INTERNATIONAL
REGISTERED GRAPHICAL ITEM
CLASS: BIIF PROFILE

ISO/IEC
BIIF PROFILE
BPJ2K01.00
Amendment 1

Version 1.0

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

BIIF Profile for JPEG 2000
Version 01.00 Amendment 1
(BPJ2K01.00 AMD1)

13 September 2007

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

Foreword

The International Standard (IS) 12087-5:1998, Basic Image Interchange Format (BIIF) provides guidance for creating profiles of BIIF. At present, two profiles of BIIF have been established: 1) the model profile of BIIF as specified in ISO 12087-5; and 2) the 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 North Atlantic Treaty Organization (NATO) Standardization Agreement (STANAG) 4545 and STANAG 4609 (Edition 3), 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 Amendment 1

This standard is normative.

This page intentionally left blank.

Introduction

The BIIF Profile for JPEG 2000 Version 01.00 Amendment 1 (BPJ2K01.00AMD1) supplements the BIIF Profile for JPEG 2000 Version 01.00 (BPJ2K01.00, 30 July 2004) with the addition of a new JPEG 2000 encoding profile. BPJ2K01.00 as amended by this document 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 BPJ2K01.00AMD1was cooperatively developed by 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 these files within the user community. STANAG 4545 and (U.S.) MIL-STD-2500C are specific user community documents describing the implementation of the NSIF01.00 BIIF Profile and now, the BPJ2K Profile.

The BPJ2K01.00AMD1 adds an *optional* JPEG 2000 encoding profile for use within NSIF01.00. It is not a normative requirement for implementers of NSIF01.00 to support this profile. The BPJ2K01.00AMD1, however, is normative for certain systems described within STANAG 4609 (Edition 3) Air Reconnaissance Primary Imagery Data Standard. The profile described by this amendment is the LVSD Preferred JPEG 2000 Encoding (LPJE). LVSD (Large Volume Streaming Data) is a new NATO designation for sensors that capture very large array imagery at rates of one frame per second (1fps) and faster for extended periods of time.

Compliant NSIF decoders are required to decode all compliant data within the limits of the BPJ2K01.00 and their BIIF compliance level (Section 7). The LPJE profile described by this amendment is *not* compliant with the NSIF Preferred JPEG 2000 Encoding (NPJE) profile (Section 8) in the BPJ2K01.00, *nor* is it compliant to the JPEG 2000 Part 1 Profile 1 within ISO/IEC 15444-1 (see Appendix C in the BPJ2K01.00). The system requirements needed to implement LPJE can be significant. In light of this, not all systems are required to support LPJE. See STANAG 4609 (Edition 3) for further details.

Appendix E LVSD Preferred JPEG 2000 Encoding

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

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

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

The LPJE profile supports the performance requirements of LVSD systems. LPJE is a superset of the NPJE profile, which means that an NPJE (or EPJE) compressed image is also an LPJE compressed image. The LPJE profile, however, offers a wider range of compression options than either the NPJE or EPJE profiles. It is possible to create LPJE complaint files that lie outside the boundaries defined within Profile 1 in JPEG 2000 Part 1 (see Annex C of BPJ2K01.00); therefore, strictly NSIF01.00 compliant systems will not be able to decode all possible LPJE files.

E.1 Scope

This profile is intended for the compression of literal imagery (*i.e.,* panchromatic, color, detected SAR, multispectral, thermal IR, *etc.*) within STANAG 4609 (Edition 3)[1]. This

[1] Note: From this point forward STANAG 4609 will be used synonymously with STANAG 4609 (Edition 3) unless explicitly indicated otherwise.

profile may be optionally used within STANAG 4545, but NSIF01.00 compliant systems are not required to create or interpret LPJE encoded imagery. The LPJE profile is not expected to handle non-literal imagery types (*i.e.*, I/Q data, M/P data, VPH data, Elevation data, Location-Grid data, *etc.*). Future versions of this profile may incorporate other sensor modalities such as hyperspectral. In this case, it is expected that the multiple component transform framework of JPEG 2000 Part 2 (ISO/IEC 15444-2) will be included once the requirements for hyperspectral imagery are defined.

E.1.1 General

The BPJ2K01.00 profile specifies allowed data values and ranges for JPEG 2000 header and subheader fields contained in an NSIF01.00 file. The BPJ2K01.00AMD1 profile (LPJE) extends these allowed data values and ranges for the JPEG 2000 header and subheader fields. These extensions exceed the bounds set forth in the NPJE and EPJE profiles of BPJ2K01.00. The NPJE and EPJE profiles represent constrained subsets of the LPJE profile defined by this amendment. As such, an LPJE encoded imagery frame may not be BPJ2K01.00 compliant, however an NPJE or EPJE encoded frame is compliant with the LPJE profile.

BPJ2K01.00 conformant systems are required to decode any JPEG 2000 Part 1 Profile 1 codestream that conforms to the CLEVEL constrains of that system. The LPJE profile does not conform to the JPEG 2000 Part 1 Profile 1. While it is possible to create LPJE conformant codestreams that do comply with JPEG 2000 Part 1 Profile 1, not all LPJE codestreams will necessarily be compliant.

E.1.2 Position Within the Graphical Item Register

BPJ2K01.00AMD1 extends the JPEG 2000 profiles found in BPJ2K01.00. It 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 graphical item registration information is as follows:

Graphical Item Class:	BIIF Profile
Graphical Item Long Name:	BIIF Profile for JPEG 2000 Version 01.00 Amendment 1
Graphical Item Short Name:	BPJ2K01.00AMD1
Sponsoring Authority:	TBD (The United Kingdom sponsors this Profile through their membership in the ISO committee)
Preparing Authority:	TBD (This document was prepared for the sponsoring authority by the STANAG 4609 Custodian; U.S.; Secretary of the Air Force,

Information Dominance Directorate,
Reconnaissance Systems Division (SAF/AQIJ))

E.1.3 User Requirements and Scenario

BPJ2K01.00AMD1 is designed to promote interoperability between LVSD systems and computer libraries that store their data. Building upon the existing BPJ2K01.00 profile, this amendment provides a new optional profile for use within NSIF as well. A common imagery compression format for LVSD sensors will enable secondary systems and products to leverage this data at reduced cost; particularly, as LVSD systems proliferate. Moreover, adoption of a common compression profile for LVSD imagery systems will foster interoperability, facilitating reuse and sharing of information.

The LPJE compression profile is designed to operate over a wide range of applications, and is sufficiently flexible to accommodate both hardware and software based codec implementations. Adoption of JPEG 2000 for compression of LVSD imagery provides the necessary functionality for these systems. A multi-resolution, multiple quality representation enables efficient bandwidth management—a strict requirement in LVSD systems. In addition, the capability to compress a wide array of imagery with varying number of spectral bands (components) and bit depths is required as well.

