required as first marker in each tile-part. Tile-part header occurs after SOT marker segment and before SOD marker.
SOD	0xFF93	NA	Req.	NA	Last marker in each tile-part header. Immediately precedes tile-part data.
EOC	0xFFD9	NA	NA	Req.	Required as the last marker in the codestream.

Table G-1 Marker and marker segment requirements within an LPJE codestream

Marker	Value	Main Header	Tile-part Header	Bitstream	Restrictions/Remarks
SIZ	0xFF51	Req.	NA	NA	Required as second marker segment in main header. Immediately follows SOC.
COD	0xFF52	Req.	Opt.	NA	One and only one required in main header. Optionally no more than one main appear in first tile-part header.
COC	0xFF53	Opt.	Opt.	NA	Optional. No more than one COC per component in main header and first tile-part header.
RGN	0xFF5E	Opt.	Opt.	NA	Optional. No more than one per component in main header or first tile-part header.
QCD	0xFF5C	Req.	Opt.	NA	One and only one required in main header. No more than one in first tile-part headers.
QCC	0xFF5D	Opt.	Opt.	NA	Optional. No more than one per component in main header or first tile-part header.
POC	0xFF5F	Opt.	Opt.	NA	Optional. No more than one in the main or any tile-part header. Must appear in a tile-part header before the packets which it describes. Required if there are progression order changes different from the main or tile header COD.
TLM	0xFF55	Rec.	NA	NA	Recommended unless encoding image in a single tile-part.
PLM	0xFF57	NR	NA	NA	Not recommended, but allowed. Main header only, PLT is preferred.
PLT	0xFF58	NA	Rec.	NA	Recommended. Multiple PLTs may appear per tile. Must appear in a tile-part header prior to the packets whose lengths are described.
PPM	0xFF60	NR	NA	NA	Not recommended, but allowed. Use of a PPM prohibits use of PPT and in-bitstream packet headers.
PPT	0xFF61	NA	NR	NA	Not recommended, but allowed. All packet headers for the tile in which PPT appears must be included in the PPT. Use of PPT prohibits use of PPM and in-bitstream packet headers for the tile.

Table G-1 Marker and marker segment requirements within an LPJE codestream

Marker	Value	Main Header	Tile-part Header	Bitstream	Restrictions/Remarks
SOP	0xFF91	NA	NA	NR	Not recommended, but allowed. May be used in front of each packet. Whether or not a SOP marker is used for a given packet, the Nsop parameter is incremented. If packet headers are moved into a PPM or PPT marker segment, the SOP marker segment may appear immediately before packet bodies in the bitstream.
EPH	0xFF92	NR	NR	NR	Not recommended, but allowed. If used, they must appear after each packet header. If packet headers are moved into PPM or PPT marker segments, the EPH markers are moved as well.
CRG	0xFF63	NR	NA	NA	Nor recommended, but allowed. Informational use only, implementations need not do anything with this information. Only one may appear in main header and it applies to all tiles.
COM	0xFF63	Opt.	Opt.	NA	Optional use. Informational only, no data necessary to decode the codestream or metadata needed to interpret the codestream shall be placed in a COM marker segment.

G.4.1 Delimiting Markers and Marker Segments

Each delimiting marker shall be present in all compliant JPEG 2000 codestreams. A codestream shall have only one SOC and EOC marker, and at least one tile-part. Each tile-part has one SOT and one SOD marker. The SOC, SOD and EOC markers are formatted the same as in ISO/IEC 15444-1 Profile-1 and the other BPJ2K01.10 encodings (see Table 7-2, Table 7-4 and Table 7-5). The SOT marker segment for LPJE has the same format and range of values as that found in Profile-1 (see Table 7-3). NPJE allows only one tile-part per tile. LPJE, however, allows multiple tile-parts per tile in a similar fashion to EPJE and TPJE. Unlike those two encodings, however, there is no implicit association between resolution level and tile-part within LPJE.

G.4.2 Fixed Information Marker Segment

The SIZ marker segment (see Table G-2) includes information required to properly decode the image. There shall be a SIZ marker segment in the main header immediately after the SOC marker segment.

Table G-2 Image and tile size (15444-1 Annex A.5.1)

Parameter	Size (bits)	Values	LPJE	Notes
SIZ	16	0xFF51	0xFF51	Image and tile size marker
Lsiz	16	41 – 49,190	41 – 49,190	Length of marker segment
Rsiz	16	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010 0000 0000 0000 0011 0000 0000 0000 0100	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010	Rsiz = 0000 0000 0000 0000 indicates that the full capabilities described by ISO/IEC 15444-1 are required. Rsiz = 0000 0000 0000 0001 indicates that the codestream is Profile 0 compliant. Rsiz = 0000 0000 0000 0010 indicates that the codestream is Profile-1 compliant. Rsiz = 0000 0000 0000 0011 and Rsiz = 0000 0000 0000 0100 see below*
Xsiz	32	$1 – (2^{32}– 1)$	$1 – (2^{32}– 1)$	Reference grid width
Ysiz	32	$1 – (2^{32}– 1)$	$1 – (2^{32}– 1)$	Reference grid height
XOsiz	32	$0 – (2^{32}– 2)$	$0 – (2^{32}– 2)$	Horizontal offset from the origin of the reference grid to the left side of the image area
YOsiz	32	$0 – (2^{32}– 2)$	$0 – (2^{32}– 2)$	Vertical offset from the origin of the reference grid to the top of the image area
XTsiz	32	$1 – (2^{32}– 1)$	$1 – (2^{32}– 1)$	Tile width on reference grid. LPJE recommends tile sizes be no smaller than 256 and no larger than 8,192.
YTsiz	32	$1 – (2^{32}– 1)$	$1 – (2^{32}– 1)$	Tile height on reference grid. LPJE recommends tile sizes be no smaller than 256 and no larger than 8,192.

Table G-2 Image and tile size (15444-1 Annex A.5.1)

Parameter	Size (bits)	Values	LPJE	Notes
XTOsiz	32	$0 – (2^{32} – 2)$	$0 – (2^{32} – 2)$	Horizontal offset from the origin of the reference grid to the left edge of the first tile
YTOsiz	32	$0 – (2^{32} – 2)$	$0 – (2^{32} – 2)$	Vertical offset from the origin of the reference grid to the top edge of the first tile
Csiz	16	1 – 16,384	1 – 16,384	The number of components in the image.
$Ssiz^i$	8	0000 0000 – 1010 0101	Unsigned: 0 – 37 Signed: 128 – 165	0xxx xxxx Unsigned data 1xxx xxxx signed data x000 0000 – x010 0101 data bit depth = value + 1
$XRsiz^i$	8	1 – 255	1 – 255	Horizontal sub-sampling on the reference grid with respect to the i^{th} component
$YRsiz^i$	8	1 – 255	1 – 255	Vertical sub-sampling on the reference grid with respect to the i^{th} component

* Two new values for the Rsiz parameter have been added in ISO/IEC 15444-1 Amendment 1, 2006. These values were assigned to the Digital Cinema Initiative to indicate which DCI profile (2K or 4K) the codestream adheres to. See the document "Digital Cinema System Specification**,** Version 1.1"**,** April 12, 2007 for further details. LPJE implementations *shall not* set Rsiz = 0000 0000 0000 0011 or Rsiz = 0000 0000 0000 0100.

The Digital Cinema Initiative has defined two new values for Rsiz. These values are given in Amendment 1 of JPEG 2000 Part 1. These values indicate that the JPEG 2000 codestream complies with the 2K (Rsiz = 3) and 4K (Rsiz = 4) DCI profiles. While both DCI profiles fall within the set of all possible LPJE codestreams, LPJE compliant encoders shall not set Rsiz to the DCI values. If an LPJE codestream happens to be compliant to either DCI profile (this is unlikely, but possible), it shall be marked Rsiz = 0 (NR profile) or Rsiz = 2 (Profile-1) as appropriate.

