

**JPEG File Interchange
Format (JFIF)**

Technical
Report



COPYRIGHT PROTECTED DOCUMENT

© Ecma International 2009

Contents

Page

1	Scope	1
1.1	Why a File Interchange Format	1
1.2	JPEG File Interchange Format features	1
2	Conformance	1
3	Normative references	2
4	Terms and definitions	2
5	Abbreviations	2
6	JPEG File Interchange Format (JFIF) Overview	2
6.1	JPEG Compression	2
6.2	Compatible across platforms	2
6.3	Standard color space	3
6.4	The APP0 marker is used to identify a JPEG FIF file	3
6.5	APP0 marker used to specify JFIF extensions	3
6.6	APP0 marker used for application-specific information	3
7	Conversion to and from RGB	3
8	Image Orientation	4
9	Spatial Relationship of Components	4
10	JPEG File Interchange Format Specification	5
10.1	JFIF Extension APP0 Marker Segment	6
10.2	JFIF Extension: Thumbnail coded using JPEG	7
10.3	JFIF Extension: Thumbnail stored using one byte per pixel	7
10.4	JFIF Extension: Thumbnail stored using three bytes per pixel	8
10.4.1	Useful tips	8

Introduction

This document specifies a file format, referred to as the JPEG File Interchange Format (JFIF), for file-based interchange of images encoded according to the JPEG standard (ITU-T Recommendation T.81 | ISO/IEC 10918-1).

The JPEG File Interchange Format (JFIF) was collaboratively developed by a group of computer, telecommunications, and imaging companies in the early 1990's. Representing C-Cube Microsystems, Eric Hamilton led the development of the specification. He hosted a meeting in late 1991 toward developing a simple file format based on JPEG which would allow for the interchange of files containing JPEG bitstreams between platforms and applications. There were about 40 representatives from various computer, telecommunications, and imaging companies at the meeting. Subsequent specification development work was conducted using e-mail and telephone discussions. The effort reached consensus fairly quickly and led to publication of version 1.0 of the JFIF specification, which had been edited by Eric Hamilton and which he distributed to the participants and to other interested parties.

Shortly thereafter, the group came to the conclusion that the spatial sampling relationship of components specified in JFIF 1.0 was not ideal since it followed digital video conventions rather than those used in common computer formats such as Postscript and QuickTime. They chose to publish another version, JFIF 1.01, which changed that part of the specification to follow the computer format convention. This was deemed to be a minor change since JFIF 1.0 had been circulated for only a short while and decoders which ignored the version number would still render similar images. The Independent JPEG Group (IJG) adopted JFIF version 1.01 for use in its public domain software, which eventually led to millions of images being published in this format. Later in 1992, user feedback led to the final version of JFIF, version 1.02, which supported additional thumbnail formats – most importantly including thumbnails stored in compressed format.

The JFIF Version 1.02 specification became available in a *de facto* manner in the public domain, and it has been implemented widely, to the extent that it became widely recognized as a *de facto* standard.

The desire to convert JFIF Version 1.02 to a formal publication status has been a topic of various discussions since its development, but no action was taken on that subject until preparation of this specification. The rationale for this Ecma International Technical Report is therefore to formally document JFIF Version 1.02 in a technically identical Ecma International Technical Report.

By preparing this specification, Ecma International acknowledges and applauds the excellent work that Eric Hamilton and the JFIF group did when creating the JFIF specification, and expresses its gratitude to the supporting organizations, including in particular C-Cube Microsystems (now LSI Logic) and the other companies that have participated in the original JFIF work.

Ecma International also expresses its sincere gratitude to Eric Hamilton for his valuable help in editing this specification.

This Ecma Technical Report has been adopted by the General Assembly of June 2009.