The LPJE compression profile is not meant to solve *all* compression needs for LVSD. For instance, future LVSD sensors will likely carry HD (high definition) FMV (full motion video) sensors and large array framing cameras. In these cases, temporal video compression technologies, like MPEG-2 or H.264, may be more appropriate (see STANAG 4609). It is anticipated that future LVSD systems will make use of multiple compression technologies (JPEG 2000 and H.264 as an example) in systems comprised of multiple sensor types.

The LPJE profile defines a codestream format for frame-based imagery compression for LVSD systems only. Other standards documentation will subsequently address additional aspects of LVSD systems. These aspects include but are not limited to: HD FMV, metadata and metadata formats, file formats, streaming imagery and video transport formats, etc. STANAG 4609 serves as a reference for related LVSD standards, and one should consult STANAG 4609 for these pertinent standards documents.

E.2 References

Normative References:

The normative references in BPJ2K01.00 (see Section 2) remain normative references for BPJ2K01.00AMD1. Additional normative references specific to this Amendment are included below. These references are not normative with respect to BPJ2K01.00.

The following documents contain provisions that, through reference in this text, constitute provisions of the BPJ2K01.00AMD1. Applicability is limited to the specific instance of the referenced document only; 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 below. 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*
STANAG 4609 (Edition 3)	NATO Digital Motion Imagery Standard
ISO/IEC 15444-1 AMD1:2006	JPEG 2000 Image Coding System -- Part 1: Image Coding System: Amendment 1

Non-Normative References:

The following non-normative references are referenced in this amendment.

Referenced Documents:	*Title*
DCI System Specification	Digital Cinema System Specification Version 1.1

E.3 Definitions

For the purposes of the BPJ2K01.00AMD1 profile, the definitions shown in ISO/IEC 15444-1 Section 3, apply.

E.4 Abbreviations

The abbreviations defined in BPJ2K01.00 apply to this amendment. The following additional abbreviations are defined for BPJ2K01.00AMD1.

C-P-R-L	Component-Position-Resolution-Layer
LPJE	LVSD Preferred JPEG 2000 Encoding
LVSD	Large Volume Streaming Data
P-C-R-L	Position-Component-Resolution-Layer
ROI	Region of Interest
R-P-C-L	Resolution-Position-Component-Layer

E.5 Conformance

Conformance is a necessary step towards achieving interoperability amongst 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 standard minimum requirements and includes test JPEG 2000 codestreams. Products that conform to the BPJ2K01.00AMD1 profile shall also meet the conformance requirements of ISO/IEC 12087-5.

E.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.

E.7 JPEG 2000 Profile and Limitations

Section 7 of BPJ2K01.00 contains the codestream restrictions defined in ISO/IEC 15444-1, Profile 1. All compliant NSIF/BIIF/NITF decoders are required to properly decode compressed data within the limits of that profile. The LPJE profile defined in this Annex allows for compressed data that does not fit within the limitations of ISO/IEC 15444-1, Profile 1. Therefore, BPJ2K01.00 compliant decoders may not be able to correctly interpret all LPJE codestreams. Any LPJE codestream, however, that conforms to the limitations of ISO/IEC 15444-1 Profile 1, will be correctly interpreted by compliant BPJ2K01.00 decoders. Implementers are cautioned to exercise care when creating LPJE codestreams that do not fit within ISO/IEC 15444-1 Profile 1. Interoperability should always be a primary concern and LPJE codestreams should conform to Profile 1 of ISO/IEC 15444-1 wherever possible.

All LPJE compliant encoders must correctly set the capability parameter, Rsiz, within the SIZ marker segment (see section E.8.6). If an LPJE codestream meets the Profile 1 restrictions of ISO/IEC 15444-1, it shall be so indicated through proper setting of the Rsiz parameter. This will allow a BPJ2K01.00 compliant decoder to decode that particular codestream. Codestreams that do not fit within Profile 1 of ISO/IEC 15444-1 will be marked to indicate that decoding resources beyond Profile 1 are required to properly interpret those codestreams. This will alert BPJ2K01.00 compliant decoders that they may not be able to properly decode these codestreams.

All compliant LPJE decoders shall be able to properly decode compressed data within the limits of the LPJE profile. Any LPJE compliant decoder will be capable of decoding BPJ2K01.00 compliant codestreams. All LPJE compliant encoders must produce compressed data that is within the limits of the LPJE profile. It is recommended that encoders adhere to the recommendations in the following section (Section E.8). There is *no requirement* that LPJE compliant encoders be able to produce BPJ2K01.00 compliant codestreams, although this functionality is desired.

E.8 LVSD Preferred JPEG 2000 Encoding (LPJE)

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

E.8.1 LPJE Overview

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

E.8.1.1 Resolution Scalability

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

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

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

E.8.1.2 Quality Scalability

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

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

E.8.1.3 Parsing and Chipping

Parsing and chipping regions of interest within an image to create new files or during JPIP streaming sessions is a very common task. Design tradeoffs with regard to parsing and chipping will affect other performance metrics. In particular, reduced-resolution roam and zoom are sensitive to choices made here. JPEG 2000 offers two methods to enable region-of-interest parsing and chipping: Tiles and precincts. The interactions between resolutions, layers, tiles and precincts, various JPEG 2000 pointer marker segments (TLM, PLT, etc.), and computer disk access patterns is complex. These interactions, however, can determine application performance and memory usage. The NPJE and EPJE profiles of BPJ2K01.00 will serve to aid in understanding this.

In BPJ2K01.00 the EPJE profile was introduced to improve reduced-resolution roam and zoom relative to the NPJE profile. The differences between EPJE and NPJE are small and it is possible to “repackage” (see below) the two profiles with a minimal amount of effort. The difference between the profiles is in the disk access patterns required to parse out a reduced-resolution version of an image. In EPJE the portions of the codestream needed to extract a reduced-resolution image tend to be located close to each other within a compressed file. Thus, when the computer operating system reads data off its disk drive the needed information is obtained through a minimum number of seeks. This is not true for NPJE. With NPJE, the operating system must seek to multiple non-contiguous locations within the file. This significantly slows reading the data into memory and impairs performance.

This problem remains in LVSD systems. In fact, with increased frame sizes, it becomes a serious concern. To improve application performance, LPJE allows the use of precincts. The NPJE profile uses simple tiles (*i.e.* each tile is comprised of one tile-part) to enable region-of-interest parsing and chipping. NPJE allows easy spatial chipping and parsing on tile boundaries at full image resolution. This helps support library functionality. The EPJE profile introduced the use of multiple tile-parts to break up simple tiles and collect together codestream data of like resolution. This improves reduced-resolution roam and zoom performance. EPJE, however, impairs tile-based chipping of large images since the codestream data for a tile is now distributed throughout the file. Users, however, are more willing to wait for the generation of a chipped image product rather than suffer poor application performance while scanning through a large image.