It is recommended that tile sizes for LPJE be restricted such that the *effective* tile size should be no smaller than 256 x 256 or larger than 8,192 x 8,192. The effective tile size, ($tcx1 – tcx0$, $tcy1 – tcy0$) see Annex B ISO/IEC 15444-1, is computed by taking into account sampling on the reference grid. For $XRsiz^i = YRsiz^i = 1$, this means that $256 \leq XTsiz \leq 8{,}192$ and $256 \leq YTsiz \leq 8{,}192$. For sampling factors $XRsiz^i$ and $YRsiz^i$ greater than 1, XTsiz and YTsiz must be correspondingly increased. For most applications this means that XTsiz and YTsiz will have a lower bound of 256 and an upper bound of 8,192 x $XRsiz^i$ and 8,192 x $YRsiz^i$. An exception to this rule occurs if the original image size is smaller than 256 x 256 and $XRsiz^i = YRsiz^i = 1$. In this case the image should be encoded as a single tile.

The recommended tile sizes of LPJE are *not* requirements. Applications can create smaller or larger tiles with the understanding that they might impair encoder performance and interoperability. The recommended lower bound of a 256 x 256 tile was chosen to accommodate hardware encoders that rely on tiling to reduce complexity. Encoders must not generate tiles that are too small and impair reduced-resolution roam and zoom performance. The recommended upper bound on tile size of 8,192 x 8,192 pixels was chosen to limit the memory requirements for encoders and decoders. While efficient software encoder and decoder implementations exist that can handle very large tiles (100,000 x 100,000 pixels or 10 Gpixel single tile images have been created), not all implementations can be relied upon to be this efficient. Implementers are urged to verify the efficiency of their JPEG 2000 codecs when attempting to implement tile sizes larger than 8,192 x 8,192 in size.

Hardware implementations may wish to limit their tile sizes to "powers of two" for ease of implementation, for example $256 \times 256 = 2^8 \times 2^8$, or $512 \times 512 = 2^9 \times 2^9$. Hardware implementations are still required to handle odd size tiles, which might occur along the borders of the image whenever the tile size does not evenly divide into the image dimensions. LPJE *does not allow* the use of "padding" along the borders of an image to make the image dimensions meet some dimension constraint. If there is a desire to encapsulate multiple sensor modalities together, for example an IR and EO framing sensor that image the same field of view but at different resolution, it is best to handle these sensors as separate JPEG 2000 codestreams. The encapsulation should be performed at a higher file format level (*e.g.* NSIF, MXF, *etc.*).

The number of decomposition levels chosen during encoding, N_{Levels}, should not "exhaust" the nominal tile dimensions. In other words, the number of wavelet transform levels should not generate empty subbands (*i.e.* subbands that contain no wavelet coefficients) for nominal (full size) tiles. Tiles on the border of the image may not be full size and empty subbands may occur in these tiles. All JPEG 2000 implementations must properly handle this case. If we consider the simple case of a one component image with no reference grid image or tile offsets and no reference grid sampling, this requirement can be explicitly stated as follows:

$$N_{Levels} \le \lfloor \log_2(\min(Xsiz, Ysiz)) \rfloor \qquad \textbf{Equation G-1}$$

Note that **Equation G-1** becomes more complex when reference grid offsets and sampling factors must be considered. Implementers should consult Annex B of ISO/IEC 15444-1 to fully understand these issues.

G.4.3 Functional Marker Segments

The functional marker segments define what parameters were used in the compression of a given tile or an image. These marker segments apply to the entire tile when in the tile header and to the image when in the main header. Markers in a tile-part header supersede markers in the main header.

G.4.3.1 Coding Style Marker Segments (COD and COC)

Table 7-7 gives the COD marker segment. LPJE allows the full range of coding options to be used when generating codestreams. This includes all progression orders, code-block sizes, wavelet transforms and multiple component transforms. No limits are placed on precinct sizes or the number of layers. The COD marker segment is required in the main header of the codestream and it contains the default encoding parameters applied to all components and tiles unless it is overridden (see COC below).

LPJE encoders should include enough layers to enable quality scalability across all resolutions. Approximately 10 – 20 layers suffice for a monochrome image. Hardware implementations may wish to use less. The maximum number of layers should not exceed 50 unless special circumstances exist (i.e. a large number of components and/or decomposition levels).

Code block sizes in JPEG 2000 are limited to a maximum number of 4,096 coefficients within the code-block. Furthermore, the minimum dimension of a code-block in the row or columnar dimension is four. Code-blocks represent the fundamental limit on spatial region of access within the JPEG 2000 codestream (barring any code-block/precinct interactions). Code-blocks also prevent error propagation in the entropy encoder. Small code-blocks, therefore, would seem like a good idea. The larger the code-blocks are, however, the greater the entropy coding efficiency of the arithmetic encoder will be. In general, code-blocks of size 32 x 32 or 64 x 64 are good choices. Hardware implementations may want to use the smaller code-block size to minimize memory costs. Arguments can also be made for using rectangular (4 x 1,024) code-blocks when performing stripe processing on very large images.

JPEG 2000 allows for several options regarding “code-block style” (see Table 7-10). These parameters are used to control the behavior of the arithmetic encoder. Annex D in ISO/IEC 15444-1 describes the meaning of these parameters. In general, these options relate to speeding up the arithmetic encoding (selective bypass) and improving error resiliency/detection (segmentation symbols, termination, resetting context probabilities, vertically causal). Some of the above options also reduce memory costs to a small extent. For more discussion on these options see Annex J in ISO/IEC 15444-1 as well.

The COD coding parameters can be overridden. A COC marker segment (see Table 7-12) can be used to override the coding parameters for a single specified component. The NPJE and EPJE encodings do not recommend the use of the COC marker segment, but they are allowed within LPJE. LPJE allows all possible coding options in a COC marker segment.

To override the coding parameters for more than one component, multiple COC marker segments must be used (to override coding parameters for all components in a tile, a tile-part COD marker segment can be used). If the COC marker appears in the main header, then the default coding parameters (as defined by the main header COD marker segment) for the specified component are replaced by those in the COC marker segment for the entire image. If the COC marker segment appears in a first tile-part header, then the coding parameters for the specified component are replaced for that tile only. We can express the relationships between main and tile-part COD and COC marker segments as follows:

Tile-part COC > Tile-part COD > Main Header COC > Main Header COD

This illustrates that tile-part COC marker segments supersede tile-part COD marker segments, which supersede main header COC marker segments, which supersede the main header COD marker segment.

Other BPJ2K01.10 encodings (NPJE, EPJE and TPJE) strictly control the encoding options that can be used. LPJE does not. Limiting the encoder's compression options would prevent LPJE from serving the wide array of conops it is meant to address. While this flexibility is necessary for LPJE systems, it does mean that implementers of LPJE systems will need to carefully choose their compression options to maximize interoperability. Transcoding services may be necessary to interact with older systems that use less capable JPEG 2000 codecs.

G.4.3.2 Region of Interest Marker Segment

The region of interest marker segment (RGN) is shown in Table 7-14. This marker segment is used for region of interest coding. Encoders can shift the wavelet coefficients corresponding to a spatial region of interest up (by left bit-shift) above the most significant bitplane of the remaining background wavelet coefficients. This has the effect that these wavelet coefficients will be encoded first before all other wavelet coefficients. The RGN marker segment alerts the decoder that this has been done so that the process can be reversed during decoding. ROI encoding will most likely be a feature of some LVSD systems so it has been included in LPJE.

G.4.3.3 Quantization Default Marker Segments (QCD and QCC)

The QCD marker segment (see Table 7-15) is required in the main header to indicate the quantization step-sizes (for the 9-7I wavelet) or reversible dynamic range (for the 5-3R wavelet) that is valid for all tile-parts. LPJE allows all quantization options including scalar derived quantization for the 9-7I wavelet transform (not allowed in NPJE, EPJE or TPJE), but it is not recommended. LPJE recommends scalar expounded quantization for the 9-7I wavelet transform. Two guard bits are also recommended for all applications, but applications can use more or less if it is deemed necessary.

The QCD marker segment applies to all tiles and components unless it is overridden by a QCC marker segment (see Table 7-19). The QCC marker segment can be used to override the quantization parameters for a single specified component. To override the quantization parameters for more than one component, multiple QCC marker segments must be used (to override coding parameters for all components a tile-part QCD marker segment can be used). If the QCC marker appears in the main header, then the default quantization parameters (as defined by the main header QCD marker segment) for the specified component are replaced by those in the QCC marker segment for the entire image. If the QCC marker segment appears in a first tile-part header, then the quantization parameters for the specified component are replaced for that tile only. We can express the relationships between main and tile-part QCD and QCC marker segments as follows:

Tile-part QCC > Tile-part QCD > Main Header QCC > Main Header QCD

LPJE allows the full range of parameter values for the QCC marker segment subject to the same recommendations made for the QCD marker segment.

