INTERNATIONAL	**ISO/IEC**
REGISTERED GRAPHICAL ITEM	**BIIF PROFILE**
CLASS: BIIF PROFILE	**BPJ2K01.00**

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

BIIF Profile for JPEG 2000
Version 01.00
(BPJ2K01.00)

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

30 July 2004

This page intentionally left blank.

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

List of Figures

List of Tables

Table of Contents (continued)

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 Organisation (NATO) Secondary Imagery Format Version 01.00 (NSIF01.00). The NSIF01.00 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 Agreement (STANAG) 4545, 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 01.00

This BIIF Profile for JPEG 2000 is a profile developed using the JPEG 2000 performa, ISO/IEC 15444-1, intended to be used with BIIF applications.

This page intentionally left blank.

Introduction

The definition of the BIIF Profile for JPEG 2000 Version 01.00 (BPJ2K01.00) is within the context of the BIIF Profile class of graphical items in accordance with the principles and procedures specified in ISO/IEC 9973, "Computer graphics and image processing -- Procedures for registration of graphical items," and Annex C of ISO/IEC 12087-5:1998, "Profiling BIIF."

The BPJ2K01.00 Profile of JPEG 2000 Part 1 (ISO/IEC 15444-1) was cooperatively developed between the ISO and NATO communities.

The NSIF01.00 is a BIIF profile intended to promote interoperability for the exchange of Imagery among military Command, Control, Communications, and Intelligence (C3I) systems. The BPJ2K01.00 is the profile for the JPEG 2000 compression of digital imagery, incorporating the compressed digital imagery into NSIF files, and exchanging them within the user community. STANAG 4545 and (U.S.) MIL-STD-2500B are specific user community documents used for implementing the NSIF01.00 BIIF Profile and now, the BPJ2K Profile.

This document includes the following sections:

1) The profile limits of JPEG 2000 for use within the NSIF (Section 7)
2) A recommended encoding for use within the NSIF (Section 8)
3) The interaction between the NSIF/BIIF file format and the JPEG 2000 file format (Section 9)
4) Several informative Appendices.

All compliant NSIF decoders are required to decode all compliant data within the limits of this profile and their BIIF compliance level (Section 7). All compliant NSIF 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 recommendations in Section 8 of this document. The preferred encoding recommendations, NSIF Preferred JPEG 2000 Encoding (NPJE), 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.

While Section 8 defines the recommended encoding, NSIF will support all encoders and encoded data that are within the limits of Section 7. Section 9 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: JPEG 2000 Processing, 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 1. Appendix D introduces the Exploitation Preferred JPEG 2000 Encoding scheme.

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 the multiple component compression in JPEG 2000 Part 2 will be included in the next version of this profile. Added functionality and new recommendations will only be added to the profile as 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 BPJ2K01.00 profile defines allowed data values and ranges for JPEG 2000 header and subheader fields contained in an NSIF01.00 or NITFS02.1 file. A BPJ2K01.00 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 BPJ2K01.00 profile meets BIIF ISO/IEC 12087-5:1998 application requirements.

1.2 Position Within the Graphical Item Register

BPJ2K01.00 is a profile for the application of ISO/IEC 15444-1, JPEG 2000 Part 1, registered under the BIIF Profile class of graphical items in accordance with ISO/IEC 9973. The BPJ2K01.00 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 graphical item registration information is as follows:

Graphical Item Class:	BIIF Profile
Graphical Item Long Name:	BIIF Profile for JPEG 2000 Version 01.00
Graphical Item Short Name:	BPJ2K01.00
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; U.S. Secretary of the Air Force, Information Dominance Directorate, Reconnaissance Systems Division (SAF/AQIJ).

1.3 User Requirements and Scenario

NSIF01.00 is designed to promote interoperability for the exchange of digital electronic imagery among multi-national Command, Control, Communications, and Intelligence (C^3I) Systems, and those systems needing to interoperate with C^3I 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.

1) The profile is comprehensive in the type and format of data permitted in the BIIF File.

2) The profile may be implemented across a wide range of computer systems without reduction of available features.

3) The profile allows extensibility to accommodate data types and functional requirements not foreseen.

4) 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 BPJ2K01.00. 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 01.00

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
ISO/IEC 15444-1:2000	JPEG 2000 Image Coding System -- Part 1: Image Coding System: Core Coding System.
ISO/IEC 15444-1:2002	JPEG 2000 Image Coding System -- Part 1: Image Coding System: Core Coding System.
ISO/IEC 15444-1-AMD1	JPEG 2000 Image Coding System -- Part 1: Image Coding System: Core Coding System, Amendment 1.
ISO/IEC 15444-1-AMD2	JPEG 2000 Image Coding System -- Part 1: Core Coding System, Amendment 2.
ISO/IEC 15444-2:2002	JPEG 2000 Image Coding System -- Part 2: Extensions.
ISO/IEC 15444-4:2002	JPEG 2000 Image Coding System -- Part 4: Conformance testing

Non-Normative References:

The following documents are included for information 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
STANAG 4545	NATO Secondary Imagery Format (NSIF)
ISO/IEC 9973	Computer Graphics And Image Processing -- Procedures For Registration Of Graphical Items
MIL-STD-2500B	National Imagery Transmission Format Version 2.1 for the National Imagery Transmission Format Standard

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 BPJ2K01.00 profile, the definitions shown in ISO/IEC 15444-1, ISO/IEC 12087-5 BIIF, and NSIF01.00 apply.

4 Abbreviations

ABPP	Actual Bits Per Pixel
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	Command, Control, Communications, and Intelligence
CIE	Commission Internationale de'lEclairage (International Commission on Illumination)
CCS	Common Coordinate System
COC	Coding style Component
COD	Coding style Default
COM	Comment Marker
COMRAT	Compression Rate
CONOPS	Concept of Operations
CRG	Component Registration
DCT	Discrete Cosine Transform
DRA	Dynamic Range Adjustment
EOC	End of Codestream
EPH	End of Packet Header
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 minimal interchange format
JPC	JPEG 2000 codestream
JPEG	Joint Photographic Experts Group

JPX	JPEG 2000 XML based file format
JTC1	Joint Technical Committee 1
L	Low pass filter
L-R-C-P	Layer-Resolution-Component-Position
LSB	Least Significant Bit
LUT	Look-up Table
MAS	Military Agency for Standardization
MIL-STD	Military Standard
MJ2	Motion JPEG 2000 Format
M/P	Magnitude/Phase Data
MSB	Most Significant Bit
MS	Multispectral
MTFC	Modulation Transfer Function Compensation
NA	Not Allowed
NATO	North Atlantic Treaty Organisation
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
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
PLM	Packet Length, in Main header
PLT	Packet Length, in Tile-part header
POC	Progression Order Change
PPM	Packed Packet headers, in Main header
PPT	Packed Packet headers, in Tile-part header
PVTYPE	Pixel Value Type
QCC	Quantization Component
QCD	Quantization Default
RCT	Reversible Component Transform
RGB	Red, Green, Blue (IREP value)
RGN	Region of interest
RRDS	Reduced Resolution Data Sets
SC29/WG1	Sub Committee 29/Working Group 1
SIZ	Image and Tile Size (marker)
SOC	Start of Codestream
SOD	Start of Data

SOP	Start of Packet
SOT	Start of Tile-part
STANAG	Standardization Agreement
TLM	Tile-part lengths Markers
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. Any JPEG 2000 decoder must meet certain requirements to be considered JPEG 2000 compliant. ISO/IEC 15444-4 describes the standard minimum requirements and includes test JPEG 2000 codestreams. Products that conform to the BPJ2K01.00 profile will also meet the conformance requirements of ISO/IEC 12087-5.

NSIF compliant BPJ2K01.00 decoders are able to fully decode NPJE compressed data that is within the limits of this profile for the NSIF Complexity Levels (CLEVELs) supported by the implementation. NSIF compliant decoders are also expected to properly decode any JPEG 2000 Compliant codestream within the conditions specified in ISO/IEC 15444-4 with respect to the NSIF Complexity Levels. NSIF compliant BPJ2K01.00 encoders produce compressed data that is within the limits of this profile and as constrained by NSIF CLEVEL constraints.

NSIF compliant BPJ2K01.00 encoders that support the preferred encoding recommendations of Section 8 have a mode of operation where compressed data is produced within the constrained limits detailed in Section 8.

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 BPJ2K01.00. The basis of this section is the limits that are associated with ISO/IEC 15444-1, Profile 1. All compliant BPJ2K01.00 decoders will be able to properly decode compressed data that is within the limits of this profile. All compliant encoders must produce compressed data that is within the limits of this profile. It is recommended that encoders adhere to the preferred encoding recommendations in the next section (Section 8).

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 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.
EOC	0xFFD9	NA	NA	Req.	Required as last marker in the code stream.
SIZ	0xFF51	Req.	NA	NA	Required as second marker (first marker segment) in the main header.
COD	0xFF52	Req.	Opt.	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	Opt.	Opt.	NA	No more than one COC per 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	Opt.	Opt.	NA	This is required if there are progression order changes different from main header. At most one may appear in any header. May appear in the first tile-part header of a given tile.
TLM	0xFF55	Opt.	NA	NA	Optional, there may be multiple TLM marker segments in the main header.
PLM	0xFF57	Opt.	NA	NA	Optional, there may be multiple PLM marker segments in the main header.
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.

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

Marker	Value	Main header	Tile Header	bit-stream	Restriction/Dependencies
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 a tile-part header before the packets whose headers are contained in the PPT 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 segment is used for a given packet, Nsop must be incremented for each packet in the bitstream. 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	Opt.	Opt.	Opt.	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	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.
COM	0xFF64	Opt.	Opt.	NA	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.

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 in bytes 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.
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 this marker segment in bytes (not including the marker).

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

Parameter	Size (bits)	Values	Notes
Rsiz	16	0000 0000 0000 0000 (no profile defined) 0000 0000 0000 0001 (Profile 0 compliant) 0000 0000 0000 0010 (Profile 1 Compliant)	This profile will be compliant with 0000 0000 0000 0010 = ISO profile 1.
Xsiz	32	1 — (2^{31}-1)	Width of reference grid. This profile is limited to Xsiz < 2^{31}
Ysiz	32	1 — (2^{31}-1)	Height of reference grid. 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)	Height 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
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
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 separation of a sample of the ith component with respect to the reference grid.
YRsiz^i	8	1 - 255	Vertical separation of a sample of the ith component with respect to the reference grid.

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 header and to the image when in the main header. Markers in the tile header supersede markers in the main header, but only apply to the given tile.