JPEG File Interchange Format (JFIF)

1 Scope

1.1 Why a File Interchange Format

JPEG File Interchange Format (JFIF) is a minimal file format which enables JPEG bitstreams (ITU-T Recommendation T.81 | ISO/IEC 10918-1) to be exchanged between a wide variety of platforms and applications. This minimal format does not include any of the advanced features found in the TIFF specification or any application-specific file format. The only purpose of this simplified format is to allow the exchange of JPEG compressed images.

1.2 JPEG File Interchange Format features

- Uses JPEG compression
- Uses JPEG interchange format compressed image representation
- PC or Mac or Unix workstation compatible
- Standard color space: one or three components. For three components, YCbCr (ITU-R BT.601, 256 levels)
- APP0 marker used to specify units, X pixel density, Y pixel density, thumbnail
- APP0 marker also used to specify JFIF extensions
- APP0 marker also used to specify application-specific information.

2 Conformance

Many requirements in this specification are expressed as format or syntax requirements rather than software or hardware implementation requirements. Implementations fall into two categories: JFIF decoders and JFIF encoders.

In order for a JFIF decoder to be considered conforming, the following rules apply:

- It **MUST NOT** report errors when processing conforming instances of the documented format except when forced to do so by resource exhaustion.
- It **SHOULD** report errors when processing non-conforming instances of the documented formats when doing so does not pose an undue processing or performance burden.

In order for a JFIF encoder to be considered conforming, the following rules apply:

- It **MUST NOT** generate any new, non-conforming instances of the documented format.
- It **MUST NOT** introduce any non-conformance when modifying an instance of the documented format.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ITU-T Recommendation T.81 | ISO/IEC 10918-1: 1992, *Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines*

ITU-R Recommendation BT.601-6 (01.07) *Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios* (Former CCIR 601 Recommendation).

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1

JPEG File Interchange Format

JFIF

minimal file format which enables JPEG bitstreams (ITU-T Recommendation T.81 | ISO/IEC 10918-1) to be exchanged between a wide variety of platforms and applications

4.2

Thumbnail

miniature (reduced resolution) representation of the JPEG image that is used to identify the image by its contents.

NOTE Thumbnails are commonly used to quickly browse multiple images using low resolution visual representation of the images rather than using file names or other metadata.

5 Abbreviations

JFIF JPEG File Interchange Format

6 JPEG File Interchange Format (JFIF) Overview

6.1 JPEG Compression

Although any JPEG process is supported by the syntax of the JPEG File Interchange Format (JFIF) it is strongly recommended that the JPEG baseline process – as defined in ITU-T Recommendation T.81 | ISO/IEC 10918-1 - be used for the purposes of file interchange. This ensures maximum compatibility with all applications supporting JPEG. JFIF conforms to the JPEG International Standard (ITU-T Recommendation T.81 | ISO/IEC 10918-1).

The JPEG File Interchange Format is entirely compatible with the standard JPEG interchange format; the only additional requirement is the mandatory presence of the APP0 marker right after the SOI marker. Note that JPEG interchange format requires (as does JFIF) that all table specifications used in the encoding process be coded in the bitstream prior to their use.

6.2 Compatible across platforms

The JPEG File Interchange Format is compatible across platforms: for example, it does not use any resource forks, supported by the Macintosh but not by PCs or workstations.

6.3 Standard color space

The color space to be used is YCbCr as defined by ITU-R BT.601 (256 levels). The RGB components calculated by linear conversion from YCbCr shall not be gamma corrected (gamma = 1.0). If only one component is used, that component shall be Y. APP0 marker used to identify JPEG FIF.

6.4 The APP0 marker is used to identify a JPEG FIF file

The JPEG FIF APP0 marker is mandatory right after the SOI marker. The JFIF APP0 marker is identified by a zero terminated string: "JFIF". The APP0 can be used for any other purpose by the application provided it can be distinguished from the JFIF APP0. The JFIF APP0 marker provides information which is missing from the JPEG stream: version number, X and Y pixel density (dots per inch or dots per cm), pixel aspect ratio (derived from X and Y pixel density), thumbnail.