G.4.3.4 Progression Order Change Marker Segment (POC)

The POC marker segment (see Table 7-20) is used to change progression orders within a codestream. POC marker segments require a good understanding of JPEG 2000 codestream construction and a sophisticated JPEG 2000 codec. ISO/IEC 15444-1 Profile-1 compliant decoders are required to handle POC marker segments. The POC marker segment allows full control over the ordering of codestream data within a file. For most applications, the POC marker segment is not necessary; one progression order will suffice for all codestream data. The POC marker segment is allowed for optional use within LPJE. NPJE and EPJE do not allow its use.

The POC marker segment is another approach that may be used to solve the reduced-resolution roam and zoom problem. In fact, the POC marker segment offers a finer degree of control than the tile-part solution used in EPJE. The Digital Cinema Initiative (DCI) has prescribed the use of a POC marker segment within their 4K codestream distributions (see the DCI System Specification). The POC marker segment is used to organize 4K codestreams so that it is easier for 2K compliant systems to parse out a 2K version of the codestreams.

G.4.4 Pointer Marker Segments

The pointer markers segments are used to gain quick access to data for parsing, chipping, and decoding. These marker segments define either the lengths of certain codestream constructs or pointers to the start of codestream constructs.

G.4.4.1 Tile-part Lengths Marker Segment (TLM)

The tile-part lengths marker segment (see Table 7-21) has the same length information as the start of tile marker segments in each tile-part, but this information is collected up front in the main header. This marker segment can be used to quickly access and chip a given tile or set of tiles in a compressed image. The LPJE preferred encoding recommends the use of the TLM marker segment unless there is only one tile-part for the entire image (i.e. single tile, single tile-part). In this case there is no point in including the marker segment since there is only one tile-part length and it is contained in the tile-part header.

NPJE recommends one TLM marker segment to carry the tile lengths (NPJE allows only one tile-part per tile). In EPJE one or more TLMs is recommended to convey lengths for all tile-parts (EPJE breaks each tile into resolution-based tile-parts). LPJE allows the use of multiple TLMs. To minimize overhead, it is recommended that the minimum number of TLMs should be used to carry the tile-part lengths. It is important to remember that the tile-part structure of LPJE can be arbitrary. In other words, an application is free to split tiles into multiple tile-parts as it sees fit. Implementers should not assume that a tile-part in LPJE equates to something like a resolution or quality level. Some restraint should be exercised when creating tile-parts, however. It is best to break a tile's bitstream at logical locations, say the end of a packet rather than at an arbitrary point within a packet.

G.4.4.2 Packet Lengths Marker Segment, Main Header and Packet Lengths Marker Segment, Tile-part Header (PLM and PLT)

The entropy-coded wavelet coefficient data in JPEG 2000 are organized into packets comprised of a packet header and packet body. Both the packet header and packet body are entropy-coded and have variable length. To locate a desired piece of codestream data, a decoder must parse and decode the packet headers to learn the lengths of the entropy-coded packet bodies which contain the wavelet coefficient information. The packet length marker segments (PLM and PLT, see Table 7-23 and Table 7-25 respectively) collect together the lengths of the packets so that a decoder may rapidly skip through them to find the packets it wants without having to decode all intermediate packet headers.

The PLM marker segment collects packet length information across all tile-parts and places the lengths in the main header. It is possible to overload the PLM marker segment due to the size of the $Nplm^i$ field being 8 bits. If there are more than 255 packets in a tile-part or the total amount of $Iplm^{ij}$ length data for a given tile-part requires more than 255 bytes, then the PLM marker segment shall not be used (see discussion in ISO/IEC 15444-1:2004, Annex A.7.2). The amount of information contained in a PLM (or PLT) marker segment can be considerable. Therefore, the length data might be spread across more than one PLM or PLT marker segment. The Zplm (or Zplt) parameter is an index that is used to reassemble the data into the correct order.

The PLT marker segment performs a similar function to the PLM marker segment, except the length information is embedded in the tile-part headers. The PLT need not be placed in the first tile-part header for a tile, but it must appear in a tile-part header prior to the packets whose lengths it contains. The PLT marker segment does not suffer from the limitations of the PLM marker segment since it does not aggregate length information across multiple tiles. Due to the limitations of the PLM marker segment, LPJE *does not* recommend the use of the PLM marker segment. The PLM marker segment is allowed within LPJE and may be preferred in some instances (e.g. only one tile in the compressed file). The LPJE encoding recommends the use of the PLT marker segment instead.

Some NPJE implementations may require a PLT marker segment for each layer in the codestream. This is an unnecessary practice; all of the lengths could be combined into one single PLT. LPJE does not prohibit this and to maintain compatibility with NPJE it is allowed. Additionally, the EPJE encoding requires a PLT per tile-part describing the packet lengths within each tile-part. Again, this is an unnecessary complication, but it too is allowed in LPJE to maintain compatibility with EPJE. JPEG 2000 implementations should be sophisticated enough to properly sort the length information contained within PLT marker segments and associate it with the correct packets. ISO/IEC 15444-1 allows use of both PLM and PLT in the same codestream; this is not recommended in LPJE.

G.4.4.3 Packed Packet Headers Marker Segment, Main Header and Packed Packet Headers Marker Segment, Tile-part Header (PPM and PPT)

There are two more marker segments that can be used to assist in the parsing of packet data. These are the PPM (packed packet headers, main header see Table 7-26) and the PPT (packed packet headers, tile-part header see Table 7-27). These marker segments aggregate the packet headers (not the packet bodies) into either the main header or tile-part headers. This allows decoders to bulk read the headers and process them instead of parsing them out of the codestream. The idea is to minimize the number of disk reads needed to parse the codestream. If a decoder has all of the packet header information for a codestream, it knows exactly how many coding passes and bytes of entropy-coded data from each code-block are present within each layer of the codestream. This allows for fast parsing decisions to be made.

If the PPM or PPT marker segments are used to relocate the packet headers, then the PLM and PLT marker segments packet length information describes only the packet bodies. It does not include the packet headers in the lengths. The presence of PPM and PPT marker segments also influences the behavior of the SOP and EPH marker segments (see section 7.6). The PPM marker segment aggregates packet header information across all tile-parts into the main header. This allows a decoder to quickly decode all packet header information at once, but it comes at a price.

The PPM marker segment loads all of the packet headers into the main header of the file. While the packet header information is necessary to understand the entropy-encoded wavelet coefficients, it conveys no image information. The packet headers are side information necessary to understand the layout of the compressed image data. The amount of packet header information can be considerable for a large image. In progressive transmission systems (*e.g.* JPIP streaming, FTP transfer) placing all of this data up front rather than letting it be distributed throughout the bitstream can affect performance. The recipient of the streamed data may have to wait an unacceptable amount of time before receiving any useable imagery data.

The PPT marker segment performs a similar function to the PPM marker segment, but it places the packet header information in the tile-part headers instead. PPT marker segments need not appear in the first tile-part header, they need only appear in a tile-part header that is located in the compressed file before the packet data that the packet headers describe. The PPT marker segment is a compromise between placing all of the packet headers in the main header and letting them be distributed through out the bitstream.

LPJE allows usage of the PPM and PPT marker segments but they are *not* recommended. In some circumstances, using the PPM and PPT marker segments has advantages. Transmission over noisy channels is one area where they might be used in an unequal bit error protection scheme. Preserving the layer contributions and packet body lengths would allow for some measure of error recovery. The PPT marker segment is preferred over the PPM marker segment since it does not front load the codestream as much. In cases where there is only one tile in the image, the PPM marker segment is as efficient as the PPT marker segment. If either marker segment is used, then the packet headers shall not appear in the bitstream data. Furthermore, if the PPM marker segment is used, the PPT marker segment shall not be used. The converse is true as well.