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	12 – 45	Length of this marker segment in bytes (not including the marker).
Scod	8	0000 0000 – 0000 0111 (See Table 7-8)	Coding style
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.
Multiple component transform	8	0000 0000 = No component transform used. 0000 0001 = Component transform used	Multiple component transformation.
SPcod	Variable	Defined below	
Number of decomposition levels (N_{Levels})	8	0 – 32	Number of wavelet decomposition levels.
Code-block width	8	0000 0000 – 0000 0100	Code-block width exponent offset value, xcb = value+2. For this profile this is limited to xcb $\leq$ 6
Code-block height	8	0000 0000 – 0000 0100	Code-block height exponent offset value, ycb = value+2. This profile is limited to ycb $\leq$ 6
Code-block style	8	0000 0000 – 0011 1111 (See Table 7-10)	Arithmetic coding parameters.
Transformation	8	0000 0000 – 0000 0001	Wavelet filter 0 = 9-7 irreversible filter 1 = 5-3 reversible filter
Precinct size	Variable	0000 0000 – 1111 1111 (See Table 7-11)	Precinct size (only if defined, Scod = xxxx xxx1).

Table 7-8. Coding style parameter values for Scod parameter

Values (bits) MSB LSB	Coding style
0000 0xx0	Entropy coder, precincts with PPx = 15 and PPy = 15
0000 0xx1	Entropy coder with precincts defined below
0000 0x0x	No SOP marker segments used
0000 0x1x	SOP marker segments may be used
0000 00xx	No EPH marker used
0000 01xx	EPH marker shall be used
	All other values reserved

Table 7-9. Progression orders for SGcod, SPcoc, and Ppoc parameters

Values (bits) MSB LSB	Progression orders
0000 0000	Layer – resolution level – component – position progression
0000 0001	Resolution level- layer – component – position progression
0000 0010	Resolution level – position – component – layer progression
0000 0011	Position – component – resolution level – layer progression
0000 0100	Component – position – resolution level – layer progression
	All other values reserved

Table 7-10. Code-block style for the SPcod and SPcoc parameters

Values (bits) MSB LSB	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 of 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 vertical casual context Vertical casual context
00x0 xxxx 00x1 xxxx	No predictable termination Predictable termination
000x xxxx 001x xxxx	No segmentation symbols are used Segmentation symbols are used
	All other values reserved

Table 7-11. Precinct width and height for the SPcod and SPcoc parameters

Values (bits) MSB LSB	Precinct size
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.

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

Parameter	Size (bits)	Values	Notes
COC	16	0xFF53	Coding style for a component if it is different than the default (COD)
Lcoc	16	9 - 43	Length
Ccoc	8 16	0 – 255; if Csiz < 257 0 – 16 383; if Csiz ≥ 257	Component index to which this marker segment applies.
Scoc	8	0000 0000 – 0000 0001 (defined Table 7-13)	
SPcoc[i]	Variable	Defined below	
Number of decomposition levels (N_{Levels})	8	0 – 32	Number of wavelet decomposition levels.
Code-block width	8	0000 0000 – 0000 0100	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 0100	Code-block height exponent offset value, ycb = value+2. For this profile is limited to ycb ≤ 6
Code-block style	8	0000 0000 – 0011 1111 (Defined in Table 7-10)	Arithmetic coding parameters.
Transformation	8	0000 0000 – 0000 0001	Wavelet filter 0 = 9-7 irreversible filter 1 = 5-3 reversible filter
Precinct size	Variable	0000 0000 – 1111 1111 (Defined in Table 7-11)	Precinct size (only if defined, Scoc = xxxx xxx1).

Table 7-13. Coding style parameter values for the Scoc parameter

Values (bits) MSB LSB	Coding style
0000 0000	Entropy coder with maximum precinct values PPx = PPy = 15
0000 0001	Entropy coder with precinct values defined in SPcoc[i]
	All other values reserved

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; if Csiz < 257 0 – 16 383; Csiz $\geq$ 257	Component index to which this marker segment applies
Srgn	8	0 = Implicit ROI (maximum shift)	All other values reserved.
SPrgn	8	0 - 37	Binary shifting of ROI coefficients above the background.

The QCD marker is used in the main header to indicate quantization step-sizes that are 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 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	4 – 197	Length of this marker segment in bytes (not including the marker).
Sqcd	8	Table 7-16	Quantization style
SPqcd[i]	Variable	Table 7-16	Quantization step size value or reversible dynamic range

Table 7-16. Quantization default values for the Sqcd and Sqcc parameters

Values (bits) MSB LSB	Quantization style	SPqcd or SPqcc size (bits)	SPqcd or SPqcc usage
xxx0 0000	No quantization	8	Table 7-17
xxx0 0001	Scalar derived (values signaled for N_LLL subband only)	16	Table 7-18
xxx0 0010	Scalar expounded (Values signaled for each subband). There are as many step sizes signaled as there are subbands.	16	Table 7-18
000x xxxx - 111x xxxx	Number of guard bits 0 – 7		
	All other values reserved		

Table 7-17. Reversible step size values for the SPqcd and SPqcc parameters (reversible transform only)

Values (bits) MSB LSB	Reversible step size values
0000 0xxx - 1111 1xxx	Exponent, ε_b, of the reversible dynamic range signaled for each subband.
	All other values reserved

Table 7-18. Quantization values for the SPqcd and SPqcc parameters (irreversible transformation only)

Values (bits) MSB LSB	Quantization step size values
xxxx x000 0000 0000 – xxxx x111 1111 1111	Mantissa, μ_b of the quantization step size value.
0000 0xxx xxxx xxxx – 1111 1xxx xxxx xxxx	Exponent, ε_b, of the quantization step size value.

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

Parameter	Size (bits)	Values	Notes
QCC	16	0xFF5D	Quantization component parameters
Lqcc	16	5 – 199	
Cqcc	8 16	0 – 255; if Csiz < 257 0 – 16 383; Csiz $\geq$ 257	Component index to which this marker segment applies
Sqcc	8	Table 7-16	Quantization style
$SPqcc^i$	Variable	Table 7-16	Quantization step size value or reversible dynamic range

Table 7-20. Progression order changes (15444-1 Annex A.7.1)

Parameter	Size (bits)	Values	Notes
POC	16	0xFF5F	Signals progression order changes.
Lpoc	16	9 – 65 535	
$RSpoc^i$	8	0 – 32	Resolution level (inclusive) for the start of a progression
CSpoc	8 16	0 – 255; if Csiz < 257 0 – 16 383; Csiz $\geq$ 257	Component index (inclusive) for the start of a progression
LYEpoc	16	1 – 65 535	Layer index (exclusive) for the end of a progression
REpoc	8	($RSpoc^i$ + 1) - 33	Resolution level (exclusive) for the end of a progression
CEpoc	8 16	($CSpoc^i$+1) – 255, 0; if Csiz < 257 ($CSpoc^i$+1) - 16 384, 0; if Csiz is $\geq$ 257 (0 is interpreted as 256)	Component index (exclusive) for the end of a progression
Ppoc	8	0000 0000 – 0000 0100 (defined in Table 7-9)	Progression order, one value for each progression change. The number of progression changes may be deduced from the marker segment length.

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 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.

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	6 – 65 535	Length of this marker segment in bytes (not including the marker).
Ztlm	8	0 – 255	
Stlm	8	0000 0000 – 0110 0000 (defined in Table 7-22)	These bits set ST and SP per 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)	Length in bytes from the beginning of the SOT marker for the i^{th} tile-part to end of bitstream data for that tile-part.

Table 7-22. Size parameters for Stlm

Values (bits) MSB LSB	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
00xx 0000	SP = 0; Ptlm parameter 16 bits
01xx 0000	SP = 1; Ptlm parameter 32 bits
	All other values reserved

Table 7-23. Packet length, main header (15444-1 Annex A.7.2)

Parameter	Size (bits)	Values	Notes
PLM	16	0xFF57	Packet length marker defined in the main header.
Lplm	16	4 – 65 535	Length of marker segment in bytes (not including the marker).
Zplm	8	0 – 255	Index of this marker segment relative to all other PLM marker segments in the main header.
$Nplm^{i}$	8	0 - 255	Number of bytes of packet header length information for the i^{th} tile-part in order as found in the codestream
$Iplm^{ij}$	Variable (succession of 8 bit units)	0000 0000 – 1111 1111	0xxx xxxx – Flag the termination of the $Iplm^{ij}$ which includes the next 7 bits. 1xxx xxxx – Continue reading next 8 bits until termination. x000 0000 – x111 1111 – 7 bits of packet length

Table 7-24. PLT Parameters Content (15444-1 Annex A.7.3)

Parameter	Size (bits)	Values	Notes
PLT	16	0xFF58	Packet length, tile-part header, marker.
Lplt	16	4 — 65535	Length of this marker segment in bytes (not including the marker).
Zplt	8	0 — 255	Index of this marker segment relative to all other PLT marker segments in the current header.
$Iplt^{i}$	Variable (succession of 8 bit units)	0000 0000 – 1111 1111	0xxx xxxx – Flag the termination of the $Iplt^{ij}$ which includes the next 7 bits. 1xxx xxxx – Continue reading next 8 bits until termination. x000 0000 – x111 1111 – 7 bits of packet length

Table 7-25. Packed packet headers, main header (15444-1 Annex A.7.4)

Parameter	Size (bits)	Values	Notes
PPM	16	0xFF60	Packed packet headers, in main header
Lppm	16	7 – 65 535	Length of this marker segment in bytes (not including the marker).
Zppm	8	0 – 255	Index of this marker segment relative to all other PPM marker segments in the main header.
$Nppm^i$	32	$0 – (2^{32} – 1)$	Number of bytes of packet header information for the i^{th} tile-part in the order found in the codestream
$Ippm^{ij}$	Variable	Packet headers	

Table 7-26. Packed packet headers, tile-part header (15444-1 Annex A.7.5)

Parameter	Size (bits)	Values	Notes
PPT	16	0xFF61	Packed packet headers, in tile-part header
Lppt	16	4 – 65 535	Length of this marker segment in bytes (not including the marker).
Zppt	8	0 – 255	Index of this marker segment relative to all other PPT marker segments in the current header.
$Ippt^i$	Variable	Packet headers	

7.6 In Bit Stream Marker and Marker Segments

These markers and marker segments are used to support error resilience. Both the start of packet (SOP) and end of packet header (EPH) are used to isolate individual packets and packet headers from each other in an environment where bit errors are likely.

Table 7-27. 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 this marker segment in bytes (not including the marker).
Nsop	16	0 – 65 535	Packet sequence number

Table 7-28. End of Packet Header (15444-1 Annex A.8.2)

Parameter	Size (bits)	Values	Notes
EPH	16	0xFF92	End of Packet Header marker

7.7 Informational Marker Segments

The informative 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 Comment marker (COM) allows for unstructured data to be included into the file.

