JPEG File Layout and Format

The File Layout

A JPEG file is partitioned by markers. Each marker is immediately preceded by an all 1 byte (0xff). Although
more markers, We will discuss the following markers:

Marker Name	Marker Identifier	Description
SOI	0xd8	Start of Image
APP0	0xe0	JFIF application segment
APPn	0xe1 – 0xef	Other APP segments
DQT	0xdb	Quantization Table
SOF0	0xc0	Start of Frame
DHT	0xc4	Huffman Table
SOS	0xda	Start of Scan
EOI	0xd9	End of Image

If a 0xff byte occurs in the compressed image data either a zero byte (0x00) or a marker identifier follows it. Nor
only marker that should be found once the image data is started is an EOI. When a 0xff byte is found followed
byte (0x00) the zero byte must be discarded.

A JPEG file consists of the eight following parts:

1. A Start of Image SOI
2. An APP0 Marker
    1. APP0 length
    2. Identifier
    3. Version
    4. Units for X & Y densities
    5. X density
    6. Y density
    7. Thumbnail horizontal pixels
    8. Thumbnail vertical pixels
    9. Thumbnail RGB bitmap
3. APPn Markers where n can be form 1 to 15 (Optional)
    1. APPn length
    2. Application specific information
4. One or more quantization tables DQT
    1. Quantization table length
    2. Quantization table number
    3. Quantization table
5. A Start of Frame SOF0
    1. Start of Frame length
    2. Precision (Bits per pixel per color component)
    3. Image height
    4. Image width
    5. Number of color components
    6. For each component
        1. An ID
        2. A vertical sample factor
        3. A horizontal sample factor
        4. A quantization table#
6. One or more huffman tables DHT

    1. Huffman table length
    2. Type, AC or DC
    3. Index
    4. A Bits table
    5. A Value table
7. A Start of Scan SOS
    1. Start of Scan length
    2. Number of color components
    3. For each component
        1. An ID
        2. An AC table #
        3. An DC table #
    4. Compressed image data (Not included in Start of Scan length)
8. An End of Image EOI

JPEG File Format

Header :

- It occupies two bytes.
- 0xff, 0xd8 (SOI : Start Of Image) (these two identify a JPEG/JFIF file)

Segments or markers:

- Following the SOI marker, there can be any number of “segments” or “markers” such as APP0,DC SOF, SOS and so on.
- An APP0 segment is immediately follows the SOI marker.

Trailer:

- It occupies two bytes.
- 0xff, 0xd9 (EOI: End of Image) (these two identify end of image).

Format of each segment:

Header (4 bytes):

0xff	1byte	identifies segment .
n	1byte	type of segment.
sh, sl	2bytes	size of the segment, including these two bytes, but not including the 0xff and the t Note, not intel order: high byte first, low byte last!

Contents of the segment: max. 65533 bytes.

Notes:

- There are parameterless segments (denoted with a '*' below) that DON'T have a size specificatio contents), just 0xff and the type byte.
- Any number of 0xff bytes between segments is legal and must be skipped.

Segment types:

SOI	0xd8	Start Of Image
APP0	0xe0	JFIF APP0 segment marker,
APP15	0xef	ignore
SOF0	0xc0	Start Of Frame (baseline JPEG), for details see below
SOF1	0xc1	Start Of Frame (baseline JPEG), for details see below
SOF2	0xc2	usually unsupported
SOF3	0xc3	usually unsupported
SOF5	0xc5	usually unsupported
SOF6	0xc6	usually unsupported
SOF7	0xc7	usually unsupported

SOF9	0xc9	for arithmetic coding, usually unsupported
SOF10	0xca	usually unsupported
SOF11	0xcb	usually unsupported
SOF13	0xcd	usually unsupported
SOF14	0xce	usually unsupported
SOF15	0xcf	usually unsupported
DHT	0xc4	Define Huffman Table
DQT	0xdb	Define Quantization Table
SOS	0xda	Start Of Scan
JPG	0xc8	undefined/reserved (causes decoding error)
JPG0	0xf0	ignore (skip)
JPG13	0xfd	ignore (skip)
DAC	0xcc	Define Arithmetic Table, usually unsupported
DNL	0xdc	usually unsupported, ignore
DRI	0xdd	Define Restart Interval, for details see below
DHP	0xde	ignore (skip)
EXP	0xdf	ignore (skip)
*RST0	0xd0	RSTn are used for resync, may be ignored
*RST1	0xd1	
*RST2	0xd2	
*RST3	0xd3	
*RST4	0xd4	
*RST5	0xd5	
*RST6	0xd6	
*RST7	0xd7	
*TEM	0x01	usually causes a decoding error, may be ignored
COM	0xfe	Comment, may be ignored
EOI	0xd9	End Of Image

All other segment types are reserved and should be ignored (skipped).

