INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.804

(08/2002)

SERIES T: TERMINALS FOR TELEMATIC SERVICES

Information technology – JPEG 2000 image coding system : PART 5 – Reference software

CAUTION !

PREPUBLISHED RECOMMENDATION

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU [had/had not] received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Foreword

Foreword to be supplied by ISO and ITU.

Introduction

Introduction to be supplied by ISO and ITU.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY - JPEG 2000 IMAGE CODING SYSTEM : PART 5 - REFERENCE SOFTWARE

1 Scope

ITU-T Rec. T.800 | ISO/IEC 15444-1 defines a set of lossless and lossy compression methods for coding continuous-tone, bi-level, greyscale or colour digital still images. This Recommendation | International Standard provides two independently created software reference implementations of ITU-T Rec. T.800 | ISO/IEC 15444-1, in order to assist implementers of ITU-T Rec. T.800 | ISO/IEC 15444-1 in testing and understanding its content.

The reference software is informative only. This Recommendation | International Standard does not define any additional part of the JPEG 2000 image coding system.

Each version of the reference software contains source code, which may be compiled to provide the following functionality:

- transcoding from selected, widely available image formats into a JPEG 2000 codestream
- transcoding from selected, widely available image formats into the JP2 file format
- selection of a wide range of JPEG 2000 encoding options (as documented in each reference software)
- decoding from a JPEG 2000 codestream to a range of selected widely available image formats
- partial processing of a JP2 file to extract a JPEG 2000 codestream for decoding to a range of selected widely available image formats

  NOTE — The reference software does not implement a conforming JP2 file reader. See section 9.
- some additional tools to help with evaluation and testing

The reference software is intended for use as a testing and validation tool for other implementations of JPEG 2000, and to help in the understanding of ITU-T Rec. T.800 | ISO/IEC 15444-1. Although components of the reference software may find application in software intended for product development, this was not an objective of the development of this software, and prospective implementors are cautioned against making any estimations of performance or resource usage based on the reference software.

2 Normative references

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation T.800 | ISO/IEC 15444-1:2000 *Information Technology - JPEG 2000 Image Coding System*

2.2 Additional references

- ISO/IEC 9899:1999: Programming languages; C
- ISO/IEC 9945-1:1995: Information technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) (C language)
- ISO/IEC 9945-2::1995: Information technology - Portable Operating System Interface (POSIX) - Part 2: Shell and utilities

3 Definitions

3.1 **big endian**: The bits of a value representation occur in order from most significant to least significant.

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

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

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

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

3.6 **box contents**: Refers to the data wrapped within the box structure. The contents of a particular box are stored within the DBox field within the Box data structure.

3.7 **byte**: Eight bits.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3.23 **image area**: A rectangular part of the reference grid, registered by offsets from the origin and the extent of the reference grid.

3.24 **image area offset**: The number of reference grid points down and to the right of the reference grid origin where the origin of the image area can be found.

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

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

3.27 **JP2:** The name of the file format defined by ITU-T Recommendation T.800 | ISO/IEC 15444-1:2000.

3.28 **JPEG**: Used to refer globally to the encoding and decoding process of the following Recommendations | International Standards:

— ITU-T Recommendation T.81 | ISO/IEC 10918-1:1994, Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines,

— ITU-T Recommendation T.83 | ISO/IEC 10918-2:1995, Information technology - Digital compression and coding of continuous-tone still images: Compliance testing,

— ITU-T Recommendation T.84 | ISO/IEC 10918-3:1996, Information technology - Digital compression and coding of continuous-tone still images: Extensions,

— ITU-T Recommendation T.84 | ISO/IEC 10918-3 AMD 1 (In preparation), Information technology - Digital compression and coding of continuous-tone still images: Extensions - Amendment 1,

— ITU-T Recommendation T.86 | ISO/IEC 10918-4, Information technology - Digital compression and coding of continuous-tone still images: Registration of JPEG Profiles, SPIFF Profiles, SPIFF Tags, SPIFF colour Spaces, APPn Markers, SPIFF, Compression types and Registration authorities (REGAUT).

3.29 **JPEG 2000**: Used to refer globally to the encoding and decoding processes in this Recommendation | International Standard and their embodiment in applications.