Table 7-29. Component registration (15444-1 Annex A.9.1)

Parameter	Size (bits)	Values	Notes
CRG	16	0xFF63	
Lcrg	16	6 – 65 534	
$Xcrg^i$	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 7-30. Comment (15444-1 Annex A.9.2)

Parameter	Size (bits)	Values	Notes
COM	16	0xFF64	
Lcom	16	5 – 65 535	
Rcom	16	0 = General binary 1 = General Latin (IS 8859-15:1999)	All other values reserved
$Ccom^i$	8	0 - 255	

7.8 Low-Low Sub-band Restrictions

JPEG 2000 Part 1 Profile 1 has a requirement for the LL sub-band resolution. The restriction is as follows: If one tile is used for whole image, $(Xsiz – XOsiz)/D(I) \leq 128$ and $(Ysiz – YOsiz)/D(I) \leq 128$, 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 Preferred JPEG 2000 Encoding (NPJE)

A BPJ2K01.00 system is expected to be able to correctly decode ***any*** *JPEG 2000 Part 1 Profile 1* codestream contained in an NSIF or NITF 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 the compression and reformatting/repackaging methodology recommended for use within the NSIF standards.

8.1 NPJE Overview

The following JPEG 2000 parameter choices are recommended for original image providers in NSIF when performing the initial first-stage compression of collected imagery.

To enable quality scalability:

- 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 bit rate. The overall bit rate 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.

To enable resolution scalability:

- 5 levels of wavelet decomposition are performed to ensure that 6 resolution levels (R0 – R5) can be accessed from the codestream.

To facilitate ease of chipping and parsing:

- Images are tiled with JPEG 2000 at a tile size of 1024x1024.
- 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.

To establish uniformity in the JPEG 2000 codestreams produced:

- JPEG 2000 code block size is 64x64.
- 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.

8.1.1 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 a NPJE codestream will maintain most of the ‘compression’ recommendations presented in this section. However, several of these compression ‘recommendations’ may be altered during repackaging. In particular, repackaging may eliminate higher quality layers and/or eliminate higher resolution data sets. Furthermore, chipping is allowed, so the image offsets may 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 8-5 though Table 8-17 have a wider NPJE range than the values shown for 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.

8.2 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 8-1. 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 8-4 illustrates the ordering of compressed data within a single tile.

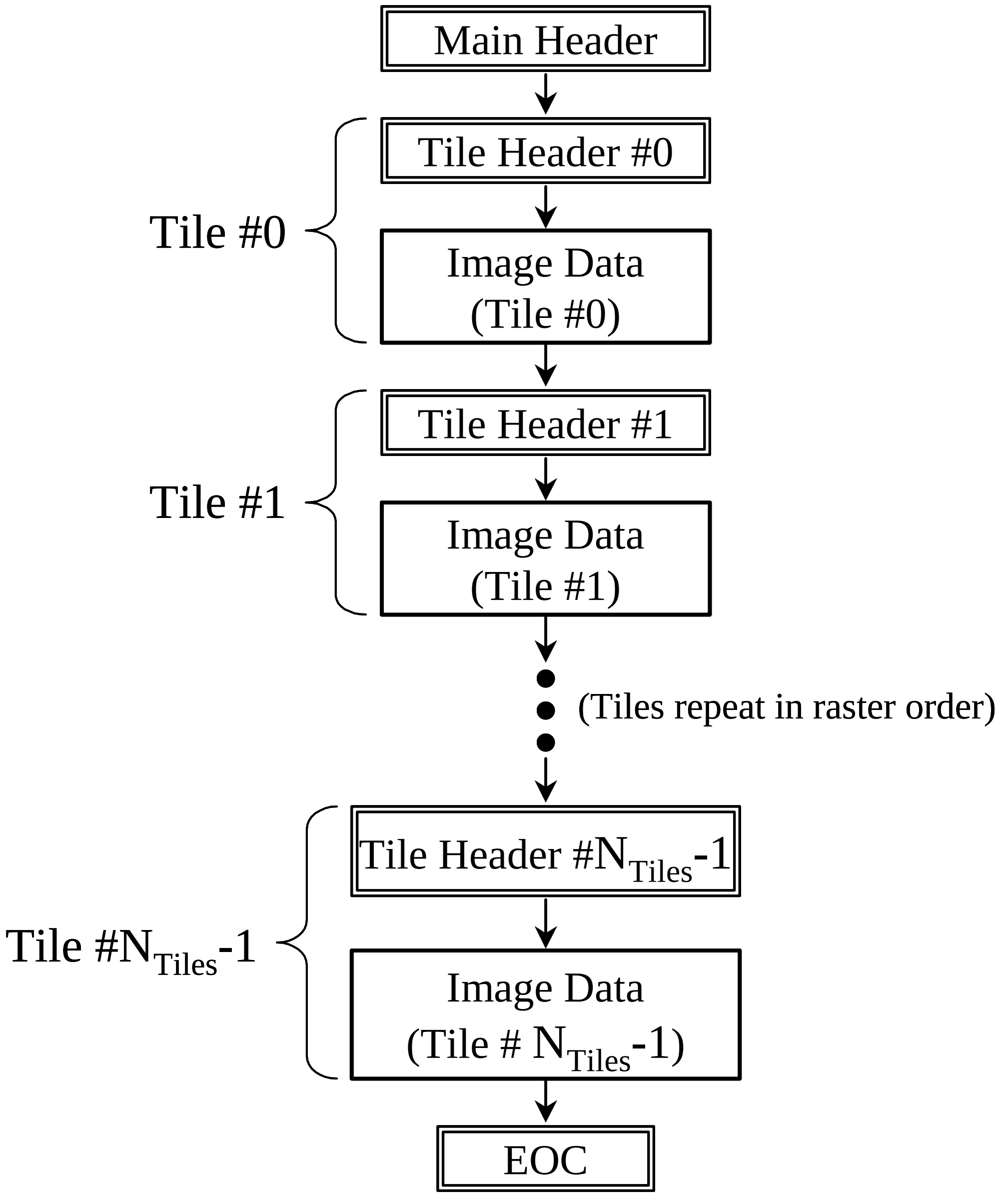

Figure 8-1. High-Level Layout of Entire JPEG 2000 Compressed File

The JPEG 2000 main header should contain markers and marker segments as shown in Figure 8-2:

The QCC marker segment is an optional marker segment. It may be used in special cases where the QCD of the main 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 COC marker segment is not recommended for this profile. It is anticipated that the COC marker segment will be made optional in a future version of this profile.

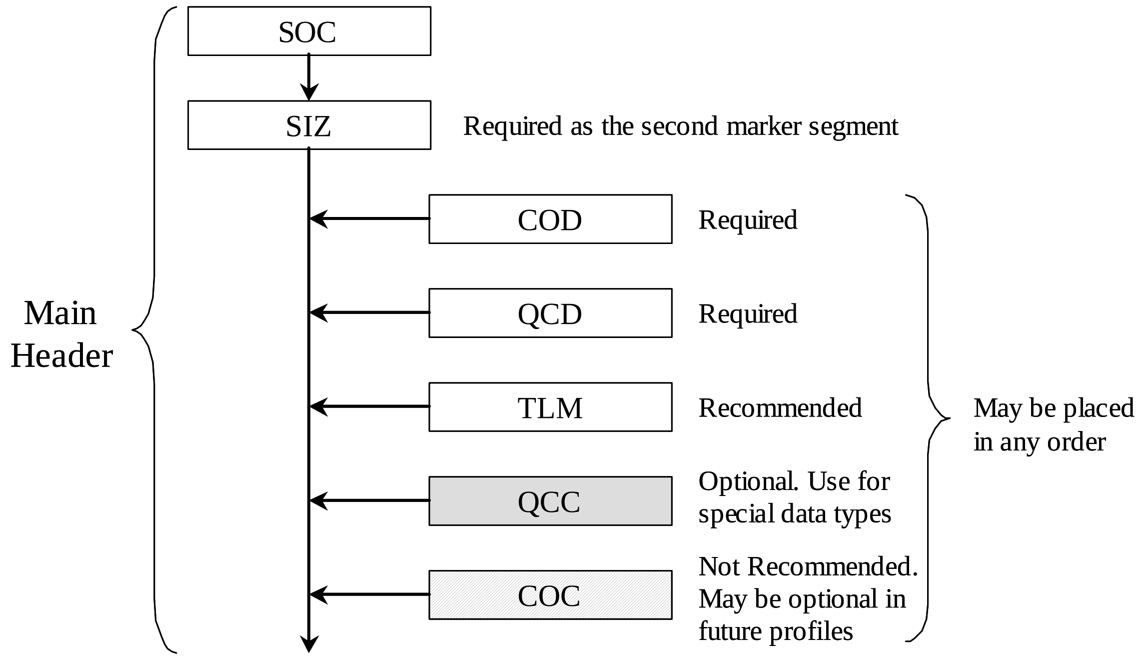

Figure 8-2. Layout of JPEG 2000 Main header

The usage of the QCD and/or QCC marker segments in a tile header is optional. For tiles with small dimensions, performing the recommended number of wavelet decompositions (N_{Levels}) may lead to empty or one-dimensional wavelet subbands. Usage of a QCD for these tiles may be appropriate to alter the quantization. Similarly, a tile may possess very different pixel statistics than other tiles in the image and changing the quantization parameters may be needed. The QCC marker segment may be used for similar reasons on a specific image component. It may 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 profile. It is anticipated that the COD and COC marker segments will be made optional in the tile headers in a future version of this profile.

The JPEG 2000 tile header (for each tile) should contain markers and marker segments and be arranged as shown Figure 8-3:

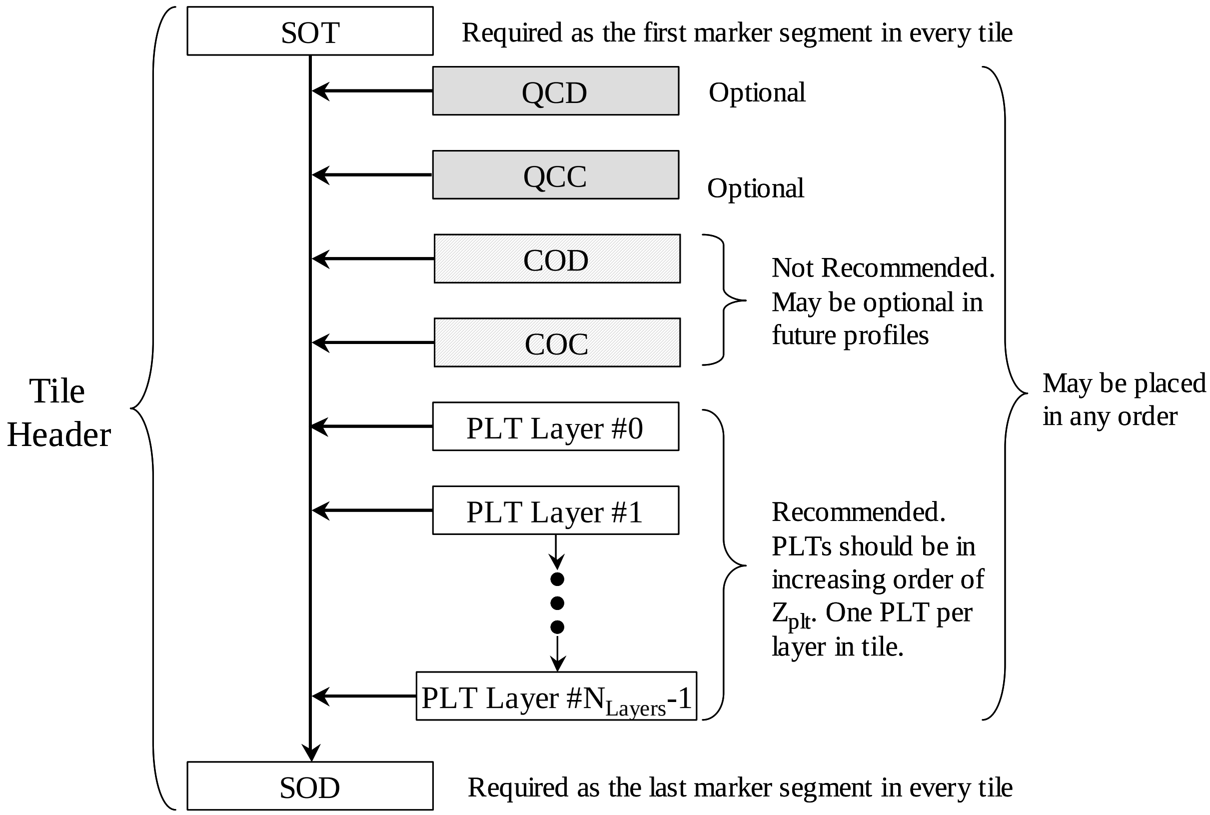

Figure 8-3. Layout of a Single JPEG 2000 Tile Header

The JPEG 2000 image bits (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 8-4. Within the LRCP 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.

