

Description of Exif file format

2005-05-18

Most of the digital camera's use Exif file format to store images. This specification was made by [JEIDA](#). Some information can be found on the internet. I could not find any upto date document. So I started collecting and published my findings on the web. Here is a tiny description about Exif.

Sometimes DCF is mentioned (Design rule for Camera File system). DCF defines whole file-system of digital camera: directory structure, file naming method, character set and file format etc. The file format of DCF is based on Exif2.1 specification and can be found on [EXIF.org](#). This document is more or less based on Exif2.1/DCF specification

This is a FREE / OPEN document, you may use this document for any purpose (commercial / non-commercial) of 'all' / 'part' of this document.

If you have information about

- 'unknown' items
- faulty data
- missing items

Please e-mail me, Ger Vermeulen geve001@casema.nl

<http://www.gvsoft.homedns.org/exif/>

[rev. 1.1 2005-06-03](#)

[rev. 1.0 2005-04-01](#)

Reference materials

[DCF](#)

[Exif2.1 Specification](#)

[JEIDA](#)

Contents

- 1. [What is Exif?](#)
- 2. [JPEG format and Marker](#)
- 3. [EXIF marker](#)
- 4. [Exif data structure](#)
 - 4.1 [TIFF Header](#)
 - 4.2 [IFD : Image File Directory](#)
 - 4.3 [IFD data structure example](#)
 - 4.4 [Thumbnail image](#)
 - 4.4.1 [JPEG format thumbnail](#)
 - 4.4.2 [TIFF format thumbnail](#)
- [Tag number used by Exif/ TIFF](#)
 - [IFD0 \(IFD of main image\) section](#)
 - [Exif SubIFD section](#)
 - [IFD1 \(IFD of thumbnail image\) section](#)
 - [Misc Tags](#)

1 What is Exif file format?

The Exif file format is based on the JPEG file format. Exif inserts some of image / digital camera information and thumbnail image to JPEG in conformity to JPEG specification. Therefore you can view Exif formatted image files by JPEG compliant Internet browser / Picture viewer / Photo retouch software etc. as normal JPEG image files.

2. JPEG format and Marker

Every JPEG file starts from binary value '0xFFD8', ends with binary value '0xFFD9'. There are several binary 0xFFXX data in JPEG data, they are known as '**Marker**', and it starts a new part of JPEG information. 0xFFD8 means **SOI** (Start of image), 0xFFD9 means **EOI** (End of image). These two special Markers have no data following, the other markers have data with it.

Basic format of Marker is below.

0xFF+Marker Number (1 byte) + Data [Size (2 bytes) + Info (n bytes)]

The '**Data**' contains Data Size descriptor and Information, **Size** (2 Bytes) has 'Motorola' byte align. The data size is the size of the total Data: Size + **Info** (value = 2 + n).

Example: **FF xx - 00 0C - 01 02 03 04 05 06 07 08 09 0A**

It means this Marker (0xFFxx) has 0x000C (12) bytes of data. The data '12 bytes' starts with 'Data size' 2 bytes, followed by 10 bytes: '01 02 03 04 05 06 07 08 09 0A'.

In JPEG format, some of Markers describe data, then **SOS** (Start Of Stream) marker placed. After the SOS marker, JPEG image stream starts and terminated by **EOI** (End Of Image) marker.

marker 'D8' SOI	marker xx size=ssss	marker yy size=tttt	marker 'DA' = SOS size=uuuu	Image stream	marker 'D9' EOI
FFD8	FFxx ssss DDDD...	FFyy tttt DDDD...	FFDA uuuu DDDD....	iiiiiiii...	FFD9

3. EXIF marker

The marker 0xFFE0~0xFFEF is named '**Application Marker**', not necessary for decoding JPEG image. They are used by user application. For example, older olympus / canon / casio / agfa digital camera use JFIF (JPEG File Interchange Format) for storing images. JFIF uses APP0 (0xFFE0) marker for inserting digital camera configuration data and thumbnail image.