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

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

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

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

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

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

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

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

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

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

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

3.41 **raster order**: A particular sequential order of data of any type within an array. The raster order starts with the top left data point and moves to the immediate right data point, and so on, to the end of the row. After the end of the row is reached the next data point in the sequence is the left-most data point immediately below the current row. This order is continued to the end of the array.

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

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

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

3.45 **reference tile**: A rectangular sub-grid of any size associated with the reference grid.

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

3.47 **resolution level**: Equivalent to decomposition level with one exception: the LL subband is also a separate resolution level.

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

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

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

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

3.52 **subband coefficient**: A transform coefficient within a given subband.

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

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

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

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

4 Abbreviations and symbols

4.1 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply.

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

ICC: International Colour Consortium

ICT: Irreversible Colour transformation

IEC: International Electrotechnical Commission

ISO: International Organization for Standardization

ITTF: Information Technology Task Force

ITU: International Telecommunication Union

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

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

JURA: JPEG Utilities Registration Authority

1D-DWT: One-dimensional Discrete Wavelet Transformation

FDWT: Forward Discrete Wavelet Transformation

IDWT: Inverse Discrete Wavelet Transformation

LSB: Least Significant Bit.

MSB: Most Significant Bit.

PCS: Profile Connection Space

RCT: Reversible Colour Transformation

ROI: Region Of Interest

SNR: Signal to Noise Ratio.

UCS: Universal Character Set

URI: Uniform Resource Identifier

URL: Uniform Resource Location

UTF-8: UCS Transformation Format 8

UUID: Universal Unique Identifier

XML: Extensible Markup Language

W3C: World-Wide Web Consortium

4.2 Symbols

For the purposes of this Recommendation | International Standard, the following symbols apply.

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

nnn: A three-digit number preceded by a backslash indicates the value of a single byte within a character string, where the three digits specify the octal value of that byte.

COC: Coding style component marker

COD: Coding style default marker

COM: Comment marker

CRG: Component registration marker

EPH: End of packet header marker

EOC: End of codestream marker

PLM: Packet length, main header marker

PLT: Packet length, tile-part header marker

POC: Progression order change marker

PPM: Packed packet headers, main header marker

PPT: Packed packet headers, tile-part header marker

QCC: Quantization component marker

QCD: Quantization default marker

RGN: Region of interest marker

SIZ: Image and tile size marker

SOC: Start of codestream marker

SOP: Start of packet marker

SOD: Start of data marker

SOT: Start of tile-part marker

TLM: Tile-part lengths marker

5 Conventions

The source files provided are supplied in the form of an individual zip file for each source tree. File locations given in this document are expressed relative to the top level of the corresponding source tree. A Unix style file structure and delimiters are assumed.

Basic instructions are provided within the reference software for the installation and compilation of the sources under a variety of operating systems and platforms. No support can be provided by ISO | ITU-T beyond that offered in this document and through links on the official JPEG web site, http://www.jpeg.org.

6 General description

Two independent and separate software source trees are provided. These are:

- **JASPER.ZIP**, provided as indicated in the file **COPYRIGHT**, contained within the zip file. This is written in the C programming language, and should compile and run on any platform with a C language implementation conforming to ISO/IEC 9899:1999, and supporting a subset of the POSIX C API, ISO/IEC 9945-1:1990.
- **JJ2000.ZIP**, provided as indicated in the file **COPYRIGHT**, contained within the zip file. This is written in the Java™ programming language, and executes under versions of the Java Virtual Machine (JVM) from version 1.1.1 onwards.

The supplied executables are described briefly in section 8, and in more detail with some information about the supplied source code in annex A (Jasper) and annex B (JJ2000).

Both distributions have been tested as meeting the coding and decoding requirements for codestreams identified in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000. Formal compliance is beyond the scope of this Recommendation | International Standard.