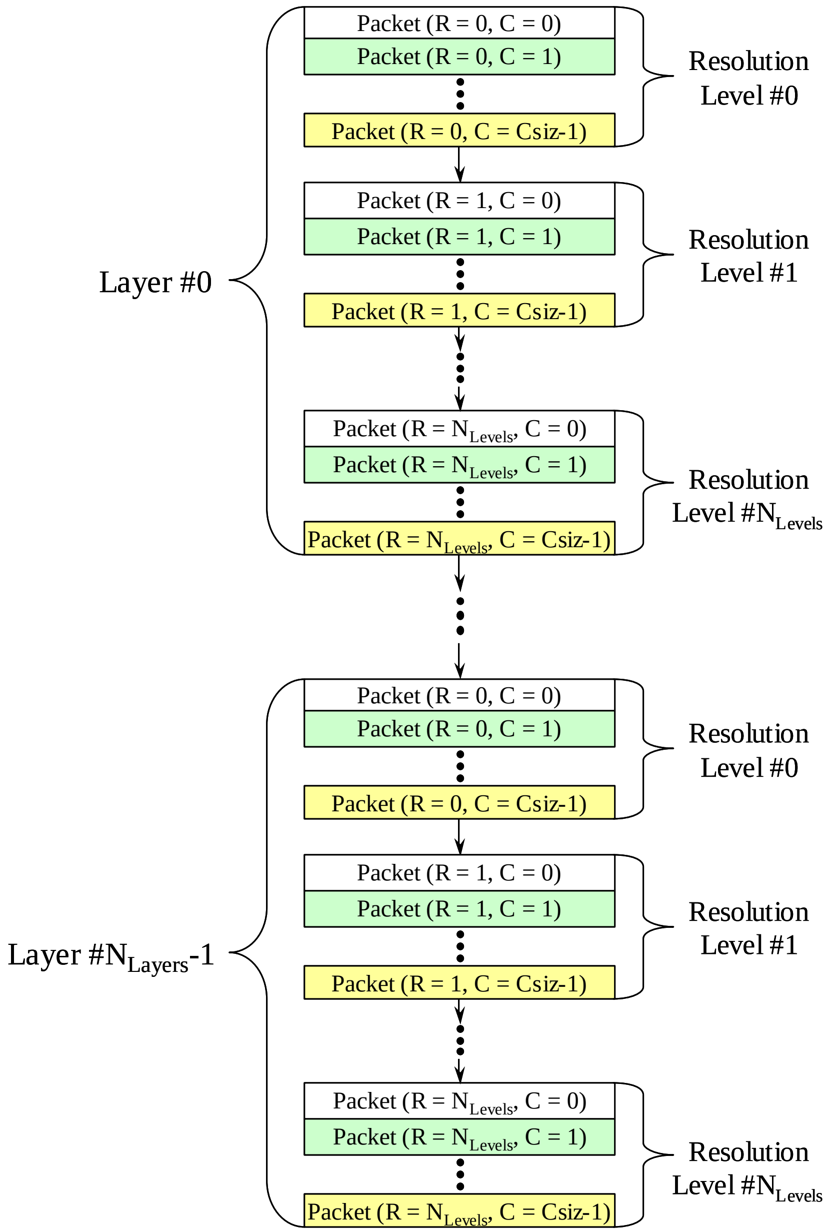

Figure 8-4. Layout of Image Bits for a Single JPEG 2000 Tile

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 8-1 through Table 8-17.

8.3 NPJE Header Information

Table 8-1. JPEG 2000 Codestream Structure (15444-1 Annex A.3)

Marker or Marker Segment	Size (bits)	Contents	Notes
Main Header	variable	See Table 8-2. JPEG 2000 Main Header Contents (15444-1 Annex A.3)	Single header for the entire image at the start of the JPEG 2000 codestream.
Tile Header	variable	See Table 8-3. Tile Header Contents (15444-1 Annex A.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 8-2. 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	$\underline{320 + 24 \cdot Csiz}$	See Table 8-9. Image and tile size (15444-1 Annex A.5.1)	Image and tile size.
COD	112	See Table 8-10. Coding style default (15444-1 Annex A.6.1)	Coding style default. 112 bits using max precincts (recommended).
QCD	9-7I: 296, or $\underline{56 + 48 \cdot N_{Levels}}$ 5-3R: 168, or $\underline{48 + 24 \cdot N_{Levels}}$	See Table 8-11. Quantization default (15444-1 Annex A.6.4)	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.
TLM	variable	See Table 8-13. Tile-part lengths (15444-1 Annex A.7.1)	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 8-2. 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 8-12. Quantization component (15444-1 Annex A.6.5)	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.

Table 8-3. Tile Header Contents (15444-1 Annex A.3)

Marker or Marker Segment	Size (bits)	Contents	Notes
SOT	96	See Table 8-6. Start of tile-part (15444-1 Annex A.4.2)	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 8-11. Quantization default (15444-1 Annex A.6.4)	Quantization default. If present in the tile header, this marker segment overrides the QCD specified in the main header. See notes in Table 8-2.
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 8-12. Quantization component (15444-1 Annex A.6.5)	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 8-2.

Table 8-3. Tile Header Contents (15444-1 Annex A.3)

Marker or Marker Segment	Size (bits)	Contents	Notes
PLT	variable	See Table 8-14. PLT Parameters Content (15444-1 Annex A.7.3)	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. There will be one PLT marker segment per layer.
SOD	16	0xFF93	Start of data marker.

8.4 Markers and Marker Segments Limits and NPJE

The following markers and marker segments are defined in Table 7-1. Table 8-4 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.

Table 8-4. 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.
EOC	0xFFD9	NA	NA	Req.	Required as last marker in the code stream.
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	NR	NR	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.

Table 8-4. Marker and marker segment requirements within a NPJE codestream

Marker	Value	Main header	Tile Header	bit-stream	Restriction/Dependencies
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	NR	NR	NA	This is required if there are progression order changes different from the main or tile header COD. At most, one may appear in any header. May appear in the first tile-part header of a given tile.
TLM	0xFF55	Rec.	NA	NA	Optional, there may be multiple TLM marker segments in the main header.
PLM	0xFF57	NR	NA	NA	Optional, there may be multiple PLM marker segments in the main header.
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	NR	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	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 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	NR	NR	NA	Repeat as many times as desired in the main or tile-part headers. This marker segment has no effect on decoding the bitstream.

8.4.1 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 8-5. Start of Codestream (15444-1 Annex A.4.1)

Parameter	Size (bits)	Values	NPJE	Notes
SOC	16	0xFF4F	0xFF4F (Required)	Start of Codestream.

Table 8-6. Start of tile-part (15444-1 Annex A.4.2)

Parameter	Size (bits)	Values	NPJE	Notes
SOT	16	0xFF90	0xFF90 (Required)	Start of tile part marker code
Lsot	16	10	10	Length of marker segment
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	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 8-7. Start of data marker (15444-1 Annex A.4.3)

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

Table 8-8. End of codestream (15444-1 Annex A.4.4)

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

8.4.2 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 8-9. Image and tile size (15444-1 Annex A.5.1)

Parameter	Size (bits)	Values	NPJE	Notes
SIZ	16	0xFF51	0xFF51 (Required)	Image and tile size marker.
Lsiz	16	41 – 49 190	38 + 3·Csiz	Length of this marker segment in bytes (not including the marker).
Rsiz	16	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010	0000 0000 0000 0010	A value of zero indicates that the full standard capabilities described by ISO/IEC IS15444-1 are required.
Xsiz	32	1 — (2^{31}-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^{31}-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^{31}-2)	0	Horizontal offset from the origin of the reference grid to the left side of the image area.
YOsiz	32	0 — (2^{31}-2)	0	Vertical offset from the origin of the reference grid to top of image area.
XTsiz	32	1 — (2^{31}-1)	1024	Width of one reference tile with respect to the reference grid.
YTsiz	32	1 — (2^{31}-1)	1024	Height of one reference tile with respect to the reference grid.
XTOsiz	32	0 — (2^{31}-2)	0	Horizontal offset from the origin of the reference grid to the left edge of the first tile.
YTOsiz	32	0 — (2^{31}-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	Number of components in the image.
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

Table 8-9. Image and tile size (15444-1 Annex A.5.1)

Parameter	Size (bits)	Values	NPJE	Notes
$XRsiz^i$	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.