SOF0 (Start Of Frame 0) marker:

Field	**Size**	**Description**
Marker Identifier	2 bytes	0xff, 0xc0 to identify SOF0 marker
Length	2 bytes	This value equals to 8 + components*3 value
Data precision	1 byte	This is in bits/sample, usually 8 (12 and 16 not supported by most software).
Image height	2 bytes	This must be > 0
Image Width	2 bytes	This must be > 0
Number of components	1 byte	Usually 1 = grey scaled, 3 = color YcbCr or YIQ 4 = color CMYK
Each component	3 bytes	Read each component data of 3 bytes. It contains, (component Id(1byte)(1 = Y, 2 = Cb, 3 = Cr, 4 = I, 5 = Q), sampling factors (1byte) (bit 0-3 vertical., 4-7 horizontal.), quantization table number (1 byte)).

Remarks: JFIF uses either 1 component (Y, greyscaled) or 3 components (YCbCr, sometimes called YUV, colou

APP0 (JFIF segment marker) marker:

Field	Size	Description
Marker Identifier	2 bytes	0xff, 0xe0 to identify APP0 marker
Length	2 bytes	It must be >= 16
File Identifier Mark	5 bytes	This identifies JFIF. 'JFIF'#0 (0x4a, 0x46, 0x49, 0x46, 0x00)
Major revision number	1 byte	Should be 1, otherwise error
Minor revision number	1 byte	Should be 0..2, otherwise try to decode anyway
Units for x/y densities	1 byte	0 = no units, x/y-density specify the aspect ratio instead 1 = x/y-density are dots/inch 2 = x/y-density are dots/cm
X-density	2 bytes	It should be <> 0
Y-density	2 bytes	It should be <> 0
Thumbnail width	1 byte	-------
Thumbnail height	1 byte	-------
Bytes to be read	n bytes	For thumbnail (RGB 24 bit), n = width*height*3 bytes should be read imr followed by thumbnail height

Remarks:

- If there's no 'JFIF'#0, or the length is < 16, then it is probably not a JFIF segment and should be ignor
- Normally units=0, x-dens=1, y-dens=1, meaning that the aspect ratio is 1:1 (evenly scaled).
- JFIF files including thumbnails are very rare, the thumbnail can usually be ignored. If there's no tl then width=0 and height=0.If the length doesn't match the thumbnail size, a warning may be prin continue decoding.

DHT(Define Huffman Table) marker:

Field	Size	Description
Marker Identifier	2 bytes	0xff, 0xc4 to identify DHT marker
Length	2 bytes	This specify length of Huffman table
HT information	1 byte	bit 0..3 : number of HT (0..3, otherwise error) bit 4 : type of HT, 0 = DC table, 1 = AC table bit 5..7 : not used, must be 0
Number of Symbols	16 bytes	Number of symbols with codes of length 1..16, the sum(n) of these bytes is the total number of codes, which must be <= 256
Symbols	n bytes	Table containing the symbols in order of increasing code length (n = total number of codes).

Remarks: A single DHT segment may contain multiple HTs, each with its own information byte.

DRI (Define Restart Interval) marker:

Field	Size	Description
Marker Identifier	2 bytes	0xff, 0xdd identifies DRI marker
Length	2 bytes	It must be 4
Restart interval	2 bytes	This is in units of MCU blocks, means that every n MCU blocks a RSTn marker can be found. The first marker will be RST0, then RST1 etc, after RST7 repeating from RST0.

DQT (Define Quantization Table) marker:

Field	Size	Description
Marker Identifier	2 bytes	0xff, 0xdb identifies DQT
Length	2 bytes	This gives the length of QT.
QT information	1 byte	bit 0..3: number of QT (0..3, otherwise error) bit 4..7: precision of QT, 0 = 8 bit, otherwise 16 bit
Bytes	n bytes	This gives QT values, n = 64*(precision+1)

Remarks:

- A single DQT segment may contain multiple QTs, each with its own information byte.
- For precision=1 (16 bit), the order is high-low for each of the 64 words.

DAC (Define Arithmetic Table) marker:

- Current software does not support arithmetic coding .
- JPEG files using arithmetic coding can not be processed.

SOS (Start Of Scan) marker:

Field	Size	Description
Marker Identifier	2 bytes	0xff, 0xda identify SOS marker
Length	2 bytes	This must be equal to 6+2*(number of components in scan).
Number of Components in scan	1 byte	This must be >= 1 and <=4 (otherwise error), usually 1 or 3
Each component	2 bytes	For each component, read 2 bytes. It contains, 1 byte Component Id (1=Y, 2=Cb, 3=Cr, 4=I, 5=Q), 1 byte Huffman table to use : bit 0..3 : AC table (0..3) bit 4..7 : DC table (0..3)
Ignorable Bytes	3 bytes	We have to skip 3 bytes.

Remarks: The **image data** (scans) is immediately following the SOS segment.

Home | Online Courses | Free C Source Code | Free C# Source Code | Free VC++ Source Code | COM/DCOM Stuff | Courses@Nagpur | Project Ideas | Ask Queries | COM FAQs | Conferences | Discussion Board | Previous Weekly Updates | Good Books | Vedic Maths | Time Pass | Submit Code | About Us | Advertise | Disclaimer

Designed and Managed by
DCube Software Technologies, Nagpur
(India)
Last Revised: 5th July 2002 8:05:20