7 Copyright and licensing

These software modules were originally developed by the parties indicated in the file **COPYRIGHT** within each package forming a part of this Recommendation | International Standard, in the course of development of ITU-T Rec. T.800 | ISO/IEC 15444-1:2000. These software modules are separate and discrete implementations of ITU-T Rec. T.800 | ISO/IEC 15444-1:2000. ISO/IEC draws the attention of the users of these software modules to the license terms and conditions specified in the file **LICENSE** in each implementation. Those intending to use these software modules in hardware or software products are advised that their use may infringe existing patents. In particular, the original developers of these software modules and their respective companies, the editors and their companies, and ISO/IEC have disclaimed liability for any proposed use of these software modules or modifications thereof. No licensing is implied for their use in whole or in part in products that fail to conform to ITU-T Rec. T.800 | ISO/IEC 15444-1:2000. The original software authors retain full rights to use the code within this Recommendation | International Standard for their own purposes, to assign or donate the code to a third party and to inhibit third parties from using the code for products that fail to conform to ITU-T Rec. T.800 | ISO/IEC 15444-1:2000.

The two distributions have differing copyright and licensing restrictions, which reflect the different requirements and operating environments of those organisations that have contributed to the development of the software.

8 Platform requirements

Both reference software implementations have been successfully built on a variety of operating platforms and with a selection of compilers. They have been written with portability and comprehensibility in mind. Platforms for which there is specific installation documentation are indicated below:

8.1 Jasper requirements

Jasper has been successfully compiled in the following environments:

- Red Hat Linux 7.0, GNU C 2.96, GNU Make 3.79.1
- SunOS/SPARC 5.5.1, GNU C 2.7.2.1, SunOS make
- SunOS/SPARC 5.7, GNU C 2.95, SunOS make
- Windows 2000 Professional, Microsoft Visual C 6.0
- Windows 98 Second Edition, Cygwin 1.1.8.2, GNU Bash 2.04, GNU C 2.95, GNU Make 3.79.1

8.2 JJ2000 requirements

JJ2000 has been tested on versions of the Java™ Virtual Machine from version 1.1.1 and later, on a variety of platforms including Sun Solaris, Microsoft Windows 95/98/NT/2000, Linux, and MacOS, using Sun's Java Development Kit (JDK), Sun's Java Runtime Environment (JRE) and Microsoft's Java Virtual Machine (JVM). Detailed instructions on how to set up and run the JJ2000 executables are included in the file **README** in the top level of the JJ2000 source tree.

9 Reference code structure

Both sets of reference software offer the capability to encode and decode codestreams that conform to the syntax defined in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000.

Both sets of reference software include a partial implementation of the JP2 file format defined in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000. However, neither implementation is a conforming JP2 file reader as defined in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000. Several features are missing from the reference software that are required of a conforming file reader. These include support for palettized colour, component mapping and Restricted ICC Profiles.

For transcoding to other image formats or for display, both sets of reference software allow the user to assume that the decoded codestream components are in an RGB colour space or greyscale. In general, additional information is required in order to determine the true nature of the component data. Support for such information is provided by the JP2 file format but is not processed by the reference software.

The executable files described below are available by compiling the reference software as indicated in the **INSTALL** file held in the top level of each respective source tree. In both cases, more than 30 optional different command line parameters may be defined, which can demonstrate aspects of the encoding for JPEG 2000 codestreams defined in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000.

9.1 Jasper executables

The Jasper reference software distribution provides three executables:

- **JASPER**, the transcoder which can be used to convert to and from a variety of image file formats including JPEG, Portable BitMap (PNM/PGM/PPM), Windows BMP, and Sun Rasterfile. The transcoder acts as both encoder and decoder for JPEG 2000 files.

  NOTE — JASPER is not a conforming JP2 file reader. See the beginning of section 9.

- **IMGCMP**, provided as a test utility to measure differences between images and provide various measures of the comparative differences, such as SNR.

- **IMGINFO**, provided as a simple command line utility to analyse JPEG 2000 files.

9.2 JJ2000 executables

The JJ2000 software distributions provides two executables:

- **JJ2KEncoder**, which acts as an encoder from PGM, PPM and PGX to JPEG 2000 (both codestream and JP2 file format).

  NOTE — PGX is a non-standard format, defined in the JJ2000 documentation, that (unlike PGM) supports arbitrary bit-depths and signed sample values.