6.5 APP0 marker used to specify JFIF extensions

Additional APP0 marker segment(s) can optionally be used to specify JFIF extensions. If used, these segment(s) must immediately follow the JFIF APP0 marker. Decoders should skip any unsupported JFIF extension segments and continue decoding. The JFIF extension APP0 marker is identified by a zero terminated string: "JFXX". The JFIF extension APP0 marker segment contains a 1-byte code which identifies the extension. This version, version 1.02, has only one extension defined: an extension for defining thumbnails stored in formats other than 24-bit RGB.

6.6 APP0 marker used for application-specific information

Additional APP0 marker segments can be used to hold application-specific information which does not affect the decodability or displayability of the JFIF file. Application-specific APP0 marker segments must appear after the JFIF APP0 and any JFXX APP0 segments. Decoders should skip any unrecognized application-specific APP0 segments. Application-specific APP0 marker segments are identified by a zero terminated string which identifies the application (not "JFIF" or "JFXX"). This string should be an organization name or company trademark. Generic strings such as dog, cat, tree, etc. should not be used.

7 Conversion to and from RGB

Y, Cb, and Cr are converted from R, G, and B as defined in ITU-R Recommendation BT.601, but are normalized so as to occupy the full 256 levels of a 8-bit binary encoding. More precisely:

$$Y = 256 * E'y$$

$$Cb = 256 * [E'Cb] + 128$$

$$Cr = 256 * [E'Cr] + 128$$

where the E'y, E'Cb and E'Cb are defined as in ITU-R BT.601. Since values of E'y have a range of 0 to 1.0 and those for E'Cb and E'Cr have a range of -0.5 to +0.5, Y, Cb, and Cr must be clamped to 255 when they are maximum value.

RGB to YCbCr Conversion

YCbCr (256 levels) can be computed directly from 8-bit RGB as follows:

$$Y = 0.299 R + 0.587 G + 0.114 B$$

$$Cb = -0.1687 R - 0.3313 G + 0.5 B + 128$$

$$Cr = 0.5 R - 0.4187 G - 0.0813 B + 128$$

NOTE Not all image file formats store image samples in the order R0, G0, B0, ... Rn, Gn, Bn. Be sure to verify the sample order before converting an RGB file to JFIF.

YCbCr to RGB Conversion

RGB can be computed directly from YCbCr (256 levels) as follows:

$$R = Y + 1.402 (Cr-128)$$

$$G = Y - 0.34414 (Cb-128) - 0.71414 (Cr-128)$$

$$B = Y + 1.772 (Cb-128)$$

8 Image Orientation

In JFIF files, the image orientation is always top-down. This means that the first image samples encoded in a JFIF file are located in the upper left hand corner of the image and encoding proceeds from left to right and top to bottom. Top-down orientation is used for both the full resolution image and the thumbnail image. The process of converting an image file having bottom-up orientation to JFIF must include inverting the order of all image lines before JPEG encoding.

9 Spatial Relationship of Components

Specification of the spatial positioning of pixel samples within components relative to the samples of other components is necessary for proper image post processing and accurate image presentation. In JFIF files, the position of the pixels in sub sampled components is defined with respect to the highest resolution component. Since components must be sampled orthogonally (along rows and columns), the spatial position of the samples in a given sub sampled component may be determined by specifying the horizontal and vertical offsets of the first sample, i.e. the sample in the upper left corner, with respect to the highest resolution component.

The horizontal and vertical offsets of the first sample in a sub sampled component, X offset $i[0,0]$ and Y offset $i[0,0]$, is defined to be

$$X \text{ offset } i[0,0] = (N \text{ samples ref } / N \text{ samples } i) / 2 - 0.5$$

$$Y \text{ offset } i[0,0] = (N \text{ lines ref } / N \text{ lines } i) / 2 - 0.5$$

where

Nsamplesref is the number of samples per line in the largest component,

N samples i is the number of samples per line in the I th component,

N lines ref is the number of lines in the largest component,

N lines i is the number of lines in the I th component.