G.4.5 Start of Packet Marker Segment and End of Packet Header Marker Segment (SOP and EPH)

ISO/IEC 15444-1 defines an in bitstream marker segment (SOP, see Table 7-28) and marker (EPH, see Table 7-29) that can be used to improve error resiliency when operating in noisy environments. Usage of these marker segments is signaled through the COD marker segment (see Table 7-7). The SOP marker segment may be placed in front of each packet in the bitstream. The SOP marker segment has a 16 bit ring counter that can be used in the detection of missing or corrupted packets to help determine which packet is missing or corrupted. If the SOP marker segment is used, it need not appear for each packet in the bitstream. The SOP ring counter (Nsop) must be incremented for each packet, however. If PPM or PPT marker segments are used to move the packet headers, the SOP marker segment does not move. Instead it is placed in front of the packet body rather than in front of the packet header.

The EPH marker is placed in the bitstream right after the packet header and before the packet body. If the EPH marker is used, it must appear for each packet. If a packet header is corrupted, the EPH marker allows some measure of recovery by delineating the end of the packet header. This prevents the decoding of the packet header from becoming confused and interpreting the packet body as part of the packet header. If PPM or PPT marker segments are used to relocate the packet headers, then the EPH marker segments are moved along with the packet headers.

The NPJE and EPJE encodings do not recommend the use of the SOP and EPH markers. For general purposes, LPJE *does not* recommend the use of SOP and EPH. LPJE *allows* and *encourages* their use, however, in situations where noisy transmission might corrupt the codestream and there are no other system mechanisms in place to handle bit errors. Some overhead price is paid for using SOP and EPH and implementers must consider this in their system design.

Trying to recover from a bit error in a JPEG 2000 codestream requires an in-depth understanding of the codestream structures, wavelets and entropy coding. It is reasonable to assume that different codec implementations will perform differently under similar environments. It is also important to realize that where a bit error occurs can have widely varying effects on the decoded image quality. The JPEG 2000 committee has developed more robust error correction and control procedures in ISO/IEC 15444-11:2007, Information technology -- JPEG 2000 image coding system: Wireless. A future version of the LPJE preferred encoding may include this standard as an option.

G.4.6 Component Registration Marker Segment and Comment Marker Segment (CRG and COM)

The informational marker segments are not required for decoding but may assist in the interpretation of the data. LPJE *does not* recommend their use and LPJE compliant systems are not required to generate or interpret these marker segments. Component registration (CRG, see Table 7-30) allows each component to be registered to each other for display. The Comment marker (COM, see Table 7-31) allows for the unstructured data to be included into the file. It is not recommended that either of these markers be used. *No data shall be placed in a comment marker segment that is necessary to decode or properly interpret a JPEG 2000 codestream.* Neither metadata that might be used to properly interpret and exploit the compressed imagery nor any information that might be used to improve JPEG 2000 decoder performance shall be placed within a COM marker segment.

Appendix H STANAG 7023 Preferred JPEG 2000 Encoding (SPJE)

NATO STANAG (STANdardization AGreement) 7023 defines NATO's Air Reconnaissance Primary Imagery Data Standard. STANAG 7023 is NATO's primary imagery format for reconnaissance sensors and it contains a preferred encoding (see STANAG 7023 Edition 4, Annex A-10.5) to use when encoding imagery with JPEG 2000. STANAG 7023 only supports the use of raw JPEG 2000 codestreams, JP2 and JPX file formats within ISO/IEC 15444-1 and ISO/IEC 15444-2 are not allowed. STANAG 7023 defines its own file format and metadata structures.

This appendix reproduces the STANAG 7023 Preferred JPEG 2000 Encoding or SPJE. This has been done to collect all known JPEG 2000 profiles in use within the NATO community into one document. Note that the defining document for the use of JPEG 2000 within NATO air reconnaissance primary imagery collects is still STANAG 7023. In the event of any discrepancy between this description of the STANAG 7023 JPEG 2000 encoding and that contained within STANAG 7023; the description in STANAG 7023 takes precedence. A future version of the STANAG will point to BPJ2K01.10 (or a future version of BPJ2K01.10) as the definition of the STANAG 7023 JPEG 2000 preferred encoding. Until that happens, STANAG 7023 remains the normative reference.

H.1 JPEG 2000 Profile and Limitations

SPJE is a superset of the ISO/IEC 15444-1 Profile-1, NPJE, EPJE and TPJE. It is similar in scope to LPJE but makes more encoding recommendations. *It is possible to create SPJE compliant files that lie outside the boundaries defined within Profile-1 in JPEG 2000 Part 1 (see Annex C of BPJ2K01.00); therefore, other BPJ2K01.10 compliant systems might not be able to decode all possible SPJE files*. Any SPJE codestream, however, that conforms to the limitations of ISO/IEC 15444-1 Profile-1, will be correctly interpreted by all compliant BPJ2K01.10 decoders. Implementers are cautioned to exercise care when creating SPJE codestreams that do not fit within ISO/IEC 15444-1 Profile-1. Interoperability should always be a primary concern and SPJE codestreams should conform to Profile-1 of ISO/IEC 15444-1 wherever possible. For further desired conformance of SPJE encoders and decoders please see section 5.7.

All SPJE compliant encoders must correctly set the capability parameter, Rsiz, within the SIZ marker segment (see Table H-2). If an SPJE codestream meets the Profile-1 restrictions of ISO/IEC 15444-1, it is indicated through proper setting of the Rsiz parameter. This will allow another BPJ2K01.10 compliant decoder to decode that particular codestream. Codestreams that do not fit within Profile-1 of ISO/IEC 15444-1 are marked to indicate that decoding resources beyond Profile-1 are required to properly interpret those codestreams. This will alert BPJ2K01.10 compliant decoders that they may not be able to properly decode these codestreams.

All compliant SPJE decoders are able to properly decode compressed data within the limits of the SPJE preferred encoding and all BPJ2K01.10 compliant codestreams. All SPJE compliant encoders must produce compressed data that is within the limits of the SPJE encoding. It is recommended that encoders adhere to the recommendations in this

Appendix. There is *no requirement* that SPJE compliant encoders be able to produce other BPJ2K01.10 compliant codestreams (NPJE, EPJE or TPJE), although this functionality is desired and should be readily achieved with software-based encoders.

H.2 SPJE Overview

SPJE was developed to address the needs airborne reconnaissance sensors. It is broad in scope (similar to LPJE) and allows implementers great flexibility in JPEG 2000 encoding choices. SPJE meets the needs of both hardware and software implementations. Implementers are cautioned to carefully consider interoperability issues when designing around SPJE. The wide choice of encoding options places greater demands on decoders. Furthermore SPJE compliant systems may wish to exchange JPEG 2000 codestreams with NSIF01.01 compliant systems.

H.2.1 SPJE Encoding Choices

The following JPEG 2000 encoding choices are recommended for SPJE.

Progression Orders and Layers

- SPJE recommends the R-L-C-P progression order. The remaining progression orders L-R-C-P, R-P-C-L, P-C-R-L and C-P-R-L are allowed.

- No specific layering scheme is required. However, encoders should provide an adequate number of layers to provide quality scalability. NPJE recommends about 20 layers as a good starting point (see section D.4.7).

- No specific recommendation is made regarding rate control. Applications should consider the use of rate control procedures described in other BPJ2K01.10 profiles (see Appendix D).

Resolution Scalability

- SPJE does not require a specific number of decomposition levels. Encoders should use enough decomposition levels to enable reduced resolution roam and zoom. To be ISO/IEC 15444-1 Profile 1 compliant, enough decomposition levels must be included such that an LL subband smaller than 128 x 128 samples in size results.

Chipping and Parsing

- Tile-based compression is not recommended, but it is allowed for implementation constraints (e.g. hardware-based compression). Multiple-tile parts are also allowed but not recommended. Tiling (and chipping) is typically handled at the file level within STANAG 7023 (i.e. separate compressed files for each tile).

- PLT, PLM and TLM pointer marker segments are not required. They are allowed for use. Whenever tiles are used, the PLT and TLM marker segments should be used, but they are not required.

Coding Style Options

- Both the integer reversible (5-3R) and irreversible (9-7I) wavelet transforms are allowed.

- All arithmetic coding styles are allowed. Tile, component and tile-component specific encoding changes are allowed (i.e. main header and tile-part header QCC, QCD, COD and COC marker segments are allowed).

- Precinct usage, other than maximal precincts, is allowed.

- There are no reference grid restrictions. Images and tiles may be offset from the reference grid origin and reference grid sampling is allowed.

H.2.2 Other STANAG 7023 requirements

STANAG 7023 requires the inclusion of several other "data tables" when compressing imagery with JPEG 2000. These tables are specific to the STANAG 7023 file format and are discussed in section H.5.