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

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

E.8.1.4 Region of Interest Encoding

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

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

E.8.2 JPEG 2000 Repackaging and Transcoding

JPEG 2000 enabled applications will occasionally need to create new JPEG 2000 files from existing ones. This requires that the JPEG 2000 codestream be modified in one or more ways so that a properly formatted, compliant codestream is the result. In this Annex we will refer to “repackaging” and “transcoding”. The difference between JPEG 2000 repackaging and transcoding is a matter of degree. Repackaging refers to simple codestream manipulations that do not require decoding of any entropy encoded wavelet coefficients (marker segment and packet header manipulation may be necessary). For example, converting NPJE and EPJE codestreams between one another requires only repackaging. Tile-based chipping of an image encoded in the NPJE or EPJE profiles requires only repackaging. Reduction of the number of layers in a codestream requires repackaging.

Transcoding refers to more sophisticated codestream manipulations where entropy coding of wavelet coefficients might be changed and possibly recalculation of wavelet coefficients may occur. LPJE allows a much broader range of compression options to be selected compared to NPJE and EPJE. As such, there are more opportunities for the need of sophisticated codestream manipulation. Adding tiling to a codestream that was not tiled requires transcoding. Wavelet coefficients must be recomputed at the tile boundaries. Adding precincts to a codestream that does not use precincts or altering the precincts in a codestream that uses them requires processing somewhere between repackaging and transcoding. Changing the JPEG 2000 codestream may result in codestream parameters falling outside of the ranges recommended in this Appendix. There is no requirement for LPJE encoders, decoders, or JPIP servers to maintain compliance to this profile after repackaging or transcoding procedures.

