JPEG File Interchange Format

Version 1.02

September 1, 1992

Eric Hamilton
C-Cube Microsystems
1778 McCarthy Blvd.
Milpitas, CA 95035

+1 408 944-6300
Fax: +1 408 944-6314
E-mail: eric@c3.pla.ca.us

JPEG File Interchange Format

Version 1.02

Why a File Interchange Format

JPEG File Interchange Format is a minimal file format which enables JPEG bitstreams 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 JPEG specification or any application specific file format. Nor should it, for the only purpose of this simplified format is to allow the exchange of JPEG compressed images.

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 (CCIR 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

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 be used for the purposes of file interchange. This ensures maximum compatibility with all applications supporting JPEG. JFIF conforms to the JPEG Draft International Standard (ISO DIS 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.

Compatible across platforms

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

Standard color space

The color space to be used is YCbCr as defined by CCIR 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

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.

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.

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.

Conversion to and from RGB

Y, Cb, and Cr are converted from R, G, and B as defined in CCIR Recommendation 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 CCIR 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 R_0, G_0, B_0, ... R_n, G_n, B_n. 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:

$$R = Y + 1.402 (Cr-128)$$
$$G = Y - 0.34414 (Cb-128) - 0.71414 (Cr-128)$$
$$B = Y + 1.772 (Cb-128)$$

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.

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 subsampled components are 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 subsampled 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 subsampled component, $Xoffset_i[0,0]$ and $Yoffset_i[0,0]$, is defined to be

$$Xoffset_i[0,0] = (Nsamples_{ref} / Nsamples_i) / 2 - 0.5$$
$$Yoffset_i[0,0] = (Nlines_{ref} / Nlines_i) / 2 - 0.5$$

where

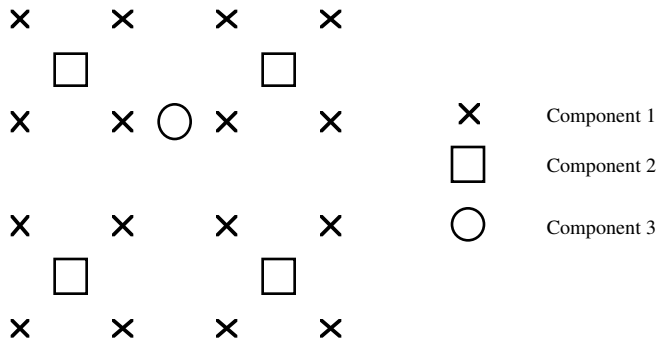
$Nsamples_{ref}$ is the number of samples per line in the largest component,
$Nsamples_i$ is the number of samples per line in the ith component,
$Nlines_{ref}$ is the number of lines in the largest component,
$Nlines_i$ is the number of lines in the ith component.

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



NOTE - This definition is compatible with industry standards such as Postcript Level 2 and QuickTime. This defintition is not compatible with the conventions used by CCIR Recommendation 601-1 and 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.

JPEG File Interchange Format Specification

The syntax of a JFIF file conforms to the syntax for interchange format defined in Annex B of ISO DIS 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', SOF_n, length, frame parameters

Number of components	Nf	= 1 or 3
1st component	C_1	= 1 = Y component
2nd component	C_2	= 2 = Cb component
3rd component	C_3	= 3 = Cr component

•
•
•

X'FF', EOI

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.

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 ISO DIS 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', SOF_n, length, frame parameters

Number of components	Nf	= 1 or 3
1st component	C_1	= 1 = Y component
2nd component	C_2	= 2 = Cb component
3rd component	C_3	= 3 = Cr component

•
•
•

X'FF', EOI

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

JFIF Extension: Thumbnail stored using three bytes per pixel

This extension supports thumbnails stored using three bytes per pixel 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
$(RGB)_n$	(3n bytes)	Packed (24-bit) RGB values for the thumbnail pixels, n = Xthumbnail * Ythumbnail

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.