H.2.3 JPEG 2000 Repackaging and Transcoding

STANAG 7023 and AEDP-9 do not make any specific recommendations regarding the repackaging and transcoding of JPEG 2000 compressed images. These concepts were discussed in section A.7. Since SPJE allows a wide range of JPEG 2000 encoding parameters similar to LPJE, implementers may wish to review the guidance given within Appendix G in addition to that found in Appendix A . Spatial chipping may occur at more than one level within STANAG 7023 when using JPEG 2000. Chipping may be accomplished at the codestream level if STANAG 7023 tiling is used. However, if tiles are used within the JPEG 2000 codestream, then spatial chipping may be accomplished through manipulation of the JPEG 2000 codestream. These procedures are discussed in Appendix A as well as other JPEG 2000 codestream manipulations that may be of interest to STANAG 7023 implementers. Implementers may also wish to review section 8 to see how these issues are dealt with within NSIF.

H.3 SPJE Codestream Structure

The SPJE JPEG 2000 main header contains markers and marker segments as shown in Figure H-1. Note that only the SIZ marker segment has a fixed placement, all other marker segments can appear in any order within the main header. The SIZ, COD and QCD marker segments are required in the main header by ISO/IEC 15444-1. SPJE prohibits the use of the POC marker segment. STANAG 7023 JPEG 2000 codestreams may not change their progression order. The remaining marker segments listed are allowed, but may not be recommended for use. See Table H-1 for a full listing of marker segments and where they may be used within SPJE.

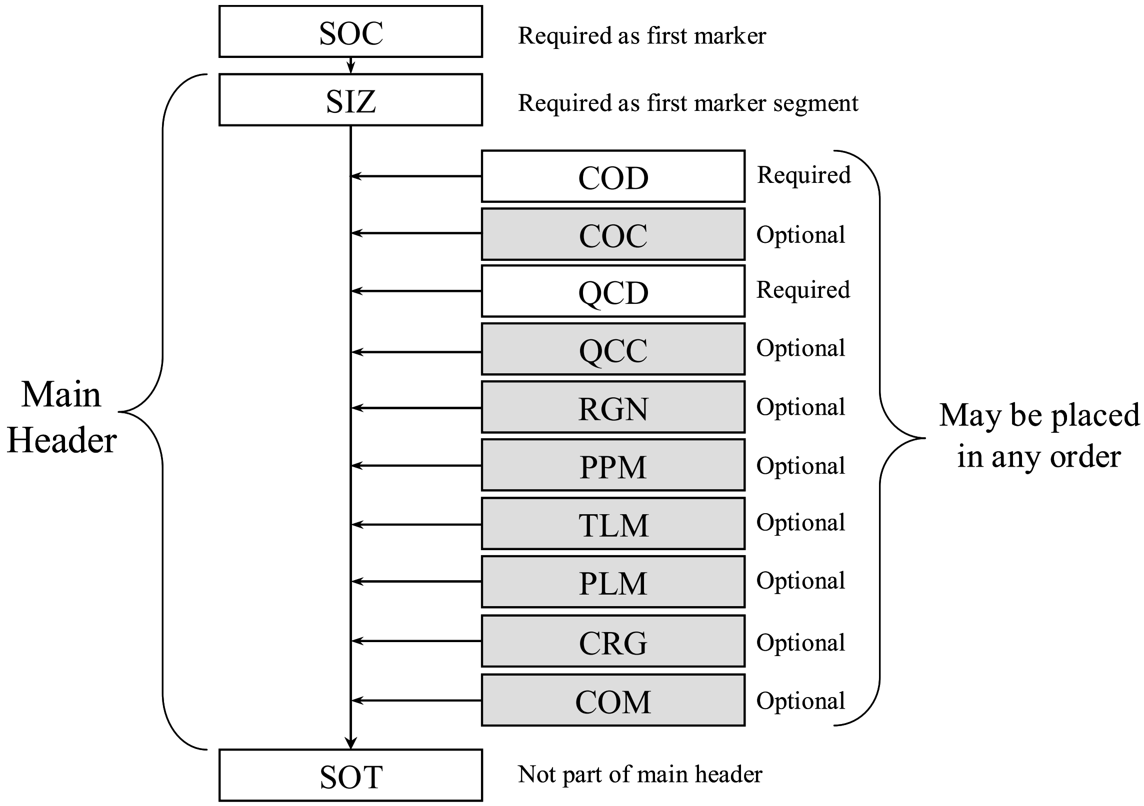

Figure H-1 Layout of SPJE JPEG 2000 main header

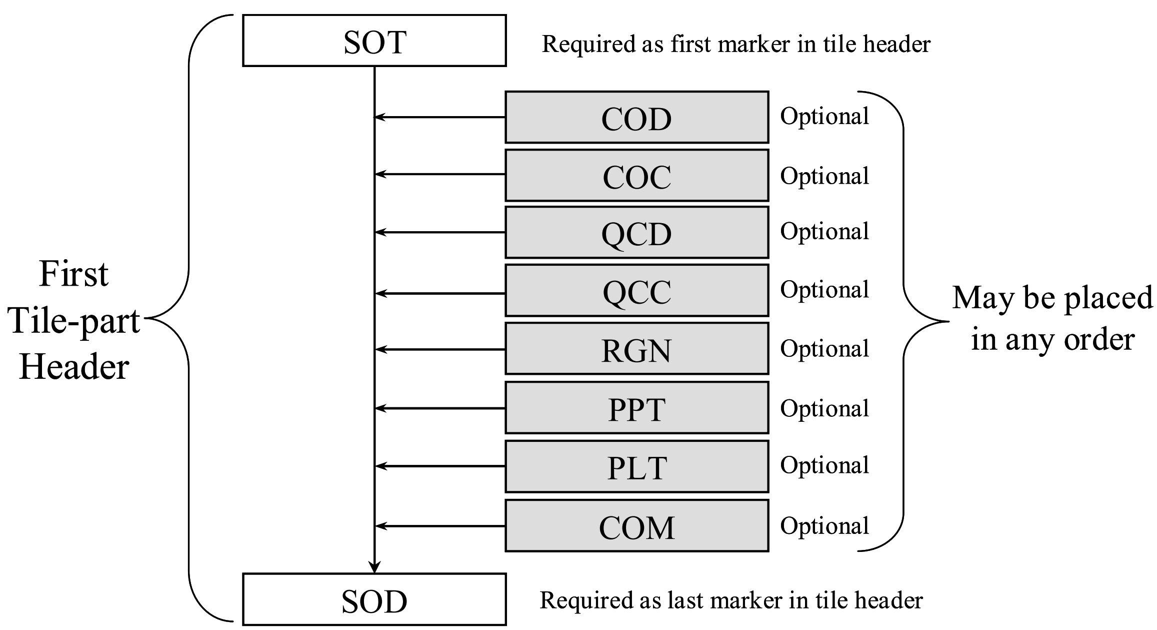

Figure H-2 Layout of a single JPEG 2000 tile-part header (first tile-part only)

Figure H-2 shows a JPEG 2000 tile-part header (first tile-part only) for SPJE. The structure of other tile-part headers (not first tile-part) is given in Figure H-3. Again, note that the POC marker segment is not allowed within SPJE. All other marker segments are optional. Refer to Table H-1 for a full listing of marker segments and where they may be used within SPJE as well as any additional applicable restrictions.