- **JJ2KDecoder**, which takes a JPEG 2000 codestream or JP2 file and decodes it to PGM, PPM, or PGX format. It can also take a number of optional parameters that emulate some of the partial decoding features that might be anticipated in a client server environment with restricted bandwidth communications. It is also capable of rendering the image to a screen display if no output file specification is provided, offering simple viewing capabilities for JPEG 2000 codestreams and JP2 files.

  NOTE — JJ2KDecoder is not a conforming JP2 file reader. See the beginning of section 9.

10 Intellectual Property

There is the possibility that, for some of the processes specified in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000, conformance or compliance may require use of an invention covered by patent rights. By publication of this Recommendation | International Standard, no position is taken with respect to the validity of these claims or of any patent rights in connection therewith. Information regarding such patents can be obtained from the organizations identified in Table L.1 in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000, which summarizes the formal patent and intellectual property rights statements that have been received in connection with the processes defined in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000.

11 Software availability and updates

The reference software sources released with this Recommendation | International Standard are the latest tested versions available at the date on which the text was released for final approval. Later versions of the software and implementation or fault reports and fixes may be made available after the publication of this Recommendation | International Standard. These may be found at the URLs contained in the documentation for each version of the reference software, or through links which may be maintained at **http://www.jpeg.org/software.**

Annex A
JASPER - C reference software- software description

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

A.1 Introduction

JasPer is a collection of software (i.e., a library and application programs) for the coding and manipulation of images. This software is written in the C programming language. Of particular relevance to this standard, the JasPer software provides an implementation of the image codec specified in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000. Support for several other popular image codecs are also included in order to facilitate the transcoding of image data to and from JPEG 2000 codestreams and the JP2 file format. The JasPer software was developed with the objective of providing a license and royalty fee free JPEG 2000 implementation to anyone wishing to use the JPEG 2000 standard, although no warranty is provided as to any potential infringement of intellectual property (see the document **LICENSE** included with the software). Users of the software are advised to satisfy themselves as to the validity of any claims made for intellectual property which are made against ITU-T Rec. T.800 | ISO/IEC 15444-1:2000 before using this software.

NOTE — JASPER is not a conforming JP2 file reader. See the beginning of section 9.

In addition to this Annex, readers interested in the JasPer software should also read the JasPer Software Reference Manual, which is included in the JasPer software distribution archive. This manual contains detailed information about the JasPer software, and includes release-specific details that have been deliberately omitted from this Annex.

A.2 Software Updates

The JasPer software release accompanying this standard represents the most recent version available at the time this document was published. The development of the JasPer software is expected to continue. As new features and functionality are added, new releases of the software may become available. The most recent version of the JasPer software can be downloaded from the Internet via the URL given in section 11 of this Recommendation | International Standard. Further information and news about the JasPer software may also be made available via the same URL.

A.3 Version Numbering

As the JasPer software is expected to evolve over time, it is important to be able to identify particular releases of the software. Each release of the JasPer software has a version identifier. For the purposes of this Annex, a version identifier is comprised of three integers separated by dots. In order, the three integers correspond to the major, minor, and micro version numbers for the software. For example, the version identifier "1.500.0" corresponds to a major version of 1, a minor version of 500, and a micro version of 0. In instances where the micro version is zero, the version identifier may be truncated after the minor version number. For example, the version identifier "1.500" is completely valid and an abbreviation for "1.500.0".

Given two different releases of the JasPer software, the most recent can be identified by comparing the version identifiers, as follows: 1) if the major version numbers differ, the release with the higher major version number is newer; 2) if the major version numbers are equal and the minor version numbers differ, the release with the higher minor version number is newer; or 3) if the major version numbers are equal and the minor version numbers are equal, the release with the higher micro version is newer.

A.4 Software Overview

The JasPer software consists of a library, and several demonstration application programs that utilize this library. The code is written in the C programming language in conformance with ISO/IEC 9899:1999. This language was chosen due to the availability of C development environments for most of today's computing platforms. In total, the software consists of about 30K lines of code.

In the sections that follow, a brief overview of the JasPer software is provided. No mention is made of details that are likely to change between software releases. These are described within documents included in the relevant software distribution, to avoid introducing inconsistencies between this Recommendation | International Standard and the documentation accompanying the relevant JasPer software release.