8.4.3 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 8-10. Coding style default (15444-1 Annex A.6.1)

Parameter	Size (bits)	Values	NPJE	Notes
COD	16	0xFF52	0xFF52	Coding style default marker.
Lcod	16	Maximal precincts: Lcod = 12 User-defined precincts: Lcod = 13 + N_{Levels}	12	Length of this marker segment in bytes (not including the marker).
Scod	8	0000 0000 – 0000 0111 (Defined in Table 7-8)	0000 0000 (as defined by Table 7-8)	Entropy coder with maximum precinct size. No SOP marker segments shall be used. EPH marker shall not be used.
SGcod	32	Defined below		

Table 8-10. Coding style default (15444-1 Annex A.6.1)

Parameter	Size (bits)	Values	NPJE	Notes
Progression order	8	0000 0000 – 0000 0100	0000 0000 (as defined by Table 7-9)	Layer-resolution level-component-position progression provides SNR progression.
Number of layers (N_{Layers})	16	1 – 65 535	19 for visually lossless (9-7I filter) 20 for Numerically lossless (5-3R)	The producer of a NPJE 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.
Code-block style	8	0000 0000 – 0011 1111 Defined in Table 7-8	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 8-11. Quantization default (15444-1 Annex A.6.4)

Parameter	Size (bits)	Values	NPJE	Notes
QCD	16	0xFF5C	0xFF5C	Quantization default marker.
Lqcd	16	Scalar quantization derived: Lqcd = 5 No quantization: Lqcd = 4 + 3·N_{Levels} Scalar quantization expounded: Lqcd = 5 + 6·N_{Levels}	For 5-3R wavelet: Lqcd = 19 For 9-7I wavelet: Lqcd = 35	Length of this marker segment in bytes (not including the marker). 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 gaurd 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 8-12. Quantization component (15444-1 Annex A.6.5)

Parameter	Size (bits)	Values	NPJE	Notes
QCC	16	0xFF5D	0xFF5D	Quantization component marker.
Lqcc	16	For Csiz < 257 Scalar quantization derived: Lqcd = 6 No quantization: Lqcd = 5 + 3·N_{Levels} Scalar quantization expounded: Lqcd = 6 + 6·N_{Levels} For Csiz ≥ 257 Scalar quantization derived: Lqcd = 7 No quantization: Lqcd = 6 + 3·N_{Levels} Scalar quantization expounded: Lqcd = 7 + 6·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 (not including the marker).
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.

Table 8-12. Quantization component (15444-1 Annex A.6.5)

Parameter	Size (bits)	Values	NPJE	Notes
$SPqcd^i$	8 (5-3R) 16 (9-7I)	Table 7-16	Table 7-17 Table 7-18	With 5-3R wavelet With 9-7I wavelet

8.4.4 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 8-13. Tile-part lengths (15444-1 Annex A.7.1)

Parameter	Size (bits)	Values	NPJE	Notes
TLM	16	0xFF55	0xFF55	Tile-part lengths marker.
Ltlm	16	$$\begin{array}{lcc} & \underline{ST} & \underline{SP} \\ Ltlm = \begin{cases} 4+2\cdot N_{tpm} & 0 & 0 \\ 4+3\cdot N_{tpm} & 1 & 0 \\ 4+4\cdot N_{tpm} & 2 & 0 \\ 4+4\cdot N_{tpm} & 0 & 1 \\ 4+5\cdot N_{tpm} & 1 & 1 \\ 4+6\cdot N_{tpm} & 2 & 1 \end{cases} \end{array}$$ N_{tpm} = number of tile-parts in this TLM marker segment	ST = 0, SP = 1 Ltlm = 4 + 4·N_{tiles} (8 – 65535)	Length of this marker segment in bytes (not including the marker). For this profile, 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	0000 0000 – 0110 0000	0100 0000	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.

Table 8-13. Tile-part lengths (15444-1 Annex A.7.1)

Parameter	Size (bits)	Values	NPJE	Notes
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 ith 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 8-14. PLT Parameters Content (15444-1 Annex A.7.3)

Parameter	Size (bits)	Values	NPJE	Notes
PLT	16	0xFF58	0xFF58	Packet length, tile-part header, marker.
Lplt	16	4 — 65535	4 — 65535	Length of this marker segment in bytes (not including the marker).
Zplt	8	0 — 255	0 — 18	Index of this marker segment relative to all other PLT marker segments in the current header. There shall be one PLT per layer in each tile.
Iplt^i	8 bits repeated as necessary See ISO/IEC IS15444-1: Table A-36	0000 0000 – 1111 1111	0xxx xxxx	Signals that the next seven bits are the last bits indicating the length of the ith packet.
			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.

8.4.5 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 Comment marker (COM) allows for the unstructured data to be included into the file. It is not recommended that either of these markers be used.

Table 8-15. Component registration (15444-1 Annex A.9.1)

Parameter	Size (bits)	Values	NPJE	Notes
CRG	16	0xFF63	0xFF63	Not recommended
Lcrg	16	6 – 65 534		
$Xcrg^i$	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 8-16. Comment (15444-1 Annex A.9.2)

Parameter	Size (bits)	Values	NPJE	Notes
COM	16	0xFF64	0xFF64	Not recommended
Lcom	16	5 – 65 535		
Rcom	16	0 = General binary 1 = General Latin (IS 8859-15:1999)		
$Ccom^i$	8	0 - 255		

8.4.6 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 may be found in NIMA N0106-97. 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 may 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 may 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 bit rates.

8.4.7 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 bit rates 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 bit rates are defined for the full resolution R0 image. When extracting a reduced resolution data set from a JPEG 2000 compressed file, the effective bit rate can be much higher than the target bit rate shown for R0. Therefore, it is important to include a good number of layers at target bit rates well below the useful range for R0. For example, in an 11-bit image, the layer 0 target bit rate of 0.03125 bpp may 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 bit rates are important for quality scalability when extracting reduced resolution data sets.

Table 8-17. Proposed Layers and Applications

Layer	Bits Per Pixel per band (bpppb)
Layer 19 (5-3R filter only)	Numerically Lossless
Layer 18 (Last layer for 9-7I filter)	3.5 bpppb Visual lossless
Layer 17	2.8 bpppb
Layer 16	2.3 bpppb
Layer 15	2.0 bpppb
Layer 14	1.7 bpppb
Layer 13	1.5 bpppb
Layer 12	1.3 bpppb
Layer 11	1.2 bpppb
Layer 10	1.1 bpppb
Layer 9	1.0 bpppb
Layer 8	0.9 bpppb
Layer 7	0.8 bpppb
Layer 6	0.7 bpppb
Layer 5	0.6 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

9 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 9-1 below) and provides information about NSIF file header, image subheader and TRE settings that are related to the JPEG 2000 codestream content.

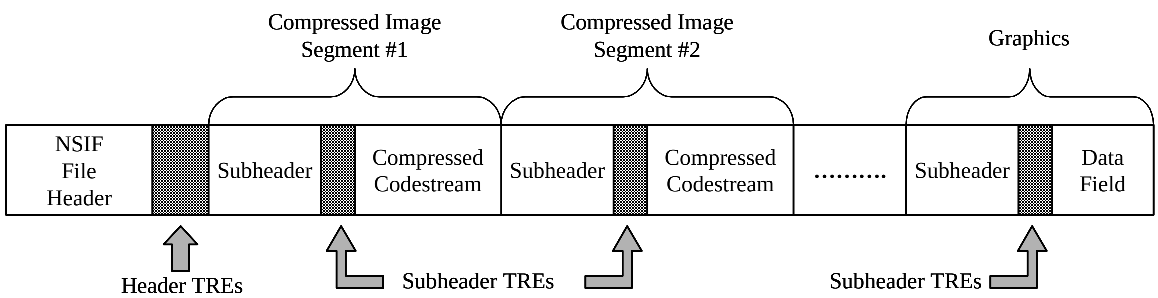

Figure 9-1. NSIF format with embedded JPEG 2000 compressed codestream

9.1 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 may not be present in the NSIF file. Specifically, the Enumerated color spaces and ICC profiles enable more accurate color renditions of data. Table 9-1 described the color interpolation/rendering techniques supported in JPEG 2000 Part 1 and Part 2. NSIF implementation must interpret and apply the color space information.

Table 9-1. 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

9.2 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 under JPEG 2000 change. The goal is to make explicitly clear how marker segment parameters in the JPEG 2000 codestream relate to fields in the NSIF image subheader. 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.

9.2.1 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.

9.2.1.1 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 BIIF CCS serves as the governing reference grid common to all image segment arrays and graphic segment grids that may be included within a BIIF 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 BIIF 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 BIIF 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).

<u>In their uncompressed state,</u> all the samples (pixels) from all image segments within a BIIF 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 a BIIF 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 a BIIF composition, especially one containing multiple image and/or graphic segments.

9.2.1.2 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. In general, this is the difference between internal tiling (JPEG 2000) and external tiling (JPEG DCT), where the tiles are defined in the file format for external tiling applications. This difference affects how certain NSIF image subheader fields must be interpreted.

9.2.1.3 Progression order and Layering

Progression order and layering is another new concept that gives 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 may 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.

9.2.1.4 Subsampling on the reference grid

The JPEG 2000 reference grid provides a means for subsampling 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 subsampling (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 subsampling (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 TREs) 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.

9.2.1.4.1 Reduced Resolutions

When a reduced resolution is parsed from a JPEG 2000 compressed image, it is common practice to use the subsampling 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 (TREs) is based.

9.2.1.4.2 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 TREs) 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 TREs 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 subsampling (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 (col) 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 (resampled 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) Resample (upsample) 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 subsampling/separation factors instead of upsampling prior to compression:
NCOLS=100, NROWS=100, (XRsiz, YRsiz) = (1,2), and the asymmetric/ anamorphic correction flag=yes.

9.2.2 NSIF Image Subheader Fields

The image subheader fields potentially affected when applying JPEG 2000 compression to the image data are listed in Table 9-2. The table includes directions for populating each of the affected fields.

Table 9-2. NSIF Field JPEG 2000 guideline

NSIF Field Name	JPEG 2000 Based Value	Notes
NROWS **NCOLS** N/8 Range: 00000002 - 99999999	NROWS and NCOLS are set to the desired size of the image plane that will be ready for display. For the original image, $\mathrm{NROWS} = Ysiz - YOsiz$ $\mathrm{NCOLS} = Xsiz - XOsiz$ For extraction and repackaging of reduced resolutions (i.e., R1) the NROWS and NCOLS should be set as follows, $\mathrm{NROWS} = \left\lceil \frac{Ysiz}{IMAG\_NEW} \right\rceil - \left\lceil \frac{YOsiz}{IMAG\_NEW} \right\rceil$ $\mathrm{NCOLS} = \left\lceil \frac{Xsiz}{IMAG\_NEW} \right\rceil - \left\lceil \frac{XOsiz}{IMAG\_NEW} \right\rceil$ For non-symmetrical pixels (either defined by metadata or XRsiz or YRsiz) the display should be adjusted to make pixels square. (Refer to sections 9.2.1.4 of this document.)	Number of Rows/Columns. For images that start with a 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.
PVTYPE A/3 Range: B INT SI R C	$max(Ssiz^i) = 0 \Rightarrow$ PVTYPE = B $1 \leqq max(Ssiz^i) \leqq 31 \Rightarrow$ PVTYPE = INT $128 \leqq max(Ssiz^i) \leqq 159 \Rightarrow$ 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 for entire image is determined 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.

Table 9-2. NSIF Field JPEG 2000 guideline

NSIF Field Name	JPEG 2000 Based Value	Notes
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, I1	C8 Do not use mask tables; do not use IC code M8.	Image Compression. Flag to signal the compression algorithm.

Table 9-2. 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 for the compressed image. (The truncation point value for R0. A computed value based on bitstream length for other resolutions.) Nxyz = JPEG 2000 numerically lossless, where "xyz" indicates the expected achieved bit rate (in bits per pixel per band) 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 bit rate (in bits per pixel per band) 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 bit rate (in bits per pixel per band) 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).	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.
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.