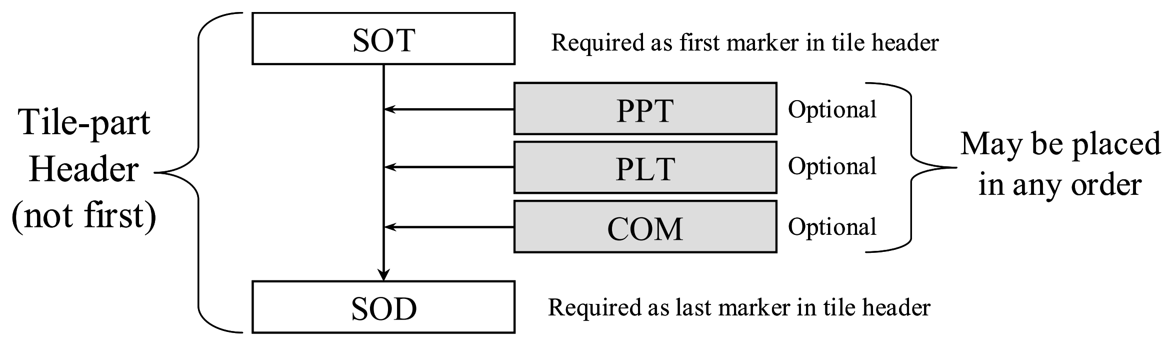

Figure H-3 Layout of a single JPEG 2000 tile-part header (not first tile-part)

H.4 Markers and Marker Segments Limits and SPJE

Table H-1 describes the recommended usage of each marker, whether it is required (Req.), not allowed (NA), optional (Opt.), recommended (Rec.) or not recommended (NR) and if there are any restrictions or dependencies. Marker segments may occur in three places within the codestream; the main header, a tile-part header, or the bitstream. A marker segment might be required in one area, the main header for example, and not allowed in others (tile-part header and bitstream). The bitstream plus the main and tile-part headers comprise the JPEG 2000 codestream. The bitstream comprises those portions of the codestream that are not in either the main or tile-part header. It is comprised mainly of entropy encoded data with some markers present.

If a marker segment is Required (Req.) it *shall* be present as indicated. There may be additional restrictions placed on the marker segment. For example, the SIZ marker segment is required in the main header and it must be the first marker segment present after the SOC marker. If a marker segment is Optional (Opt.) it *might* be present. The decision to use the marker segment is up to the implementation. The marker segment may also be subject to restrictions (e.g. COM marker segment). Marker segments that are not recommended (NR) are *not recommended* for general use. This *does not* mean that the marker segments are *forbidden*. Instead the marker segment might be used in special circumstances. Any marker segment marked Not Allowed (NA) *shall not* be used as indicated.

The encoding recommendations made in this Appendix should be compared against ISO/IEC 15444-1 Profile-1. Readers should compare Table 7-1 and Table H-1 to see the differences between the two encodings. The values column in the following tables give the full ISO/IEC 15444-1 ranges for comparison.

Table H-1 Marker and marker segment requirements within an SPJE codestream

Marker	Value	Main Header	Tile-part Header	Bitstream	Restrictions/Remarks
SOC	0xFF4F	Req.	NA	NA	Required as first marker in the file. Main headers starts immediately after SOC.
SOT	0xFF90	NA	Req.	NA	Required as first marker in each tile-part. Tile-part header occurs after SOT marker segment and before SOD marker.
SOD	0xFF93	NA	Req.	NA	Last marker in each tile-part header. Immediately precedes tile-part data.
EOC	0xFFD9	NA	NA	Req.	Required as the last marker in the codestream.
SIZ	0xFF51	Req.	NA	NA	Required as second marker segment in main header. Immediately follows SOC.
COD	0xFF52	Req.	Opt.	NA	One and only one required in main header. Optionally no more than one main appear in first tile-part header.
COC	0xFF53	Opt.	Opt.	NA	Optional. No more than one COC per component in main header and first tile-part header.
RGN	0xFF5E	Opt.	Opt.	NA	Optional. No more than one per component in main header or first tile-part header.
QCD	0xFF5C	Req.	Opt.	NA	One and only one required in main header. No more than one in first tile-part headers.
QCC	0xFF5D	Opt.	Opt.	NA	Optional. No more than one per component in main header or first tile-part header.
POC	0xFF5F	NA	NA	NA	Not Allowed. SPJE prohibits the used of progression order changes.
TLM	0xFF55	Rec.	NA	NA	Recommended unless encoding image in a single tile-part.
PLM	0xFF57	NR	NA	NA	Not recommended, but allowed. Main header only, PLT is preferred.
PLT	0xFF58	NA	Rec.	NA	Recommended. Multiple PLTs may appear per tile. Must appear in a tile-part header prior to the packets whose lengths are described.
PPM	0xFF60	NR	NA	NA	Not recommended, but allowed. Use of a PPM prohibits use of PPT and in-bitstream packet headers.

Table H-1 Marker and marker segment requirements within an SPJE codestream

Marker	Value	Main Header	Tile-part Header	Bitstream	Restrictions/Remarks
PPT	0xFF61	NA	NR	NA	Not recommended, but allowed. All packet headers for the tile in which PPT appears must be included in the PPT. Use of PPT prohibits use of PPM and in-bitstream packet headers for the tile.
SOP	0xFF91	NA	NA	Opt.	Optional, recommended for use in noisy communication environments. May be used in front of each packet. Whether or not a SOP marker is used for a given packet, the Nsop parameter is incremented. If packet headers are moved into a PPM or PPT marker segment, the SOP marker segment may appear immediately before packet bodies in the bitstream.
EPH	0xFF92	Opt.	Opt.	Opt.	Optional, recommended for use in noisy communication environments. If used, they must appear after each packet header. If packet headers are moved into PPM or PPT marker segments, the EPH markers are moved as well.
CRG	0xFF63	NR	NA	NA	Nor recommended, but allowed. Informational use only, implementations need not do anything with this information. Only one may appear in main header and it applies to all tiles.
COM	0xFF63	Opt.	Opt.	NA	Optional use. Informational only, no data necessary to decode the codestream or metadata needed to interpret the codestream shall be placed in a COM marker segment.

H.4.1 Delimiting Markers and Marker Segments

Each delimiting marker shall be present in all compliant JPEG 2000 codestreams. A codestream shall have only one SOC and EOC marker, and at least one tile-part. Each tile-part has one SOT and one SOD marker. The SOC, SOD and EOC markers are formatted the same as in ISO/IEC 15444-1 Profile-1 and the other BPJ2K01.10 encodings (see Table 7-2, Table 7-4 and Table 7-5). The SOT marker segment for SPJE has the same format and range of values as that found in Profile-1 (see Table 7-3). Although SPJE allows the use of JPEG 2000 tiles and even multiple tile-parts, tiling is not recommended within STANAG 7023 JPEG 2000 codestreams.

H.4.2 Fixed Information Marker Segment

The SIZ marker segment (see Table H-2) includes information required to properly decode the image. There shall be a SIZ marker segment in the main header immediately after the SOC marker segment.

Table H-2 Image and tile size (15444-1 Annex A.5.1)

Parameter	Size (bits)	Values	SPJE	Notes
SIZ	16	0xFF51	0xFF51	Image and tile size marker
Lsiz	16	41 – 49,190	41 – 49,190	Length of marker segment
Rsiz	16	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010 0000 0000 0000 0011 0000 0000 0000 0100	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010	Rsiz = 0000 0000 0000 0000 indicates that the full capabilities described by ISO/IEC 15444-1 are required. Rsiz = 0000 0000 0000 0001 indicates that the codestream is Profile 0 compliant. Rsiz = 0000 0000 0000 0010 indicates that the codestream is Profile-1 compliant. Rsiz = 0000 0000 0000 0011 and Rsiz = 0000 0000 0000 0100 see below*
Xsiz	32	$1 - (2^{32} - 1)$	$1 - (2^{32} - 1)$	Reference grid width
Ysiz	32	$1 - (2^{32} - 1)$	$1 - (2^{32} - 1)$	Reference grid height
XOsiz	32	$0 - (2^{32} - 2)$	$0 - (2^{32} - 2)$	Horizontal offset from the origin of the reference grid to the left side of the image area
YOsiz	32	$0 - (2^{32} - 2)$	$0 - (2^{32} - 2)$	Vertical offset from the origin of the reference grid to the top of the image area
XTsiz	32	$1 - (2^{32} - 1)$	$1 - (2^{32} - 1)$	Tile width on reference grid
YTsiz	32	$1 - (2^{32} - 1)$	$1 - (2^{32} - 1)$	Tile height on reference grid

Table H-2 Image and tile size (15444-1 Annex A.5.1)

Parameter	Size (bits)	Values	SPJE	Notes
XTOsiz	32	$0 – (2^{32} – 2)$	$0 – (2^{32} – 2)$	Horizontal offset from the origin of the reference grid to the left edge of the first tile
YTOsiz	32	$0 – (2^{32} – 2)$	$0 – (2^{32} – 2)$	Vertical offset from the origin of the reference grid to the top edge of the first tile
Csiz	16	1 – 16,384	1 – 16,384	The number of components in the image.
$Ssiz^i$	8	0000 0000 – 1010 0101	Unsigned: 0 – 37 Signed: 128 – 165	0xxx xxxx Unsigned data 1xxx xxxx signed data x000 0000 – x010 0101 data bit depth = value + 1
$XRsiz^i$	8	1 – 255	1 – 255	Horizontal sub sampling on the reference grid with respect to the i^{th} component
$YRsiz^i$	8	1 – 255	1 – 255	Vertical sub sampling on the reference grid with respect to the i^{th} component

* Two new values for the Rsiz parameter have been added in ISO/IEC 15444-1 Amendment 1, 2006. These values were assigned to the Digital Cinema Initiative to indicate which DCI profile (2K or 4K) the codestream adheres to. See the document "Digital Cinema System Specification**,** Version 1.1"**,** April 12, 2007 for further details. SPJE implementations *shall not* set Rsiz = 0000 0000 0000 0011 or Rsiz = 0000 0000 0000 0100.