A.5 JasPer Library

At the core of the JasPer software is the JasPer library, named libjasper. The library consists of two distinct categories of code: 1) base/core code, and 2) codec drivers. The base code provides generic routines for manipulating images and provides a framework for constructing codec drivers. The codec drivers provide a means for encoding/decoding images in specific formats. The library has been designed to be extensible, so that adding support for new image formats should be straightforward.

Currently, codec drivers are provided for the following image formats:

1) Partial support (non-conforming) for JPEG 2000 JP2 file (JP2),

2) JPEG 2000 code stream (JPC),

3) JPEG JFIF (JPG),

4) Portable Bitmap (PNM),

5) Windows BMP (BMP), and

6) Sun Rasterfile (RAS).

In addition, support is also provided for a number of non-standard formats, which offer features that are not available through any of the above formats (for example unusual image geometries). These formats are documented in the JasPer Software Reference Manual. The JPG codec driver requires that the IJG JPEG library be available on the system used to compile the software. The IJG JPEG library is free software and can be obtained from the Internet via the URL:

http://www.ijg.org

For licensing reasons, the IJG JPEG library is not distributed with the JasPer software.

The JP2 and JPC codecs provided are the main element of relevance to this Recommendation | International Standard. The JPG codec within JasPer provides support for the original JPEG format, and can be used to convert JPEG-formatted data into JPEG 2000.

The JasPer library provides the following key classes:

1) An image class. This class is used to represent an image, and also provides access to codec drivers for encoding/decoding image data in various formats.

2) A sequence/matrix class. This class provides matrix and two-dimensional sequence classes.

3) A I/O stream class. This class provides I/O streams similar to that of the standard C library, but with additional functionality required by other code in JasPer.

4) A fixed-point number class. This (templated) class is used for performing fixed-point arithmetic.

5) A tag-value parser class. This class facilitates the parsing of tag-value pairs. A tag-value pair is a string of the form "tag=value". Such pairs are used by some interfaces within JasPer in order to pass parameters.

6) A command line option class. This class allows the parsing of command lines. This code is similar to the getopt function available on most UNIX systems.

A.6 JasPer Demo Application Programs

Three demonstration application programs are included that utilize the JasPer library. These application programs are:

1) **jasper**. This is a simple transcoder application. It converts image data from one format to another.

2) **imgcmp**. This is a image comparison utility. It can be used to quantify the difference between two images (using various distortion metrics).

3) **imginfo.** This program outputs basic information about the type and geometry of an image.

The **jasper** program is the main reference software provided and can be used as a JPEG 2000 encoder and/or decoder. The **imgcmp** utility is useful for analyzing image coding performance.

A.7 Software Requirements

The JasPer code is intended to compile on any platform with a C language implementation conforming to ISO/IEC 9899:1999 and supporting a subset of ISO/IEC 9945-1:1995 (i.e., the POSIX C API). Only limited POSIX support is required (i.e., the open, close, read, write, and lseek functions must be supported).

The GNU C compiler is suggested for building the JasPer software as this compiler is compliant for the purposes of this reference software with ISO/IEC 9899:1999, and produces good quality code. This compiler is available for systems based on POSIX/UNIX and Microsoft Windows, amongst others. More information about the GNU C compiler can be found on the Internet at:

http://www.gnu.org

Portability was a major consideration during the design of the JasPer software. For this reason, the software makes minimal assumptions about the runtime environment. The code uses very little floating-point arithmetic, most of which can be attributed to floating-point conversions in printf's. This minimal use of floating-point arithmetic should make the code easier to port to platforms lacking hardware support for floating-point arithmetic.

A.8 Building the Software

The JasPer software can be built using one of two different methods. The first method is based on a configure script (produced using the well known autoconf tool). This method should work in most mainstream UNIX- or POSIX-like environments (e.g., Linux, Solaris, and the Cygwin environment under Microsoft Windows). This is the preferred and recommended method for building JasPer. The second method is specifically tailored to the needs of Microsoft Visual C users under Microsoft Windows. Detailed instructions for each of these two build methods can be found in the JasPer Software Reference Manual. (See the section titled "Building the Software".)