Proper sub sampling of components incorporates an anti-aliasing filter which reduces the spectral bandwidth of the full resolution components. Sub sampling can easily be accomplished using a symmetrical digital filter with an even number of taps (coefficients). A commonly used filter for 2:1 sub sampling utilizes two taps (1/2,1/2).

As an example, consider a 3 component image which is comprised of components having the following dimensions:

Component 1: 256 samples, 288 lines

Component 2: 128 samples, 144 lines

Component 3: 64 samples, 96 lines.

In a JFIF file, centers of the samples are positioned as illustrated below:

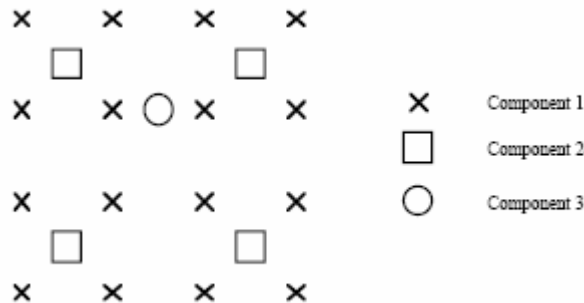


Figure 1 — Title

NOTE This definition is compatible with various industry specifications such as Postscript Level 2 and QuickTime. This definition is not compatible with the conventions used by ITU-R Recommendation 601 and some other digital video formats. For these formats, pre-processing of the chrominance components is necessary prior to compression in order to ensure accurate reconstruction of the compressed image.

10 JPEG File Interchange Format Specification

The syntax of a JFIF file conforms to the syntax for interchange format defined in Annex B of ITU-T Recommendation T.81 | ISO/IEC 10918-1. In addition, a JFIF file uses APP0 marker segments and constrains certain parameters in the frame header as defined below.

X'FF', SOI

X'FF', APP0, length, identifier, version, units, Xdensity, Ydensity, Xthumbnail, Ythumbnail, (RGB)n

Length	(2 bytes)	Total APP0 field byte count, including the byte count value (2 bytes), but excluding the APP0 marker itself
Identifier	(5 bytes)	= X'4A', X'46', X'49', X'46', X'00' This zero terminated string ("JFIF") uniquely identifies this APP0 marker. This string shall have zero parity (bit 7=0).
Version	(2 bytes)	= X'0102' The most significant byte is used for major revisions, the least significant byte for minor revisions. Version 1.02 is the current released revision.
Units	(1 byte)	Units for the X and Y densities. units = 0: no units, X and Y specify the pixel aspect ratio units = 1: X and Y are dots per inch units = 2: X and Y are dots per cm
Xdensity	(2 bytes)	Horizontal pixel density
Ydensity	(2 bytes)	Vertical pixel density
Xthumbnail	(1 byte)	Thumbnail horizontal pixel count

Ythumbnail	(1 byte)	Thumbnail vertical pixel count
(RGB)n	(3n bytes)	Packed (24-bit) RGB values for the thumbnail pixels, n = Xthumbnail * Ythumbnail

[Optional JFIF extension APP0 marker segment(s) - see below]

-
-
-

X'FF', SOFn, length, frame parameters

Number of components	Nf	= 1 or 3
1st component	C1	= 1 = Y component
2nd component	C2	= 2 = Cb component
3rd component	C3	= 3 = Cr component

-
-
-

X'FF', EOI

10.1 JFIF Extension APP0 Marker Segment

Immediately following the JFIF APP0 marker segment may be a JFIF extension APP0 marker. This JFIF extension APP0 marker segment may only be present for JFIF versions 1.02 and above. The syntax of the JFIF extension APP0 marker segment is:

X'FF', APP0, length, identifier, extension_code, extension_data

length	(2 bytes)	Total APP0 field byte count, including the byte count value (2 bytes), but excluding the APP0 marker itself
identifier	(5 bytes)	= X'4A', X'46', X'58', X'58', X'00' This zero terminated string ("JFXX") uniquely identifies this APP0 marker. This string shall have zero parity (bit 7=0).
extension_code	(1 byte)	= Code which identifies the extension. In this version, the following extensions are defined: = X'10' Thumbnail coded using JPEG = X'11' Thumbnail stored using 1 byte/pixel = X'13' Thumbnail stored using 3 bytes/pixel
extension_data	(variable)	= The specification of the remainder of the JFIF extension APP0 marker segment varies with the extension. See below for a specification of extension_data for each extension.