The Digital Cinema Initiative has defined two new values for Rsiz. These values are given in Amendment 1 of JPEG 2000 Part 1. These values indicate that the JPEG 2000 codestream complies with the 2K (Rsiz = 3) and 4K (Rsiz = 4) DCI profiles. The DCI 2K profile does fall within scope of SPJE. It is possible, but unlikely, that an SPJE encoder creates a DCI 2K profile compliant codestream. In this case, the SPJE encoder shall not set Rsiz = 3, rather the codestream shall be marked as Rsiz = 0 (NR profile) or as Rsiz = 2 (Profile-1) compliant as appropriate.

H.4.3 Functional Marker Segments

The functional marker segments define what parameters were used in the compression of a given tile or an image. These marker segments apply to the entire tile when in the tile header and to the image when in the main header. Markers in a tile-part header supersede markers in the main header.

H.4.3.1 Coding Style Marker Segments (COD and COC)

Table 7-7 gives the COD marker segment. SPJE allows the full range of coding options to be used when generating codestreams. This includes all progression orders, code-block sizes, wavelet transforms and multiple component transforms. No limits are placed on precinct sizes or the number of layers. The COD marker segment is required in the main header of the codestream and it contains the default encoding parameters applied to all components and tiles unless it is overridden (see COC below).

SPJE encoders should include enough layers to enable quality scalability across all resolutions. Approximately 10 – 20 layers suffice for a monochrome image. Hardware implementations may wish to use less. The maximum number of layers should not exceed 50 unless special circumstances exist (i.e. a large number of components and/or decomposition levels).

Code block sizes in JPEG 2000 are limited to a maximum number of 4,096 coefficients within the code-block. SPJE allows the full range of codeblock sizes. This facilitates development of hardware-based compression systems that may seek to minimize memory requirements and software implementations that may seek to maximize compression performance. See section G.4.3.1 for discussion regarding choice of code-block size.

JPEG 2000 allows for several options regarding “code-block style” (see Table 7-10). SPJE allows the use of all code-block styles. These parameters are used to control the behavior of the arithmetic encoder. Annex D in ISO/IEC 15444-1 describes the meaning of these parameters. In general, these options relate to speeding up the arithmetic encoding (selective bypass) and improving error resiliency/detection (segmentation symbols, termination, resetting context probabilities, vertically causal). Some of the above options also reduce memory costs to a small extent. For more discussion on these options see Annex J in ISO/IEC 15444-1 as well.

The default COD coding parameters can be overridden. A COC marker segment (see Table 7-12) can be used to override the coding parameters for a single specified component. To override the coding parameters for more than one component, multiple COC marker segments must be used (to override coding parameters for all components in a tile, a tile-part COD marker segment can be used). If the COC marker appears in the main header, then the default coding parameters (as defined by the main header COD marker segment) for the specified component are replaced by those in the COC marker segment for the entire image. If the COC marker segment appears in a first tile-part header, then the coding parameters for the specified component are replaced for that tile only. We can express the relationships between main and tile-part COD and COC marker segments as follows:

Tile-part COC > Tile-part COD > Main Header COC > Main Header COD

This illustrates that tile-part COC marker segments supersede tile-part COD marker segments, which supersede main header COC marker segments, which supersede the main header COD marker segment.

H.4.3.2 Region of Interest Marker Segment

The region of interest marker segment (RGN) is shown in Table 7-14. The RGM marker segment is allowed within SPJE. This marker segment is used for region of interest coding. Encoders can shift the wavelet coefficients corresponding to a spatial region of interest up (by left bit-shift) above the most significant bitplane of the remaining background wavelet coefficients. This has the effect that these wavelet coefficients will be encoded first before all other wavelet coefficients. The RGN marker segment alerts the decoder that this has been done so that the process can be reversed during decoding.

H.4.3.3 Quantization Default Marker Segments (QCD and QCC)

The QCD marker segment (see Table 7-15) is required in the main header to indicate the quantization step-sizes (for the 9-7I wavelet) or reversible dynamic range (for the 5-3R wavelet) that is valid for all tile-parts. SPJE allows all quantization options including scalar derived quantization for the 9-7I wavelet transform. Two guard bits are also recommended for all applications, but applications can use more or less if it is deemed necessary.

The main header QCD marker segment can be overridden by a QCC marker segment (see Table 7-19) in the main header or by tile-part QCD and QCC marker segments. The rules for QCD/QCD precedence are similar to those for COD/COC precedence. The following relationships hold for main and tile-part header QCD and QCC marker segments:

Tile-part QCC > Tile-part QCD > Main Header QCC > Main Header QCD

SPJE allows the full range of parameter values for the QCC marker segment.

H.4.3.4 Progression Order Change Marker Segment (POC)

The POC marker segment (see Table 7-20) is used to change progression orders within a codestream. POC marker segments require a good understanding of JPEG 2000 codestream construction and a sophisticated JPEG 2000 codec. The POC marker segment allows full control over the ordering of codestream data within a file. For most applications, the POC marker segment is not necessary; one progression order will suffice for all codestream data. ISO/IEC 15444-1 Profile-1 compliant decoders are required to handle POC marker segments. Therefore all SPJE decoders *shall* properly decode codestreams containing POC marker segments.

SPJE encoders *shall not* use the POC marker segment. The POC marker segment is allowed within LPJE (Appendix G). The usage of the POC marker segment represents the major difference between LPJE and SPJE. This difference, however, does not pose an interoperability issue since all SPJE decoders must properly interpret the POC marker segment.

H.4.4 Pointer Marker Segments

The pointer markers segments are used to gain quick access to data for parsing, chipping, and decoding. These marker segments define either the lengths of certain codestream constructs or pointers to the start of codestream constructs.

H.4.4.1 Tile-part Lengths Marker Segment (TLM)

The tile-part lengths marker segment (see Table 7-21) has the same length information as the start of tile marker segments in each tile-part, but this information is collected up front in the main header. This marker segment can be used to quickly access and chip a given tile or set of tiles in a compressed image. SPJE makes no recommendation regarding the use of the TLM marker segment. The SPJE encoding recommends that codestreams contain a single-tile part (i.e. single tile, single tile-part) for the entire image. In this case the TLM marker segment is not needed since the same information is simply contained in the tile-part header. In situations where multiple tiles/tile-parts are used, the TLM marker segment is very useful and should be utilized.

H.4.4.2 Packet Lengths Marker Segment, Main Header and Packet Lengths Marker Segment, Tile-part Header (PLM and PLT)

The packet length marker segments (PLM and PLT, see Table 7-23 and Table 7-25 respectively) collect together the lengths of the packets so that a decoder may rapidly skip through them to find the packets it wants without having to decode all intermediate packet headers (see discussion in section G.4.4.2). SPJE allows the use of both marker segments, but the PLT marker segment is recommended. The PLT marker segment is preferred, because it is possible to overload the PLM marker segment (see discussion in ISO/IEC 15444-1:2004, Annex A.7.2). For this reason, SPJE does not recommend the use of the PLM marker segment.

H.4.4.3 Packed Packet Headers Marker Segment, Main Header and Packed Packet Headers Marker Segment, Tile-part Header (PPM and PPT)

There are two more marker segments that can be used to assist in the parsing of packet data. These are the PPM (packed packet headers, main header see Table 7-26) and the PPT (packed packet headers, tile-part header see Table 7-27). These marker segments aggregate the packet headers (not the packet bodies) into either the main header or tile-part headers. This allows decoders to bulk read the headers and process them instead of parsing them out of the codestream (see discussion in section G.4.4.3).