E.8.3 LPJE Codestream Structure

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

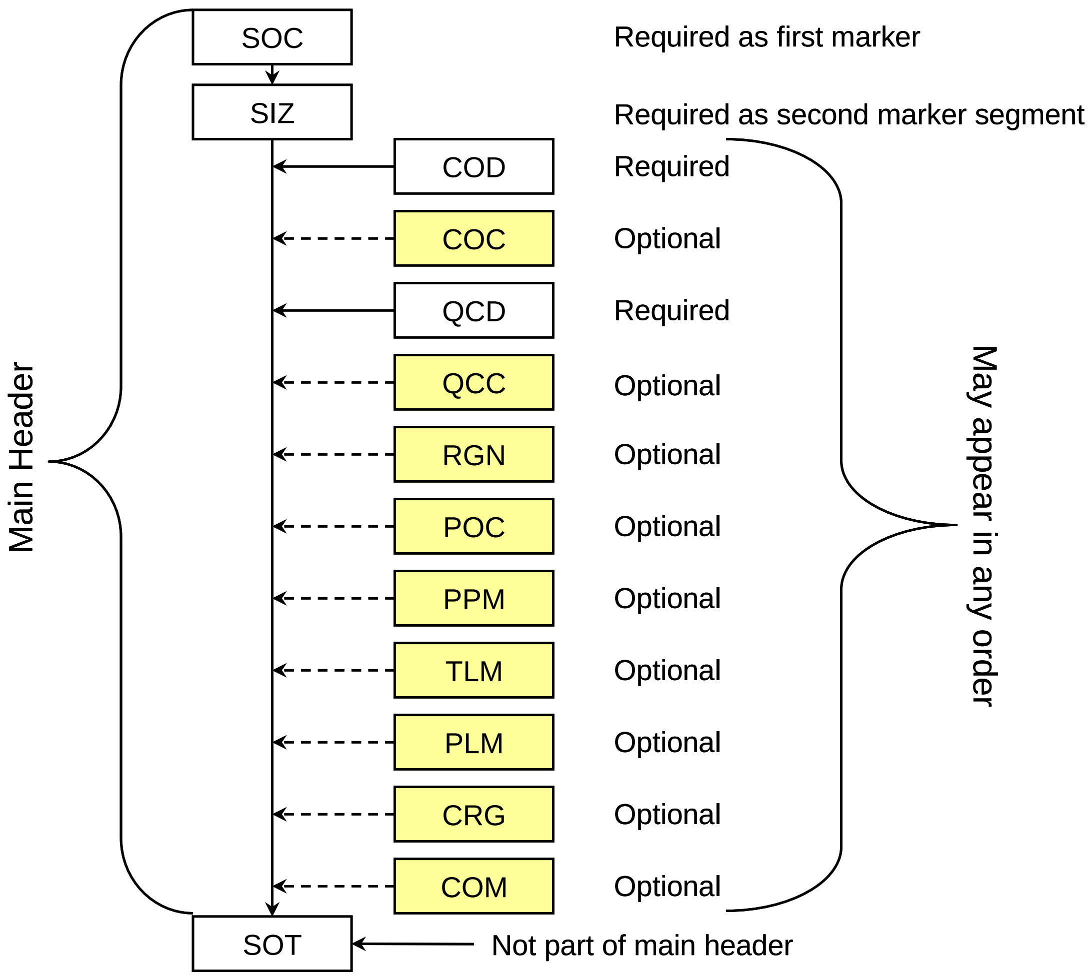

Figure E-1 Layout of LPJE JPEG 2000 Main header

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

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

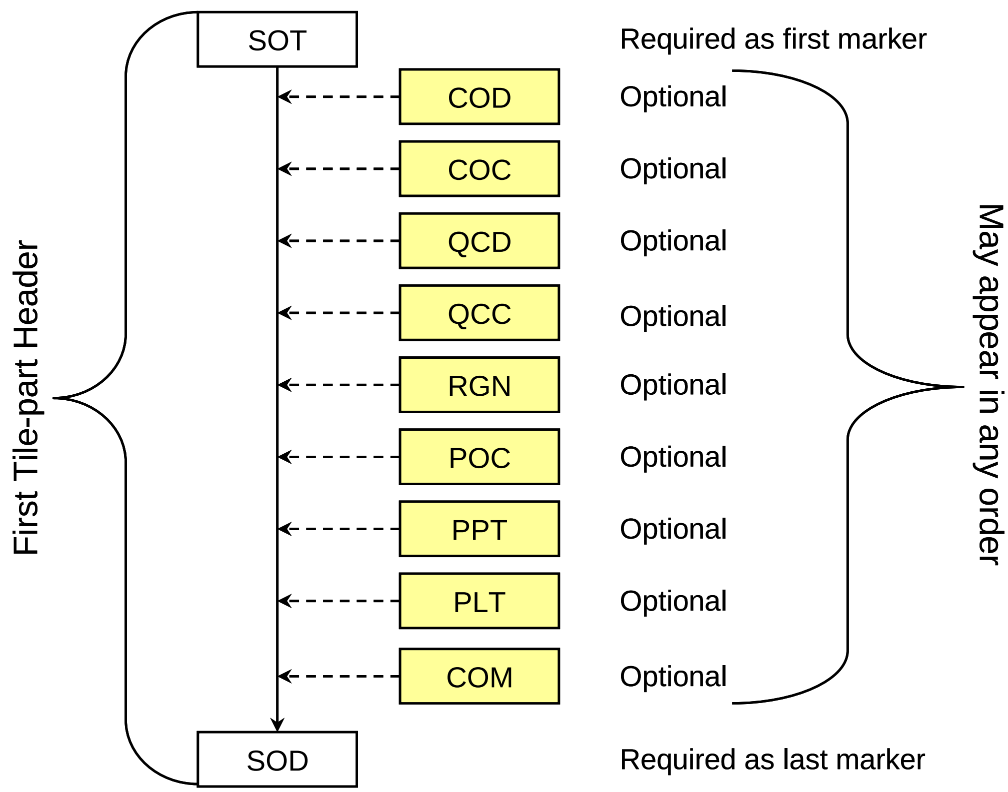

Figure E-2 Layout of a Single JPEG 2000 Tile-part Header (first tile-part only)

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

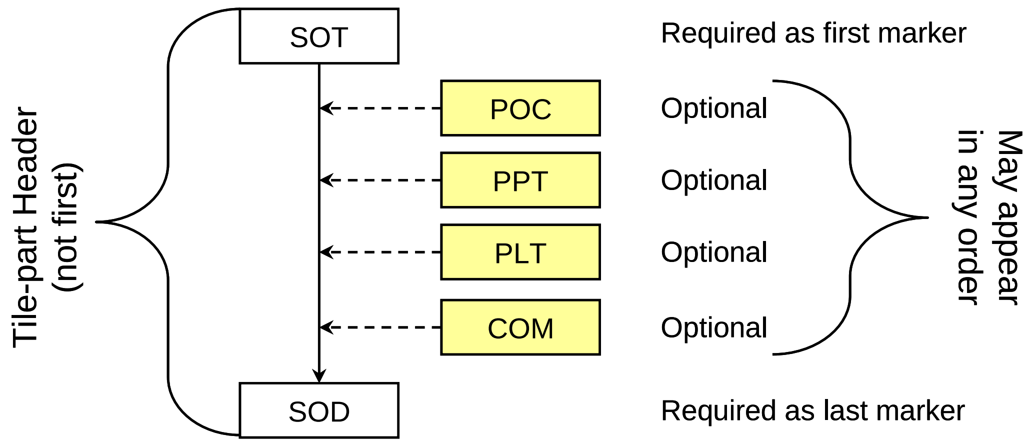

Figure E-3 Layout of a Single JPEG 2000 Tile-part Header (not first tile-part)

E.8.4 Markers and Marker Segments Limits and LPJE

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

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

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

Marker	Value	Main Header	Tile-part Header	Bitstream	Restrictions/Remarks
SOC	0xFF4F	Req.	NA	NA	Required as first marker in the file. Main headers starts immediately after SOC.
SOT	0xFF90	NA	Req.	NA	Required as first marker in each tile-part. Tile-part header occurs after SOT marker segment and before SOD marker.
SOD	0xFF93	NA	Req.	NA	Last marker in each tile-part header. Immediately precedes tile-part data.
EOC	0xFFD9	NA	NA	Req.	Required as the last marker in the codestream.
SIZ	0xFF51	Req.	NA	NA	Required as second marker segment in main header. Immediately follows SOC.
COD	0xFF52	Req.	Opt.	NA	One and only one required in main header. Optionally no more than one main appear in first tile-part header.
COC	0xFF53	Opt.	Opt.	NA	Optional. No more than one COC per component in main header and first tile-part header.
RGN	0xFF5E	Opt.	Opt.	NA	Optional. No more than one per component in main header or first tile-part header.
QCD	0xFF5C	Req.	Opt.	NA	One and only one required in main header. No more than one in first tile-part headers.
QCC	0xFF5D	Opt.	Opt.	NA	Optional. No more than one per component in main header or first tile-part header.
POC	0xFF5F	Opt.	Opt.	NA	Optional. No more than one in the main or any tile-part header. Must appear in a tile-part header before the packets which it describes.
TLM	0xFF55	Rec.	NA	NA	Recommended unless encoding image in a single tile-part
PLM	0xFF57	NR	NA	NA	Not recommended, but allowed. Main header only, PLT is preferred.

Marker	Value	Main Header	Tile-part Header	Bitstream	Restrictions/Remarks
PLT	0xFF58	NA	Rec.	NA	Recommended. Multiple PLTs may appear per tile. Must appear in a tile-part header prior to the packets whose lengths are described.
PPM	0xFF60	NR	NA	NA	Not recommended, but allowed. Use of a PPM prohibits use of PPT and in-bitstream packet headers.
PPT	0xFF61	NA	NR	NA	Not recommended, but allowed. All packet headers for the tile in which PPT appears must be included in the PPT. Use of PPT prohibits use of PPM and in-bitstream packet headers for the tile.
SOP	0xFF91	NA	NA	NR	Not recommended, but allowed. May be used in front of each packet. Whether or not a SOP marker is used for a given packet, the Nsop parameter is incremented. If packet headers are moved into a PPM or PPT marker segment, the SOP marker segment may appear immediately before packet bodies in the bitstream.
EPH	0xFF92	NR	NR	NR	Not recommended, but allowed. If used, they must appear after each packet header. If packet headers are moved into PPM or PPT marker segments, the EPH markers are moved as well.
CRG	0xFF63	NR	NA	NA	Nor recommended, but allowed. Informational use only, implementations need not do anything with this information. Only one may appear in main header and it applies to all tiles.
COM	0xFF63	Opt.	Opt.	NA	Optional use. Informational only, no data necessary to decode the codestream or metadata needed to interpret the codestream may be placed in a COM marker segment.