For inserting data Exif uses the Application Marker **APP1 (0xFFE1)** marker to avoid a conflict with the JFIF format. Every Exif file format starts from this marker.

SOI marker	APP1 marker	APP1 Data	Other marker
FFD8	FFE1	ssss 457869660000 TTTT.....	FFXX SSSS DDDD.....

It starts with SOI (0xFFD8) marker, so it's a JPEG file. Then an APP1 marker follows immediately. All the data of Exif is stored in the APP1 data area. The APP1 Data part starts with size 'ssss' (Exif data area). The

size 'ssss' includes the size field also.

After the 'SSSS', APP1 data starts. This part is explained in the next section.

After the APP1 marker area other JPEG markers follow.

4. Exif data structure

The structure of Exif data (APP1) is shown below. This is a case of 'Intel' byte aligns and it contains JPEG format thumbnail. The Exif data starts from ASCII character 'Exif' and 2bytes of 0x00, and then Exif data follows. Exif uses TIFF format to store data. For more details of TIFF format, please refer to ['TIFF6.0 specification'](#).

APP1 Marker		FFE1	Application marker E1	
	APP1 Data Size	ssss		
	Exif Header	45786966 0000		
	TIFF Header	4949		
		2A00		
		08000000		
	IFD0 (main image)	xxxx	Number of fields in directory	
		aaaabbbbccccccccddddddd	Directory	Image width
		aaaabbbbccccccccddddddd	
		aaaabbbbccccccccddddddd		Exif offset
		aaaabbbbccccccccddddddd		GPS offset
		xxxxxxxx	Next IFD offset (IFD1)	
		xxxx. . .	Data area of IFD0	
	EXIF IFD	xxxx	Number of fields in directory	
		aaaabbbbccccccccddddddd	Directory	EXIF version
		aaaabbbbccccccccddddddd		Date digitised
		aaaabbbbccccccccddddddd		Makernote Offset
		aaaabbbbccccccccddddddd	
		00000000	No next IFD	
		xxxx. . .	Data area of Exif IFD	
	GPS IFD	xxxx	Number of fields in directory	
		aaaabbbbccccccccddddddd	Directory	GPS version
		aaaabbbbccccccccddddddd	
		aaaabbbbccccccccddddddd	
		00000000	No next IFD	
		xxxx. . .	Data area of Gps IFD	
		FUJIFILM	Optional recognise string	

APP1 Data	Makernote IFD	xxxx	Number of fields in directory	
		aaaabbbbccccccccddddddd	Directory	Familie ID
		aaaabbbbccccccccddddddd		ISO speed
		aaaabbbbccccccccddddddd	
		00000000	No next IFD	
		xxxx. . . .	Data area of makernote IFD	
	Interoperability IFD	xxxx	Number of fields in directory	
		aaaabbbbccccccccddddddd	Directory	Index
		aaaabbbbccccccccddddddd		Version
		aaaabbbbccccccccddddddd	
		00000000	No next IFD	
		xxxx. . . .	Data area of Interoperability IFD	
	IFD1 (thumbnail image)	xxxx	Number of fields in directory	
		aaaabbbbccccccccddddddd	Directory	Compression
		aaaabbbbccccccccddddddd		XResolution
		aaaabbbbccccccccddddddd	
		00000000	No next IFD	
		xxxx. . . .	Data area of Interoperability IFD	
	Sometimes some unknown stuff			
	Thumbnail image	FFD8	SOI - Start of Image	
		FFDB ssss dddd. . . .	DQT - Define quantization table	
		FFC0 ssss dddd. . . .	SOF0 - Baseline	
		FFC4 ssss dddd. . . .	DHT - Define Huffman table	
		FFDA ssss dddd. . . .	SOS - Start of Scan	
		xxxxxxxx	image	
		FFD9	EOI - End of Image	

4.1 TIFF header structure

Byte align	TAG mark	Offset to first IFD
0x4D4D = 'MM'	0x002a	0x00000008
or		
0x4949 = 'I I'	0x2a00	0x08000000

First 8 bytes of TIFF format is the TIFF header.