SPJE allows usage of the PPM and PPT marker segments but they are *not* recommended. In some circumstances, using the PPM and PPT marker segments is appropriate such as noisy channel communications in conjunction with an unequal bit error protection scheme. Preserving the layer contributions and packet body lengths would allow for some measure of error recovery. The PPT marker segment is preferred over the PPM marker segment since it does not front load the codestream as much. In cases where there is only one tile in the image, the PPM marker segment is as efficient as the PPT marker segment. If either marker segment is used, then the packet headers shall not appear in the bitstream data. Furthermore, if the PPM marker segment is used, the PPT marker segment shall not be used. The converse is true as well.

H.4.5 Start of Packet Marker Segment and End of Packet Header Marker Segment (SOP and EPH)

ISO/IEC 15444-1:2004 defines an in bitstream marker segment (SOP, see Table 7-28) and marker (EPH, see Table 7-29) that can be used to improve error resiliency when operating in noisy environments. Usage of these marker segments is signaled through the COD marker segment (see Table 7-7). SPJE encourages the use of the SOP and EPH markers in noisy channel communication situations.

H.4.6 Component Registration Marker Segment and Comment Marker Segment (CRG and COM)

The informational marker segments are not required for decoding but may assist in the interpretation of the data. The SPJE preferred encoding *does not* recommend their use and SPJE compliant systems are not required to generate or interpret these marker segments. Component registration (CRG, see Table 7-30) allows each component to be registered to each other for display. The Comment marker (COM, see Table 7-31) allows for the unstructured data to be included into the file. It is not recommended that either of these markers be used. *No data shall be placed in a comment marker segment that is necessary to decode or properly interpret a JPEG 2000 codestream.* Neither metadata that might be used to properly interpret and exploit the compressed imagery nor any information that might be used to improve JPEG 2000 decoder performance shall be placed within a COM marker segment.

H.5 Additional STANAG 7023 Format Requirements

The STANAG 7023 format requires the inclusion of other additional “data tables” that lie outside of the JPEG 2000 codestream and file formats whenever imagery is compressed with JPEG 2000. These data tables form parts of the STANAG 7023 file format. Note that the following description does not constitute a complete description of a valid STANAG 7023 file carrying a JPEG 2000 encoded image. There are many different required data tables that must be carried to form a valid STANAG 7023 file. We describe here instead those specific data tables that are required when compressing primary imagery using JPEG 2000. For a complete description of the STANAG 7023 format, the interested reader is referred to the STANAG 7023 document.

H.5.1 Sensor Compression Data Table

The sensor compression data table (see STANAG 7023, Annex A-10.1) is required whenever imagery is compressed using JPEG (ISO/IEC 10918-1) or JPEG 2000 (ISO/IEC 15444-1). Table H-3 gives the sensor compression data table. For SPJE the value of the compression algorithm field shall be set to 3.

Table H-3 Sensor compression data table (STANAG 7023 Annex A-10.1)

Field	Field Name	Req.	Number Bytes	Field Type	Encoding Scheme	Description/Encoding Units
1	Compression algorithm	Mand.	1	Encode	Unsigned Binary	Algorithm used to compress sensor data. $02 JPEG (ISO/IEC 10918-1:1994) $03 JPEG 2000 (ISO/IEC 15444-1)

H.5.2 JPEG 2000 Description Data Table

The JPEG 2000 description data table (see STANAG 7023, Annex A-10.6) collects together certain codestream attributes into one table. Although this information may be gleaned from the JPEG 2000 codestream, the JPEG 2000 description data table is used to collect the information into an easy accessed place. Table H-4 shows the JPEG 2000 description data table from STANAG 7023. Note that the sixth field in the data table indicates if tiling has been used in the JPEG 2000 codestream. Depending on the value of this field, the JPEG 2000 Index Data Table (see section H.5.3) shall or shall not be present in the STANAG 7023 file.

Table H-4 JPEG 2000 description data table (STANAG 7023 Annex A-10.6)

Field	Field Name	Req.	Number Bytes	Field Type	Encoding Scheme	Description/Encoding Units
1	Codestream capability	Mand.	1	Encode	Unsigned Binary	Describes the codestream compliance The possible value is (see ISO/IEC 15444-1): $01 Profile-0 $02 Profile-1
2	Progression order	Mand.	1	Encode	Unsigned Binary	The progression order type. Possible values are: $00 **L**ayer-**R**esolution-**C**omponent-**P**osition $01 **R**esolution-**L**ayer-**C**omponent-**P**osition $02 **R**esolution-**P**osition-**C**omponent-**L**ayer $03 **P**osition-**C**omponent-**R**esolution-**L**ayer $04 **C**omponent-**P**osition-**R**esolution-**L**ayer
3	Number of decomposition levels	Mand.	1	Immed	Unsigned Binary	The number of wavelet transformations applied. N decomposition levels lead to N+1 resolution levels (from the coarsest to the finest resolution). Range of values: 0-32.
4	Number of layers	Mand.	2	Immed	Unsigned Binary	The number of layers (quality increments). Range of values: 0-65535.
5	Number of components	Mand.	2	Immed	Unsigned Binary	The number of components (bands) of the sensor data. Range of values: 1-16384.
6	JPEG 2000 tiling performed*	Mand.	1	Encode	Unsigned Binary	Possible values are: $00 No JPEG 2000 Tiling has been used $01 JPEG 2000 Tiling has been used

Table H-4 JPEG 2000 description data table (STANAG 7023 Annex A-10.6)

Field	Field Name	Req.	Number Bytes	Field Type	Encoding Scheme	Description/Encoding Units
7	IREP	Opt.	1	Encode	Unsigned Binary	This field shall contain a valid indicator of the processing required in order to display an image. Valid representation indicators are MONO for monochrome, RGB for red, green, or blue true color, RGB/LUT for mapped color, MULTI for multiband imagery, NODISPLY for an image not intended for display, NVECTOR and POLAR for vectors with Cartesian and polar coordinates respectively, and VPH for SAR video phase history. In addition, compressed imagery can have this field set to YCbCr601 when compressed in the ITU-R Recommendation BT.601-5 color space using JPEG. Valid codes: $00 = BCS-A $01 = MONO $02 = RGB $03 = RGB/LUT $04 = MULTI $05 = NODISPLY $06 = NVECTOR $07 = POLAR $09 = VPH $0A = YCbCr601 (See also STANAG 4545 Table C-1-3)

* Note: Field 6 describes if JPEG 2000 Tiling has been performed.

• If No JPEG 2000 Tiling has been used, the JPEG 2000 Index Data Table shall be used (value 0).

• If JPEG 2000 Tiling has been used, the JPEG 2000 Index Data Table shall not be used (value 2).

H.5.3 JPEG 2000 Index Data Table

The JPEG 2000 index data table is conditionally present based on the JPEG 2000 description data table. If no JPEG 2000 tiling has been used in the codestream (indicated by field six in the JPEG 2000 description data table, see Table H-4), the JPEG 2000 index data table shall be present. This table lists the byte offsets (from the start of codestream) to the end of successive progression order points within the codestream. Pointers are only generated for the outermost loop in the progression order.

For example, if the progression order is R-L-C-P with three levels of wavelet decomposition, there will be four offsets. The first offset will point to the last byte in the codestream that completes resolution level zero. The second offset will point to the last byte in the codestream that completes resolution level one, and so on. All offsets are measured from the start of the codestream, they are not cumulative offsets. If the progression order were different, say L-R-C-P, then the offset would indicate where each layer ends within the codestream.

Table H-5 JPEG 2000 index data table (STANAG 7023 Annex A-10.7)

Field	Field Name	Req.	Number Bytes	Field Type	Encoding Scheme	Description/Encoding Units
1	Highest order of progression index	Mand.	4	Immed	Unsigned Binary	Byte count offset from start of codestream to the end of the first complete step of progression of the respective progression order.
2	(*)					Repeat for each successive step of progression.

Table H-5 shows the JPEG 2000 index data table. The table length will be a function of the number of progression levels within the outermost loop of the progression order. STANAG 7023 gives several examples in Annex A-10.7 of how to fill out Table H-5 for different progression orders. Readers are encouraged to review these examples to better understand this data table.