A.9 Using the Software

As described earlier, the JasPer software consists of a library and several application programs. All of the application programs have a similar UNIX-style command line interface. The specifics of the command line interface for each application program are given in the JasPer Software Reference Manual. (See the section titled "Using the Software".) It is also possible to use the JasPer library to build new application programs (or even other libraries). Information on how to do this can also be found in the same section in the JasPer Software Reference Manual.

.

Annex B
JJ2000 - Java™ reference software - software description

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

B.1 Introduction

JJ2000 is a Java™ implementation of JPEG 2000 as defined in ITU-T Rec. T.800 | ISO/IEC 15444-1:2000. The software consists of an encoder and a decoder. The encoder compresses image files from a number of image formats to a JPEG 2000 Codestream or a JP2 file (see section 9.2). The encoder provides an implementation of the image encoder specified in ISO/IEC 15444-1 and supports a number of additional features such as paralellization and running the encoder as an Java™ applet. The decoder provides decompression of a JPEG 2000 codestream into a number of image formats, and partial support (non-conforming) for the JP2 file format (see section 9.2). It can also take a number of optional parameters that emulate some of the partial decoding features that might be anticipated in a client server environment with restricted bandwidth communications. It is also capable of rendering the image to a screen display if no output file specification is provided, offering simple viewing capabilities for JPEG 2000 codestreams and JP2 files.

NOTE — JJ2KDecoder is not a conforming JP2 file reader. See the beginning of section 9.

How to install and use the encoder and the decoder is described in the INSTALL document provided in the software distribution

B.2 Software Updates

The latest version of the JJ2000 software, as of the time this document was published, can be found in the software release accompanying this standard.. The most recent version of the JJ2000 software can be downloaded from the Internet via the URL given in Section 11. Additional information about the implementation, documentation, byte-code and executables can also be found on the same URL.

B.3 Software Architecture

The JJ2000 software consists of several Java hierarchical packages, each one corresponding to a specific module of the JJ2000 encoder-decoder. Each of these packages correspond to a directory with the same name within the zip-file containing the software distribution accompanying this document.

jj2000	
jj2000.disp	Display of decoded images
jj2000.j2k	JPEG 2000 modules
jj2000.j2k.io	Writing to / Reading from files
jj2000.j2k.roi	Region Of Interest support
jj2000.j2k.util	Useful tools for JJ2000 implementation
jj2000.j2k.codestream	
jj2000.j2k.codestream.writer	Writing of the code-stream
jj2000.j2k.codestream.reader	Reading of the code-stream
jj2000.j2k.decoder	Decoder module
jj2000.j2k.encoder	Encoder module
jj2000.j2k.entropy	Arithmetic encoding/decoding
jj2000.j2k.entropy.decoder	MQ and entropy decoders
jj2000.j2k.entropy.encoder	MQ, entropy encoder, EBCOT
jj2000.j2k.wavelet	Filters and filtering
jj2000.j2k.wavelet.analysis	Forward wavelet decomposition
jj2000.j2k.wavelet.synthesis	Inverse wavelet decomposition
jj2000.j2k.image	Spatial domain operations
jj2000.j2k.image.forwcompTransf	Forward Component transformation
jj2000.j2k.image.invcomptransf	Inverse Component transformation

jj2000.j2k.image.output	Writing of PGM/PPM/PGX files
jj2000.j2k.image.input	Reading of PGM/PPM/PGX files
jj2000.j2k.quantization	
jj2000.j2k.quantization.quantizer	Dead-zone Scalar quantizer
jj2000.j2k.quamtization.dequantizer	Dead-zone Scalar Dequantizer

A more thorough description of the design and classes of JJ2000 can be found in the README file provided in the software distribution. For more information about encoder/decoder, please refer to the embedded command-line help and the source-code documentation.

B.4 Installing and Running the Software

JJ2000 is written in Java™ and to run the software a Java™ compiler and a Java Virtual Machine is needed. Instructions of how to compile and run the software on different platforms is given in the INSTALL document provided in the software distribution. The same document also describes how to generate documentation from the source code and how to use some of the additional features of the encoder.