E.8.5 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 E-2 Start of Codestream (15444-1 Annex A.4.1)

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

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

Parameter	Size (bits)	Values	LPJE	Notes
SOT	16	0xFF90	0xFF90	Start of tile part marker code
Lsot	16	10	10	Length of marker segment
Isot	16	0 – 65,534	0 – 65,534	Tile index in raster order starting at index 0
Psot	32	0, 14 – (2^{32} –1)	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 finalized (written to disk or encapsulated). If Psot=0 is maintained in an NITF file, the current tile part will be interpreted to extend to the end of the current NITF image segment. If Psot=0 is maintained in any JPEG 2000 format, the tile-part is interpreted to extend to the end of the file.
Tpsot	8	0 – 254	0 – 254	Tile-part index
Tnsot	8	0 – 255	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 E-4 Start of data marker (15444-1 Annex A.4.3)

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

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

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

E.8.6 Fixed Information Marker Segment

The SIZ 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 E-6 Image and tile size (15444-1 Annex A.5.1)

Parameter	Size (bits)	Values	LPJE	Notes
SIZ	16	0xFF51	0xFF51	Image and tile size marker
Lsiz	16	41 – 49,190	41 – 49,190	Length of marker segment
Rsiz	16	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010 0000 0000 0000 0011 0000 0000 0000 0100	0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0010	Rsiz = 0000 0000 0000 0000 indicates that the full capabilities described by ISO/IEC IS15444-1 are required. Rsiz = 0000 0000 0000 0001 indicates that the codestream is Profile 0 compliant. Rsiz = 0000 0000 0000 0010 indicates that the codestream is Profile 1 compliant. Rsiz = 0000 0000 0000 0011 and Rsiz = 0000 0000 0000 0100 see below*
Xsiz	32	$1 – (2^{32} – 1)$	$1 – (2^{32} – 1)$	Reference grid width
Ysiz	32	$1 – (2^{32} – 1)$	$1 – (2^{32} – 1)$	Reference grid height
XOsiz	32	$0 – (2^{32} – 2)$	$0 – (2^{32} – 2)$	Horizontal offset from the origin of the reference grid to the left side of the image area
YOsiz	32	$0 – (2^{32} – 2)$	$0 – (2^{32} – 2)$	Vertical offset from the origin of the reference grid to the top of the image area

Parameter	Size (bits)	Values	LPJE	Notes
XTsiz	32	$1 – (2^{32} – 1)$	$256 – (2^{32} – 1)$	Tile width on reference grid. LPJE limits tile sizes to no smaller than 256 and recommends a maximum tile size 8,192.
YTsiz	32	$1 – (2^{32} – 1)$	$256 – (2^{32} – 1)$	Tile height on reference grid. LPJE limits tile sizes to no smaller than 256 and recommends a maximum tile size 8,192.
XTOsiz	32	$0 – (2^{32} – 2)$	$0 – (2^{32} – 2)$	Horizontal offset from the origin of the reference grid to the left edge of the first tile
YTOsiz	32	$0 – (2^{32} – 2)$	$0 – (2^{32} – 2)$	Vertical offset from the origin of the reference grid to the top edge of the first tile
Csiz	16	1 – 16,384	1 – 16,384	The number of components in the image.
$Ssiz^i$	8	0000 0000 – 1010 0101	Unsigned: 0 – 37 Signed: 128 – 165	0xxx xxxx Unsigned data 1xxx xxxx signed data x000 0000 – x010 0101 bit depth of data = value + 1
$XRsiz^i$	8	1 – 255	1 – 255	Horizontal subsampling on the reference grid with respect to the i^{th} component
$YRsiz^i$	8	1 – 255	1 – 255	Vertical subsampling on the reference grid with respect to the i^{th} component

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

Tile sizes for LPJE are restricted such that the *effective* tile size should be no smaller than 256 x 256. The effective tile size, ($tcx1 – tcx0, tcy1 – tcy0$) see Annex B ISO/IEC 15444-1, is computed by taking into account sampling on the reference grid. For $XRsiz^i = YRsiz^i = 1$, this means that $XTsiz \geq 256$ and $YTsiz \geq 256$. For sampling factors $XRsiz^i$ and $YRsiz^i$ greater than 1, XTsiz and YTsiz must be correspondingly increased. Thus the lower bound on XTsiz and YTsiz will be 256. The only exception to this rule is if the original image size is smaller than 256 x 256 and $XRsiz^i = YRsiz^i = 1$. In this case the image should be encoded as a single tile.

The lower bound of a 256 x 256 tile was chosen to accommodate hardware encoders that rely on tiling to reduce complexity. Encoders must not generate tiles that are too small and impair reduced-resolution roam and zoom performance. There is no upper bound on tile sizes other than that imposed by ISO/IEC 15444-1 (2^{32}-1). LPJE recommends an upper bound on tile size of 8,192 x 8,192 pixels. This value was chosen to limit the memory requirements for encoders and decoders. While efficient software encoder and decoder implementations exist that can handle very large tiles (100,000 x 100,000 pixels), not all implementations can be relied upon to be this efficient. Implementers are urged to verify the efficiency of their JPEG 2000 codecs when attempting to implement tile sizes larger than 8,192 x 8,192 in size.

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

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

$$N_{Levels} \leq \lfloor \log_2(\min(Xsiz, Ysiz)) \rfloor \qquad \text{Equation E-1}$$

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

E.8.7 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 E-7 gives the COD marker segment. The COD marker segment is required in the main header of the codestream and it contains the default encoding parameters applied to all components and tiles unless it is overridden (see COC below).

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

Parameter	Size (bits)	Values	LPJE	Notes
COD	16	0xFF52	0xFF52	Coding style default marker
Lcod	16	Lcod = 12 (maximal precincts) Lcod = 13 + N_{Levels} (user-defined)	12 – 45	Length of marker segment
Scod	8	0000 0000 – 0000 0111 (see Table E-8)	0000 0000 – 0000 0111	All coding style parameters allowed in LPJE
SGcod	32	Defined below		
Progression order	8	0000 0000 – 0000 0100	0000 0000 – 0000 0100 (see Table E-9)	LPJE allows all progression orders
Number of layers (N_{Layers})	16	1 – 65,535	1 – 65,535	LPJE encoders should include enough layers to enable quality scalability across all resolutions. Approximately 10 – 20 for a monochrome image. Hardware implementations may use less. Maximum number of layers should not exceed 50 unless special circumstances exist.
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	0 – 32	Number of decomposition levels, N_{Levels}
Code-block width	8	0000 0000 – 0000 1000	0000 0000 – 0000 1000	LPJE allows all code-block sizes
Code-block height	8	0000 0000 – 0000 1000	0000 0000 – 0000 1000	LPJE allows all code-block sizes
Code-block style	8	0000 0000 – 0011 1111 (see Table E-10)	0000 0000 – 0011 1111	All code-block styles are allowed
Transformation	8	0000 0000 – 0000 0001	0000 0001 0000 0000	5-3 reversible filter for numerically lossless applications 9-7 irreversible filter for applications that do not require lossless data

Parameter	Size (bits)	Values	LPJE	Notes
Precinct size	Variable	NA or 0000 0000 – 1111 1111	NA or 0000 0000 – 1111 1111	Not present if precincts not used. Otherwise user-defined precincts follow.