Table 9-2. NSIF Field JPEG 2000 guideline

NSIF Field Name	JPEG 2000 Based Value	Notes
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.
NBPR **NBPC** N/4 Range: 0001 - 9999	$\mathrm{NBPR} = \left\lceil \frac{Xsiz - XTOsiz}{XTsiz} \right\rceil$ $\mathrm{NBPC} = \left\lceil \frac{Ysiz - YTOsiz}{YTsiz} \right\rceil$	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. $\mathrm{NPPBH} = \left\lfloor \frac{XTsiz}{IMAG\_NEW} \right\rfloor$ $\mathrm{NPPBV} = \left\lfloor \frac{YTsiz}{IMAG\_NEW} \right\rfloor$	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 may have fewer samples.

Table 9-2. NSIF Field JPEG 2000 guideline

NSIF Field Name	JPEG 2000 Based Value	Notes
NBPP N/2 Range: 01 - 96	$\mathrm{NBPP} = \max_i\left(Ssiz^i \ \&\ 0\mathrm{x}7\mathrm{F}\right)+1$ 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 bit-wise "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.
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.

9.3 Recommended J2KLRA TRE

The J2KLRA TRE was primarily developed to allow providers and users of NPJE 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 may 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 9-3. 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 the value given in the CEL field)	5, BCS-N	R	Variable Calculated for each specific TRE.
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* 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	00000 - 16384
NLAYERS_O	Number of Layers in original image Indicates the number of layers in original image.	3, BCS-N	R	000 - 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 - 999

Table 9-3. Recommended J2KLRA TRE

Field	Name/description	Size bytes and format	Req. or Con.	Value Range
$BITRATE_n$	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 layer is an expected value based on past performance. If there is not a target bit rate, report the achieved bit rate.	9, BCS-A	R	Value 00.000000 – 37.000000
(This is the end of a repeating section.)				
Conditional fields if the data has been parsed; when e.g. ORIG = 1, 3, and 9				
NLEVELS_I	Number of Wavelet levels in this image Indicates the number of wavelet decompositions levels included in this image as defined in COD.	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 SIZ.	5, BCS-N	C	00000 - 16384
NLAYERS_I	Number of Layers in this image Indicates the number of layers in this image as defined in COD.	3, BCS-N	C	000 - 999

* EPJE is described in Appendix D

During primary compression, all TRE fields except the conditional Nxxxx_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).

9.4 Image Segments When Using JPEG 2000

A JPEG 2000 codestream can contain at most 65535 tiles, so every image segment containing JPEG 2000 compressed data will contain at most 65535 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 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 main header QCD/QCC and COD/COC are the same across image segments.

.

This page intentionally left blank.

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.

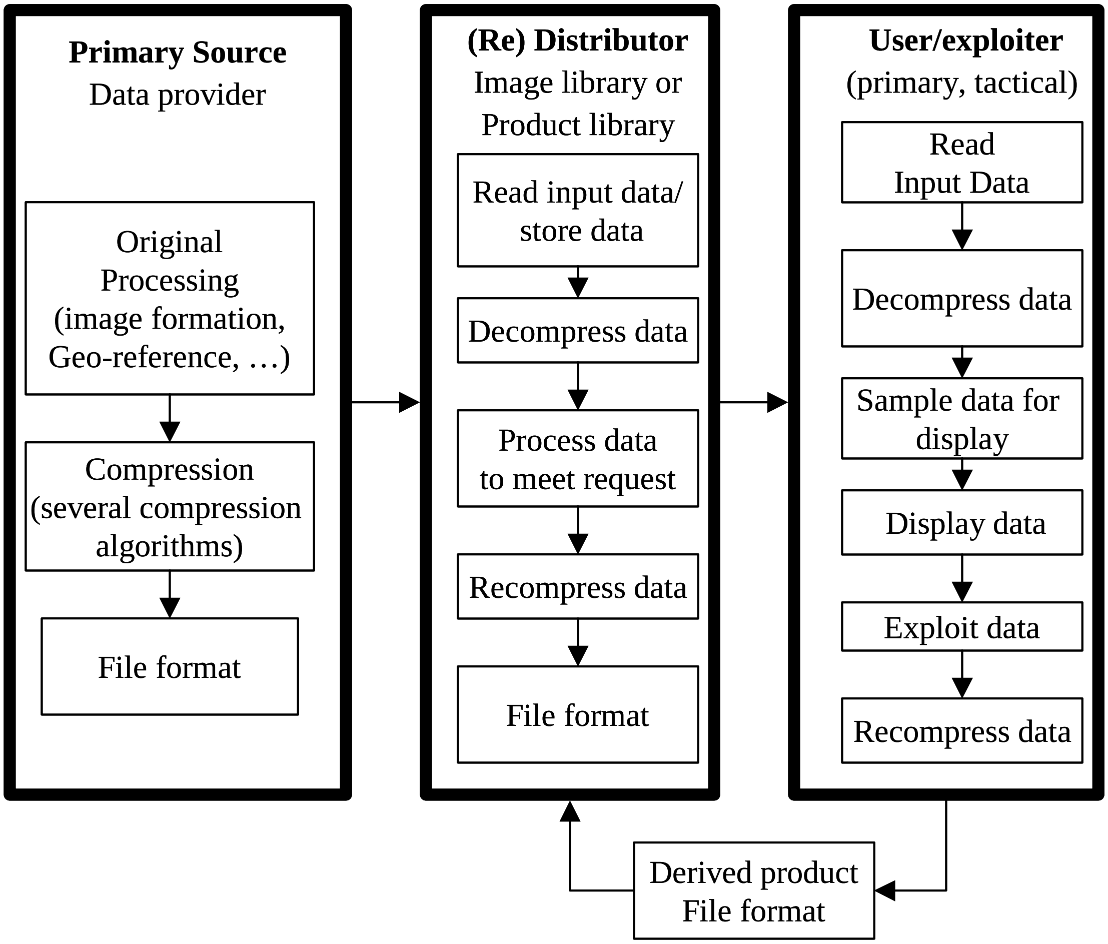

Figure A-1. Current general End-to-End CONOPS

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 in NPJE format, 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.

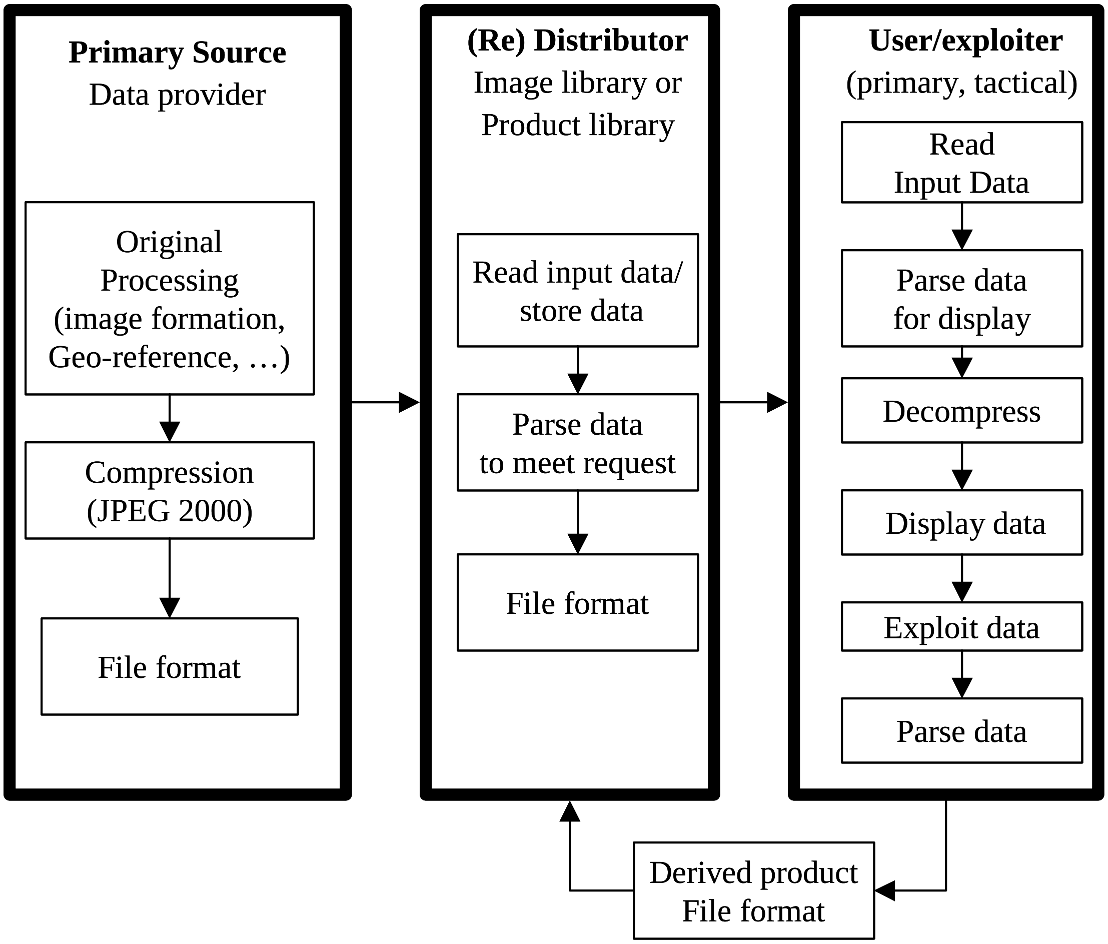

Figure A-2. JPEG 2000 End-to-End CONOPS

A.2 JPEG 2000 Encoder Overview

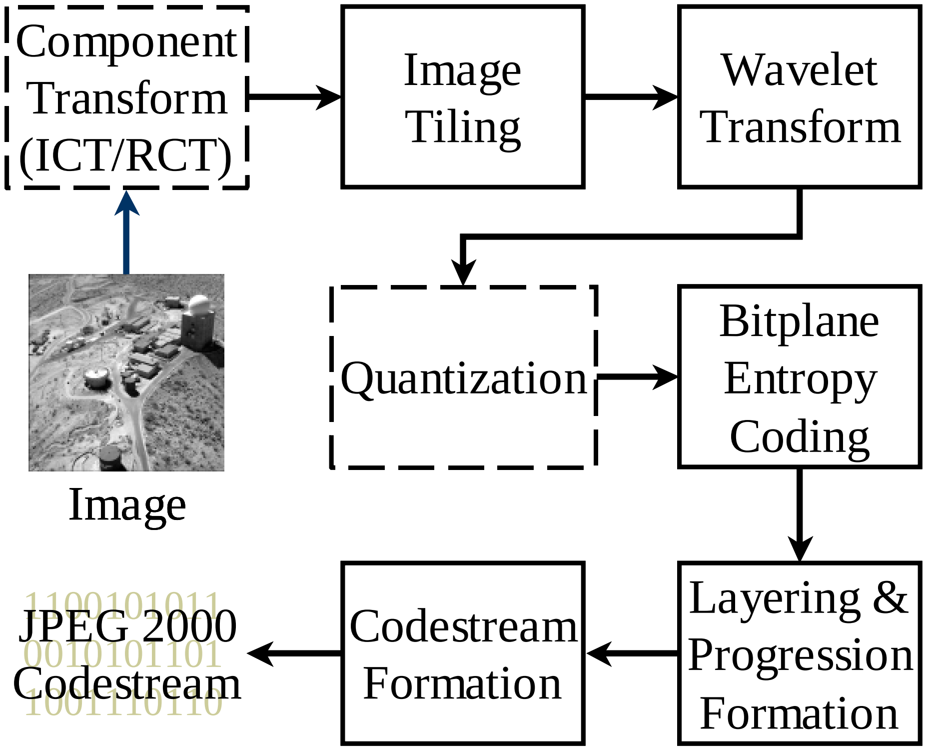

Figure A-3. JPEG 2000 Encoder block diagram

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 subsampling of chrominance bands is performed. With the use of the wavelet transform, chrominance subsampling 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 may 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 may 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 may 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 [N0106], 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 parsing section). 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 may 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 may 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 parsing section) 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 (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