10.2 JFIF Extension: Thumbnail coded using JPEG

This extension supports thumbnails compressed using JPEG. The compressed thumbnail immediately follows the extension_code (X'10') in the extension_data field and the length of the compressed data must be included in the JFIF extension APP0 marker length field.

The syntax of the extension_data field conforms to the syntax for interchange format defined in Annex B of ITU-T Recommendation T.81 | ISO/IEC 10918-1. However, no “JFIF” or “JFXX” marker segments shall be present. As in the full resolution image of the JFIF file, the syntax of extension_data constrains parameters in the frame header as defined below:

X'FF', SOI

-
-
-

X'FF', SOFn, length, frame parameters

Number of components	Nf	= 1 or 3
1st component	C1	= 1 = Y component
2nd component	C2	= 2 = Cb component
3rd component	C3	= 3 = Cr component

-
-
-

X'FF', EOI

10.3 JFIF Extension: Thumbnail stored using one byte per pixel

This extension supports thumbnails stored using one byte per pixel and a color palette in the extension_data field. The syntax of extension_data is:

Xthumbnail	(1 byte)	Thumbnail horizontal pixel count
Ythumbnail	(1 byte)	Thumbnail vertical pixel count
Palette	(768 bytes)	24-bit RGB pixel values for the color palette. The RGB values define the colors represented by each value of an 8-bit binary encoding (0 - 255).
(pixel)n	(n bytes)	8-bit values for the thumbnail pixels n = Xthumbnail * Ythumbnail

10.4 JFIF Extension: Thumbnail stored using three bytes per pixel

This extension supports thumbnails stored using three bytes per pixel in the extension_datafield. The syntax of extension_data is:

Xthumbnail	(1 byte)	Thumbnail horizontal pixel count
Ythumbnail	(1 byte)	Thumbnail vertical pixel count
(RGB)n	(3n bytes)	Packed (24-bit) RGB values for the thumbnail pixels, $n = X_{\text{thumbnail}} * Y_{\text{thumbnail}}$

10.4.1 Useful tips

- you can identify a JFIF file by looking for the following sequence: X'FF', SOI, X'FF', APP0, <2 bytes to be skipped>, "JFIF", X'00'.
- if you use APP0 elsewhere, be sure not to have the strings "JFIF" or "JFXX" right after the APP0 marker.
- if you do not want to include a thumbnail, just program Xthumbnail = Ythumbnail = 0.
- be sure to check the version number in the special APP0 field. In general, if the major version number of the JFIF file matches that supported by the decoder, the file will be decodable.
- if you only want to specify a pixel aspect ratio, put 0 for the units field in the special APP0 field. Xdensity and Ydensity can then be programmed for the desired aspect ratio. Xdensity = 1, Ydensity = 1 will program a 1:1 aspect ratio. Xdensity and Ydensity should always be non-zero.

Bibliography

- [1] [JPEG File Interchange Format Specification v1.02](#), September 1, 1992.
- [2] [Supporting Data Compression in PostScript Level 2](#) and the filter Operator Technical Note #5115, 31 March 1992, Adobe Developer Support, PN LPS5115, Adobe Systems Incorporated
- [3] [QuickTime File Format Specification](#)
- [4] [Adobe TIFF](#) Revision 6.0 (1992), Adobe Systems Incorporated (available at <http://www.adobe.com/Support/TechNotes.html> and <ftp://ftp.adobe.com/pub/adobe/DeveloperSupport/TechNotes/PDFfiles>)