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

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

Value (bits)	Coding style
0000 0xx0	Entropy coder, precincts with PPx = 15 and PPy = 15 (maximal precincts)
0000 0xx1	Entropy coder with user-defined precincts
0000 0x0x	No SOP marker segments used
0000 0x1x	SOP marker segments may be used
0000 00xx	No EPH marker used
0000 01xx	EPH marker shall be used

Table E-9 Progression order (SGcod or Ppoc parameters, 15444-1 Annex A.6.1)

Value (bits)	Progression order
0000 0000	Layer-resolution level-component-position progression (L-R-C-P)
0000 0001	Resolution level-layer-component-position progression (R-L-C-P)
0000 0010	Resolution level-position-component-layer progression (R-P-C-L)
0000 0011	Position-component-resolution level-layer progression (P-C-R-L)
0000 0100	Component-position-resolution level-layer progression (C-P-R-L)

Table E-10 Code-block style (SPcod and SPcoc parameters, 15444-1 Annex A.6.1)

Value (bits)	Code-block style
00xx xxx0	No selective arithmetic coding bypass
00xx xxx1	Selective arithmetic coding bypass
00xx xx0x	No reset of context probabilities on coding pass boundaries
00xx xx1x	Reset context probabilities on coding pass boundaries
00xx x0xx	No termination on each coding pass
00xx x1xx	Termination on each coding pass
00xx 0xxx	No vertically causal context
00xx 1xxx	Vertically causal context
00x0 xxxx	No predictable termination
00x1 xxxx	Predictable termination
000x xxxx	No segmentation symbols are used
001x xxxx	Segmentation symbols are used

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

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

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

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

The NPJE and EPJE profiles do not recommend the use of the COC marker segment, but they are allowed within the LPJE profile. Table E-11 gives the COC marker segment along with the LPJE parameter ranges. Note that the progression order, number of layers and use of the multiple component transform cannot be modified on a per component basis.

Table E-11 Coding style component (15444-1 Annex A.6.2)

Parameter	Size (bits)	Values	LPJE	Notes
COC	16	0xFF53	0xFF53	Coding style component marker
Lcoc	16	9 (max precincts & Csiz < 257) 10 (max precincts & Csiz ≥ 257) $10 + N_{Levels}$ (user-defined & Csiz < 257) $11 + N_{Levels}$ (user-defined & Csiz ≥ 257	9 – 43	Length of marker segment
Ccoc	8 16	0 – 255 (8 bits, Csiz < 257) 0 – 16,383 (16 bits, Csiz ≥ 257)	0 – 255 0 – 16,383	Index of component to which the marker segment applies
Scoc	8	0000 0000 – 0000 0001 (see Table E-12)	0000 0000 – 0000 0001	All coding style parameters allowed
SPcoc	Variable	Defined below		
Number of decomposition levels (N_{Levels})	8	0 – 32	0 – 32	Number of decomposition levels, N_{Levels}
Code-block width	8	0000 0000 – 0000 1000	0000 0000 – 0000 1000	LPJE allows all code-block sizes
Code-block height	8	0000 0000 – 0000 1000	0000 0000 – 0000 1000	LPJE allows all code-block sizes
Code-block style	8	0000 0000 – 0011 1111 (see Table E-10)	0000 0000 – 0011 1111	All code-block styles are allowed
Transformation	8	0000 0000 – 0000 0001	0000 0001 0000 0000	5-3 reversible filter for numerically lossless applications 9-7 irreversible filter for applications that do not require lossless data
Precinct size	Variable	NA or 0000 0000 – 1111 1111	NA or 0000 0000 – 1111 1111	Not present if precincts not used. Otherwise user-defined precincts follow.

Table E-12 Scoc coding style parameters (15444-1 Annex A.6.2)

Value (bits)	Coding style
0000 0000	Entropy coder, precincts with PPx = 15 and PPy = 15 (maximal precincts)
0000 0001	Entropy coder with user-defined precincts

The region of interest marker segment (RGN) is shown in Table E-13. This marker segment is used for region of interest coding. Encoders may shift the wavelet coefficients corresponding to a spatial region of interest up (by left bit-shift) above the most significant bitplane of the remaining background wavelet coefficients. This has the effect that these wavelet coefficients will be encoded first before all other wavelet coefficients. The RGN marker segment alerts the decoder that this has been done so that the process may be reversed during decoding. ROI encoding will most likely be a feature of some LVSD systems so it has been included in the LPJE profile.

Table E-13 Region of interest (15444-1 Annex A.6.3)

Parameter	Size (bits)	Values	LPJE	Notes
RGN	16	0xFF5E	0xFF5E	Region of interest marker
Lrgn	16	5 – 6	5 – 6	Length of marker segment
Crgn	8 16	0 – 255 (8 bits, Csiz < 257) 0 – 16,383 (16 bits, Csiz ≥ 257)	0 – 255 0 – 16,383	Index of component to which the marker segment applies
Srgn	8	0000 0000	0000 0000	Implicit ROI (maximum shift method)
SPrgn	8	0 – 255	0 – 37	Limit data to 38 bit signed range

The QCD marker segment is required in the main header to indicate the quantization step-sizes (for the 9-7I wavelet) or reversible dynamic range (for the 5-3R wavelet) that is valid for all tile-parts. The QCD marker segment applies to all tiles and components unless it is overridden (see QCC below).

Table E-14 Quantization default (15444-1 Annex A.6.4)

Parameter	Size (bits)	Values	LPJE	Notes
QCD	16	0xFF5C	0xFF5C	Quantization default marker
Lqcd	16	No quantization: $Lqcd = 4 + 3 \cdot N_{Levels}$ Scalar quantization derived: Lqcd = 5 Scalar quantization expounded: $Lqcd = 5 + 6 \cdot N_{Levels}$	For 5-3R wavelet: Lqcd = 4 – 100 For 9-7I wavelet: Lqcd = 5 – 197	Length of this marker segment. For the 5-3R wavelet, no quantization is used. For the 9-7I wavelet, scalar derived or expounded quantization is used. LPJE recommends expounded quantization.
Sqcd	8	xxx0 0000 xxx0 0001 xxx0 0010 000x xxxx - 111x xxxx (see Table E-15)	xxx0 0000 xxx0 0001 xxx0 0010 000x xxxx - 111x xxxx	With 5-3R filter: No quantization With 9-7I filter: Recommend scalar expounded quantization, scalar derived is allowed Recommend two guard bits for all applications
SPqcd[i]	8 (5-3R) 16 (9-7I)	variable	Table E-16 Table E-17	With 5-3R wavelet With 9-7I wavelet