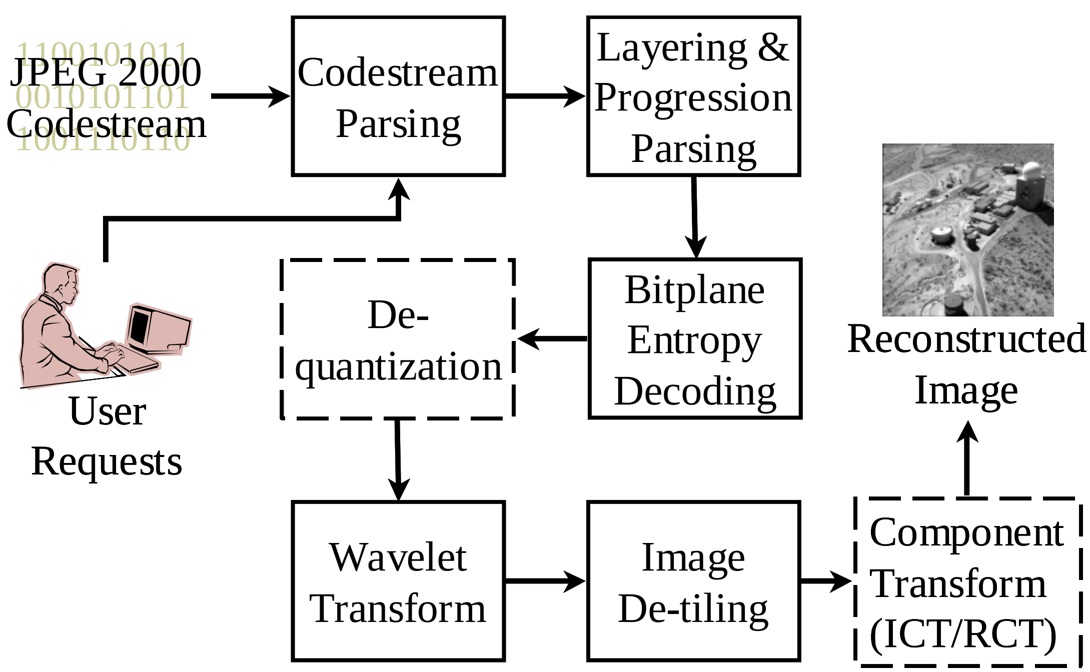

Figure A-4. JPEG 2000 Decoder block diagram

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 (see parsing section).

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 may 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, bitdepth, 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.

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 may 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.

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 subsampling was performed. If a chrominance subsampling 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 stand-alone 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

All data for a single tile in a NPJE 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 $Ptlm^i = Psot^i$.)

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

In generic JPEG 2000 files (not NPJE), tiles may 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.

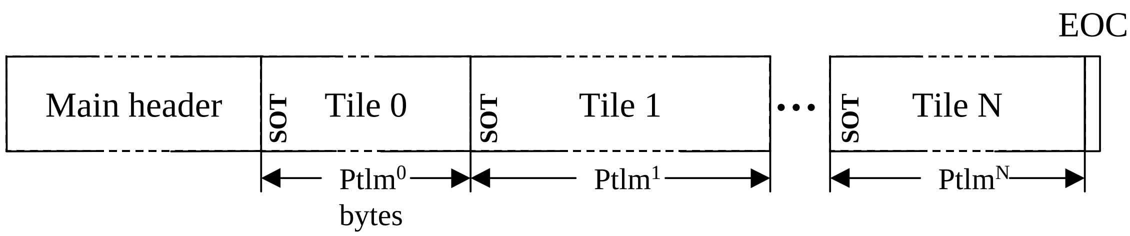

Figure A-5. Locating Tile Boundaries Using Tile Lengths

A single compressed ‘tile’ will contain *all* the component, resolution, and quality data for a particular spatial location. Therefore, descriptions of other processes such as packet parsing and expansion focus on a single tile. With multiple tiles, each 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 recommended in a static NPJE, but may appear in a generic JPEG 2000 codestream.)

The recommended progression is L-R-C-P (layer-resolution-component-precinct), but when decompressing generic JPEG 2000 imagery other progression orders may be encountered.

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 indicate other orderings of the above ‘for’ loops. See ISO/IEC 15444-1 or the Taubman/Marcellin book 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.

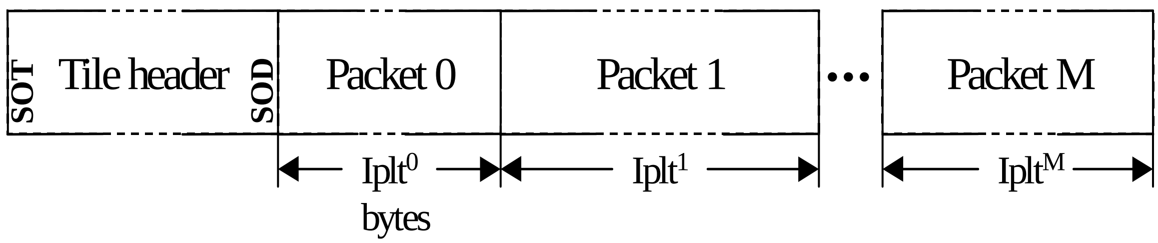

Figure A-6. Locating Packet Boundaries by Length

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 bit rates (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 bit rate 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 layer partitioning 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 bit rate 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 bit rates for the layers that span the full range of possible requirements (in terms of quality vs. bit rate) of the various users of imagery from that provider. This target bit rate 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 NITF J2KLRA TRE that is part of the image segment subheader. A priori knowledge of each layer's target bit rate 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 bit rates, a parser can still get to a requested bit rate 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 bit rate to highest bit rate. 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.

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 bit rates 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 bit rate 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 bit rate 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 bit rates 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 bit rates for the reduced resolution data sets. Having lots of layers at many different bit rates (particularly very low bit-rates) for R0 allows a parser the flexibility to get close to the desired target bit rate 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 bit rates.

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. A codestream with a POC marker (not NPJE) may contain tiles with differing highest resolutions. For example, many tiles may be represented at R5 only, while only a few tiles contain R0 information. When a tile is missing packets for a particular resolution, the application must chose an appropriate decode resolution. A codestream with mosaicking at different resolutions will have tiles whose highest resolutions differ. For example, many tiles may decode at R5 only, while only a few tiles decode all the way to R0. This can only occur when a POC marker segment is present.

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.

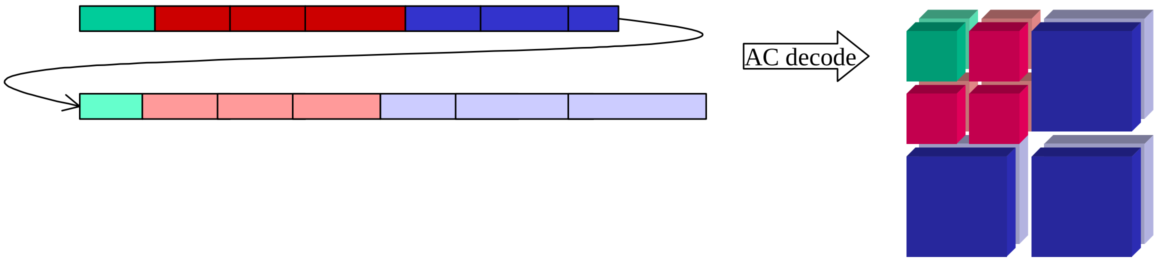

Figure A-7. JPEG 2000 Codestream to Quantized Transform Coefficients

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 accesses. 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 that are parsed and decoded from the bitstream.

Once a quality/truncation point is chosen, the expansion can proceed to the dequantization stage. For R5, it is possible to generate a very high quality image from just a single layer (i.e. truncation point 0.03125). 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 recomputing 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 cooefficient 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.

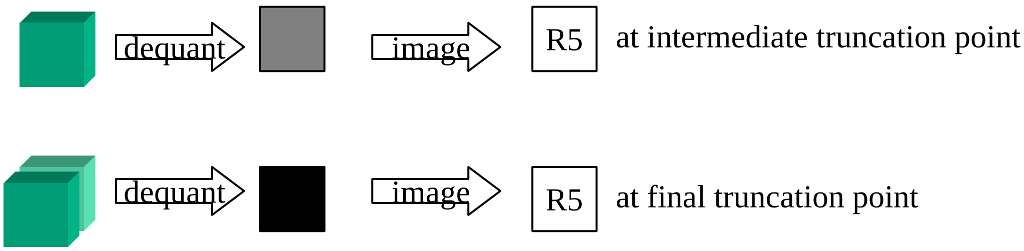

Figure A-8. Quantized Transform Coefficients to Image at R5. Extra layer adds extra quality (darker shade) to LL transform coefficients

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 highpass 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.

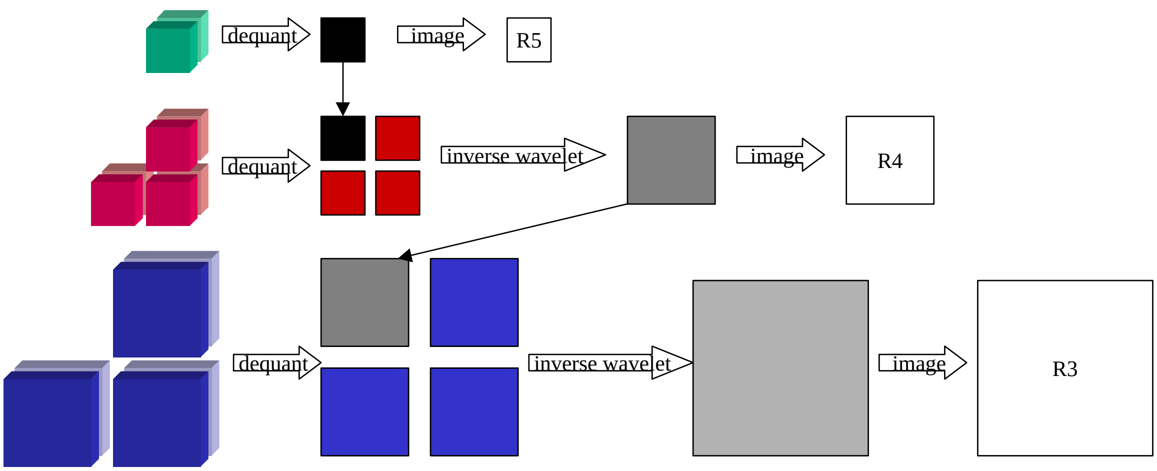

Figure A-9. JPEG 2000 Expansion with R-set Generation

If no (or a limited subset of) reduced resolution data sets are required in a particular application, then the final image generation stage is only performed for the requested data sets. Image display applications, which already have tools in place to 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 then how the data is store. 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 current 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.