Table E-15 Quantization values (Sqcd and Sqcc parameters, 15444-1 Annex A.6.4)

Value (bits)	Quantization Style	SPqcd[i] Size (bits)	SPqcd or SPqcc usage
xxx0 0000	No quantization (5-3R wavelet)	8	Table E-16
xxx0 0001	Scalar derived (values signaled for NL_{LL} subband only)	16	Table E-17
xxx0 0010	Scalar expounded. One step size signaled for each subband.	16	Table E-17
000x xxxx – 111x xxxx	Number of guard bits (0 – 7)		

Table E-16 Reversible step size (SPqcd^{i} and SPqcc^{i} parameters, 15444-1 Annex A.6.4)

Value (bits)	Reversible step size values
0000 0xxx – 1111 1xxx	Exponent, ε_b, of the reversible dynamic range signaled for each subband. See equation E.2 in ISO/IEC 15444-1 (note 15444-1 makes reference to equation E.5, this is incorrect).

Table E-17 Quantization step size (SPqcd^{i} and SPqcc^{i} parameters, 15444-1 Annex A.6.4)

Value (bits)	Quantization step size values
xxxx x000 0000 0000 – xxxx x111 1111 1111	Mantissa, μ_b, of the quantization step size value. See Equation E.3 in ISO/IEC 15444-1.
0000 0xxx xxxx xxxx – 1111 1xxx xxxx xxxx	Exponent, ε_b, of the quantization step size value. See Equation E.3 in ISO/IEC 15444-1.

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

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

Table E-18 gives the QCC marker segment along with the LPJE parameter ranges. The LPJE profile recommends two guard bits for both 5-3R and 9-7I wavelet processing. Two guard bits are sufficient for the vast majority of imagery. There may arise some circumstances where two guard bits are not sufficient but these are very rare.

Table E-18 Quantization component (15444-1 Annex A.6.5)

Parameter	Size (bits)	Values	LPJE	Notes
QCC	16	0xFF5D	0xFF5D	Quantization component marker
Lqcc	16	For Csiz < 257 No quantization: Lqcd = 5 + 3·N_{Levels} Scalar quantization derived: Lqcd = 6 Scalar quantization expounded: Lqcd = 6 + 6·N_{Levels}	For Csiz < 257 For 5-3R wavelet: Lqcd = 5 – 101 For 9-7I wavelet: Lqcd = 6 – 198	Length of marker segment
		For Csiz ≥ 257 No quantization: Lqcd = 6 + 3·N_{Levels} Scalar quantization derived: Lqcd = 7 Scalar quantization expounded: Lqcd = 7 + 6·N_{Levels}	For Csiz ≥ 257 For 5-3R wavelet: Lqcd = 6 – 102 For 9-7I wavelet: Lqcd = 7 – 199	
Sqcc	8	xxx0 0000 xxx0 0001 xxx0 0010 000x xxxx - 111x xxxx (see Table E-15)	xxx0 0000 xxx0 0001 xxx0 0010 000x xxxx - 111x xxxx	With 5-3R filter: No quantization With 9-7I filter: Recommend scalar expounded quantization, scalar derived is allowed Recommend two guard bits for all applications
SPqcc[i]	8 (5-3R) 16 (9-7I)	variable	Table E-16 Table E-17	With 5-3R wavelet With 9-7I wavelet

The POC marker segment, Table E-19, is used to change progression orders within a codestream. POC marker segments require a good understanding of JPEG 2000 codestream construction and a sophisticated JPEG 2000 codec. ISO/IEC 15444-1 Profile 1 compliant decoders are required to handle POC marker segments. The POC marker segment allows full control over the ordering of codestream data within a file. For most applications the POC marker segment is not necessary, one progression order will suffice for all codestream data. The POC marker segment is allowed for optional use within the LPJE profile. The NPJE and EPJE profiles do not recommend its use.

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

Table E-19 Progression order change (15444-1 Annex A.6.6)

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

Parameter	Size (bits)	Values	LPJE	Notes
$Ppoc^i$	8	0000 0000 – 0000 0100 (see Table E-9)	0000 0000 – 0000 0100	Progression order for i^{th} progression. One value for each progression change.

E.8.8 Pointer Marker Segments

The pointer markers segments are used to gain quick access to data for parsing, chipping, and decoding. These marker segments define either 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.

Table E-20 Tile-part lengths (15444-1 Annex A.7.1)

Parameter	Size (bits)	Values	LPJE	Notes
TLM	16	0xFF55	0xFF55	Tile-part lengths marker
Ltlm	16	$Ltlm = \begin{array}{lcc} & ST & SP \\ \left\{ \begin{array}{l} 4+2\cdot N_{tpm} \\ 4+3\cdot N_{tpm} \\ 4+4\cdot N_{tpm} \\ 4+4\cdot N_{tpm} \\ 4+5\cdot N_{tpm} \\ 4+6\cdot N_{tpm} \end{array} \right. & \begin{array}{c} 0 \\ 1 \\ 2 \\ 0 \\ 1 \\ 2 \end{array} & \begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{array} \end{array}$ N_{tpm} = number of tile-parts in this TLM marker segment	6 – 65,535	Length of marker segment. See Table E-21.
Ztlm	8	0 – 255	0 – 255	Index of this marker segment relative to all other TLM marker segments present in the current header. Multiple TLMs are allowed in LPJE.
Stlm	8	0x00 0000 0x01 0000 0x10 0000 00xx 0000 01xx 0000	0x00 0000 0x01 0000 0x10 0000 00xx 0000 01xx 0000	See Table E-21. LPJE allows all combinations of ST and SP parameters.
$Ttlm^i$	0 if ST = 0 8 if ST = 1 16 if ST = 2	Tiles in order 0 – 254 0 – 65,534	NA (Stlm = 0x00 0000) 0 – 254 (Stlm = 0x01 0000) 0 – 65,534 (Stlm = 0x10 0000)	LPJE allows simple tiles with one-tile part and tiles with multiple tile-parts

Parameter	Size (bits)	Values	LPJE	Notes
$Ptlm^i$	16 if SP = 0 32 if SP = 1	14 – 65,535 14 – (2^{32} – 1)	14 – 65,535 (Stlm = 00xx 0000) 14 – (2^{32} – 1) (Stlm = 01xx 0000)	The length, in bytes, from the beginning of the SOT marker of the i^{th} tile-part to the end of the codestream data for that tile-part. There should be one Ptlm for every tile-part.