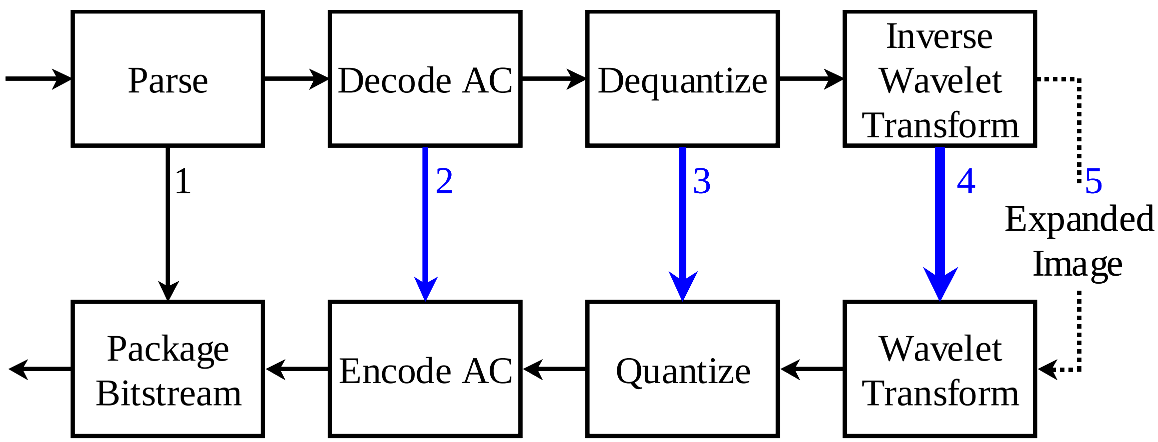

Figure A-10. Repackaging path contrasted with image expansion and recompression

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.

Examples of JPEG 2000 marker segments that may change during standard repackaging of a recommended NPJE 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, 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 TREs must be updated to reflect repackaging modifications. In addition any spatial chipping requires the addition or modification of the ICHIPB TRE.

A.7.1 NPJE Repackaging Changes

Table A-1shows which JPEG 2000 and NSIF header fields may 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. An ‘x’ entry indicates that the particular header field may change during the repackaging indicated in that column. The entry "remove" indicates that particular fields or portions of the codestream are removed entirely. "Remove some" means that some elements of this type 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. Anticipated Codestream Modifications

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

[1] C= conditional change. For NPJE with tilesize = 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. Anticipated Codestream Modifications

Header Field	Spatial chipping	Reduce Quality	Reduce Resolution	Fewer Components	Transcode to EPJE
JPEG 2000					
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			X
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 Image Segment Boundaries

When the area being repackaged crosses an image segment boundary, it may be desirable to reduce the number of image segments during repackaging. This can be accomplished by merging individual repackaged image segment codestreams into one JPEG 2000 codestream.

This 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 TREs (ICHIPB and J2K) should be modified as appropriate for the resegmentation.

Merging across segment boundaries must obey segment size limitations.

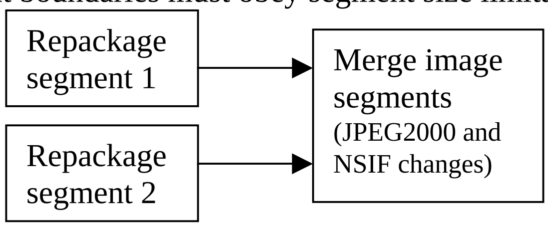

Figure A-11. Merging image segments

A.7.3 Changing JPEG 2000 Progression Order

The NPJE progression order and layering scheme should meet most applications. For applications that require other progression orders or layering, transcoding to new progression orders and layering is completed without actually changing any of the encoded data. Encoded data packets are rearranged to meet the new progression orders and layering. 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 R6+ Generation

At times a resolution lower than R5 is required for purposes such as thumbnail and overview generation. Such lower resolutions can be created by using 9-7I lowpass filtering for subsampling. This is achieved by reconstructing all tiles at R5 and forming a single image without tiles. If this image is too large for processing without tiles, then new large tiles (e.g. 1024x1024) can be created. JPEG 2000 9-7I lowpass filtering is then applied to the retiled image to generate a floating point LL band of reduced image dimension. This LL band can be converted to integer pixel values for R6 and/or can have a successive 9-7I lowpass filter applied to the floating point coefficients. This successive subsampling is continued until the lowest required resolution is generated.

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 a test image the R3 bpp values were:

Layer	Orig 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 Rx 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 contains R0-R5. In SIZ: multiply XRsiz and YRsiz, by 2^x. In COD: subtract x from N_{Levels}. Any QCD/QCCs whether in the main or tile headers may be shortened to reflect the change in N_{Levels}, and the PLTs need to be shortened to eliminate interspersed non-existent packet info. All tile lengths must be updated (TLM and Psots).

The following NSIF fields and TREs will need to be updated appropriately: NROWS/NCOLS, IMAG, COMRAT, J2KLRA TRE and NPPBH/V.

B.2 Spatial Chipping at Tile Boundaries

Spatial chipping is most efficiently performed as the combination of rectangular subsetting, and emptying of 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.

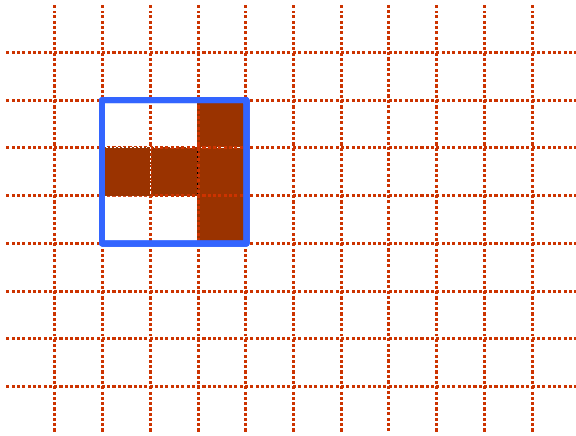

Figure B-1. Chipping: rectangular subset (blue target rubber banding) with four emptied tiles (shown in white). Only shaded tiles retain image data

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 subsampling 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, ILOC, and NBPR/NBPC. An ICHIPB TRE must be added (or modified) to correctly identify where the chip resides relative to the originally collected image.

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)
  1. Tile offsets remain at 0
  2. Image offset = Original reference grid offset for the chip modulo tile size
- Very unusual case:
  1. Image offset = Reference grid chip offset
  2. 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.

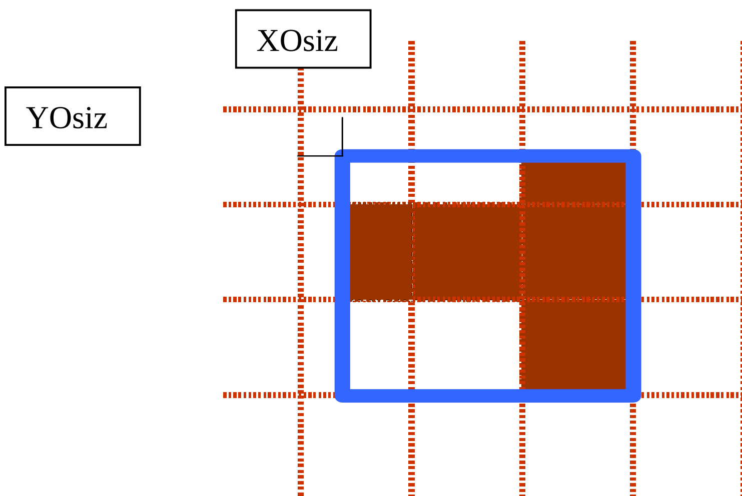

Figure B-2. Spatial Chipping at Non-Tile Boundaries: Only shaded tiles retain image data

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 bit-rate. 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. For NPJE, there are a fixed number of packets per layer, so computing the truncation point is simple. 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.

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)

In order to promote the wide interoperability of JPEG-2000 codestreams, codestream restrictions are introduced. "Codestream Restrictions" have two profiles, profile-0 and profile-1. The case of "No Restrictions" meaning conforming to JPEG-2000 Part-1 standard can be called Profile-2. 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 as defined below.

Table C-1. Codestream Restrictions

Restrictions	Profile-0	Profile-1
SIZ Marker		
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	$XTsiz/min(XRsiz^i, YRsiz^i) \le 1024$ XTsiz=YTsiz or one tile for the whole image: YTsiz+YTOsiz>=Ysiz XTsiz+XTOsiz>=Xsiz
Image & tile origin	$XO_{SIZ}=YO_{SIZ}=XTO_{SIZ}=YTO_{SIZ}=0$	$XO_{SIZ}, YO_{SIZ}, XTO_{SIZ}, YTO_{SIZ} < 2^{31}$
RGN marker segment	SPrgn ≤ 37	SPrgn ≤ 37
Sub-sampling	XR_{SIZ}^i = 1, 2, or 4 YR_{SIZ}^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 s=1 for segmentation symbols	No restriction

Table C-1. Codestream Restrictions

Restrictions	Profile-0	Profile-1
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^"number of decomposition levels" in SPcod or SPcoc, for I = component 0 to 2	If one tile is used for whole image, (Xsiz – XOsiz)/D(I) ≤ 128 and (Ysiz – YOsiz)/D(I) ≤ 128 where D(I)= 2^"number of decomposition levels" in SPcod or SPcoc, for I = component 0 to 3
Parsability	If the POC marker is present, the POC marker shall have RS_{POC}^{0}=0 and CS_{POC}^{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) 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

Appendix D Exploitation Preferred JPEG 2000 Encoding

EPJE (Exploitation Preferred JPEG 2000 Encoding) is a reordering of NPJE, which facilitates rapid access to a variety of resolution levels when working in a NFS disk environment. 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 40Kx40K images) at lower resolutions. EPJE is a formatting alternative and produces identical decoded data to NPJE. The following paragraphs give a brief overview of this format. More complete details may appear in a later amendment to this document.

As alternate formats with identical decodable content, EPJE and NPJE use the same structural basis for tranform 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 equi-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 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.

This page intentionally left blank.