Table E-21 Stlm size parameters (15444-1 Annex A.7.1)

Value (bits)	Size parameters
0x00 0000	ST = 0; Ttlm parameter is 0 bits, only one tile-part per tile and the tiles are in index order without omission or repetition.
0x01 0000	ST = 1: Ttlm parameter 8 bits
0x10 0000	ST = 2: Ttlm parameter 16 bits
00xx 0000	SP = 0; Ptlm parameter 16 bits
01xx 0000	SP = 1; Ptlm parameter 32 bits

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

The PLM marker segment (packet lengths, main header) collects packet length information across all tile-parts and places the lengths in the main header. The marker segment is shown in Table E-22. It is possible to overload the PLM marker segment due to the size of the $Nplm^i$ field being 8 bits. If there are more than 255 packets in a tile-part or the total amount of length data for a given tile-part requires more than 255 bytes *for any given tile-part*, then the PLM marker segment may not be used. The amount of information contained in a PLM (or PLT) marker segment can be considerable. Therefore, the length data may be spread across more than one PLM or PLT marker segment. The Zplm (or Zplt) parameter is an index that is used to reassemble the data into the correct order.

Table E-22 Packet lengths, main header (15444-1 Annex A.7.2)

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

Table E-23 Iplm, Iplt packet lengths (15444-1 Annex A.7.2)

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

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

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

Table E-24 Packet lengths, tile-part header (15444-1 Annex A.7.3)

Parameter	Size (bits)	Values	LPJE	Notes
PLT	16	0xFF58	0xFF58	Packet lengths, tile-part header, marker
Lplt	16	4 – 65,535	4 – 65,535	Length of marker segment
Zplt	8	0 – 255	0 – 255	Index of this marker segment relative to all other PLT marker segments in the current tile-part header. Multiple PLTs are allowed in LPJE.
$Iplm^i$	variable	variable (see Table E-23)	variable (see Table E-23)	Length of the i^{th} packet in this tile. If packet headers are stored with the packet bodies this length includes the packet header. If packet headers are stored in a PPM or PPT marker segment this length does not include the packet header length. One value for each packet in the tile.

Due to the limitations of the PLM marker segment, the LPJE profile recommends the use of the PLT marker segment. The PLM marker segment is allowed within LPJE and may be preferred in some instances (*e.g.* only one tile in the compressed file). The NPJE profile requires a PLT marker segment for each layer in the codestream. This is unnecessary, all of the lengths could be combined into one single PLT, and is not required (but it is allowed) within LPJE. Additionally, the EPJE profile required a PLT per tile-part describing the packet lengths within each tile-part. Again, this is an unnecessary complication and while it is allowed in LPJE, it is not a requirement. JPEG 2000 implementations should be sophisticated enough to properly sort the length information contained within PLT marker segments and associate it with the correct packets. ISO/IEC 15444-1 allows use of both PLM and PLT in the same codestream, this is not recommend in the LPJE profile.

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

If the PPM or PPT marker segments are used to relocate the packet headers, then the PLM and PLT marker segments packet length information describes only the packet bodies. It does not include the packet headers in the lengths. The presence of PPM and PPT marker segments also influences the behavior of the SOP and EPH marker segments (see below). Table E-25 gives the PPM marker segment. The PPM marker segment aggregates packet header information across all tile-parts into the main header. This allows a decoder to quickly decode all packet header information at once, but it comes at a price.

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

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

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

The PPT marker segment is given in Table E-26. It performs a similar function to the PPM marker segment, but it places the packet header information in tile-part headers instead. PPT marker segments need not appear in the first tile-part header, they need only appear in a tile-part header that is located in the compressed file before the packet data that the packet headers describe. The PPT marker segment is a compromise between placing all of the packet headers in the main header and letting them be distributed through out the bitstream.

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

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

The LPJE profile allows usage of the PPM and PPT marker segments. The PPT marker segment is preferred since it does not front load the codestream too much. In cases where there is only one tile in the image, the PPM marker segment is as efficient as the PPT marker segment. If either marker segment is used, then the packet headers may not appear in the bitstream data. Furthermore, if the PPM marker segment is used, the PPT marker segment may not be used. The converse is true as well.

E.8.9 In bitstream Marker and Marker Segment

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

Table E-27 Start of packet (15444-1 Annex A.8.1)

Parameter	Size (bits)	Values	LPJE	Notes
SOP	16	0xFF91	0xFF91	Start of packet marker
Lsop	16	4	4	Length of marker segment
Nsop	16	0 – 65,535	0 – 65,535	Packet sequence number

Table E-27 shows the SOP marker segment. Nsop is incremented for each packet and if it goes beyond 65,535, the count is reset back to 0 and started again. If PPM or PPT marker segments are used to move the packet headers, the SOP marker segment does not move. Instead it is placed in front of the packet body rather than in front of the packet header.

Table E-28 End of packet header (15444-1 Annex A.8.2)

Parameter	Size (bits)	Values	LPJE	Notes
EPH	16	0xFF92	0xFF92	End of packet header marker

Table E-28 gives the EPH marker. The EPH marker is placed in the bitstream right after the packet header and before the packet body. If the EPH marker is used, it must appear for each packet. If a packet header is corrupted, the EPH marker allows some measure of recovery by delineating the end of the packet header. This prevents the decoding of the packet header from becoming confused and interpreting the packet body as part of the packet header. If PPM or PPT marker segments are used to relocate the packet headers, then the EPH marker segments are moved along with the packet headers.

The NPJE and EPJE profiles do not recommend the use of the SOP and EPH markers. The LPJE profile allows their use and encourages it in situations where noisy transmission might corrupt the codestream and there are no other system mechanisms in place to handle bit errors. Some overhead price is paid for using SOP and EPH and implementers must consider this in their system design. Trying to recover from a bit error in a JPEG 2000 codestream requires an in-depth understanding of the codestream structures, wavelets and entropy coding. It is reasonable to assume that different codec implementations will perform differently under similar environments. It is also important to realize that where a bit error occurs can have widely varying effects on the decoded image quality. The JPEG 2000 committee has developed more robust error correction and control procedures in ISO/IEC 15444-11:2007, Information technology -- JPEG 2000 image coding system: Wireless. A future version of the LPJE profile may make include this as an option.

E.8.10 Informational Marker Segments

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

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

Parameter	Size (bits)	Values	LPJE	Notes
CRG	16	0xFF63	0xFF63	Component registration marker
Lcrg	16	6 – 65,534	6 – 65,534	Length of marker segment
$Xcrg^i$	16	0 – 65,535	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	0 – 65,535	Value of vertical offset in units of 1/65536 of the vertical separation $YRsiz^i$, for the i^{th} component

Table E-30 Comment (15444-1 Annex A.9.2)

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