First 2 bytes defines byte align of TIFF data. 0x4949='I I' is 'Intel' type byte align, 0x4d4d='MM' is

'Motorola' type byte align. For example, decimal value '305.419.896' is noted as hexadecimal 0x12345678. If Motorola aligned, it is noted as 0x12345678, if it's Intel aligned it is noted as 0x78563412. A lot of digital camera uses Intel align, but not all. Therefore when we need the value of Exif data, we MUST check byte align field every time. Though JPEG data uses Motorola align only, Exif allows both alignments. I can't understand why Exif didn't fix a byte align to Motorola.

Next 2 bytes always have the value of 0x002A.

The last 4 bytes of TIFF header is the offset to the first IFD (Image File Directory, described in next chapter). Includes this offset, all the offset value used in TIFF format counts offset bytes from the first byte of the TIFF header ('I I' or 'MM'). Usually the first IFD starts immediately next to TIFF header, so this offset has often the value '0x00000008'.

4.2 IFD : Image File Directory

Next to TIFF header, there is the first IFD : Image File Directory - IFD-0. It contains image information data.

eeee	Number of directory entries
tttt - ffff - nnnnnnnn - dddddddd	Entry 0
tttt - ffff - nnnnnnnn - dddddddd	Entry 1
.....
tttt - ffff - nnnnnnnn - dddddddd	Entry eeee- 1
oooooooo	Offset to next IFD

The first 2 bytes ('eeee') contains the number of directory entries in this IFD.

Then directory entries (12 bytes per entry) follows.

'tttt' 2 bytes is the Tag number, this shows a kind of data.

'ffff' 2 bytes is data format.

'nnnnnnnn' 4 bytes is number of components.

'ddddddd' 4 bytes contains a data value or offset to data value.

After last directory entry, there is a 4bytes of data ('oooooooo'), it has the offset to next IFD. If its value is '0x00000000', it means this is the last IFD and there is no linked IFD.

Data format ('ffff') is defined as see table.

Value	Format	Bytes/component
1	unsigned byte	1
2	ascii strings	1
3	unsigned short	2
4	unsigned long	4
5	unsigned rational	8
6	signed byte	1
7	undefined	1

8	signed short	2
9	signed long	4
10	signed rational	8
11	single float	4
12	double float	8

'rational' means a fractional value, it contains 2-signed/unsigned long integer values, and the first represents the numerator, and the second, the denominator.

You can get the total data byte length by multiplies a 'bytes/components' value by number of components stored 'nnnnnnnnn' area. If total data length is less or equal than 4bytes, 'dddddddd' contains the value of that Tag. If its size is over 4bytes, 'dddddddd' contains the offset to data stored address.

4.3 IFD data structure example

At Exif format, the first IFD is IFD0 (IFD of main image), then it links to IFD1 (IFD of thumbnail image) and IFD link is terminated. But IFD0/IFD1 doesn't contain any digital camera's information such as shutter speed, focal length etc. IFD0 always contains special Tag **Exif Offset (0x8769)**, it shows an offset to **Exif SubIFD**. Exif SubIFD is IFD formatted data also, it contains digital camera's information.

In case of Extended Exif format (Exif2.1/DCF), Exif SubIFD contains special Tag **Exif Interoperability Offset (0xa005)**. It also points to the Interoperability IFD. By the DCF specification, this tag is mandatory and both SubIFD (main image IFD) and IFD1 (thumbnail image IFD) may have Interoperability IFD. In usual, only main image have this tag.

And some of digital camera uses IFD data format for Makernote; Maker-specific magic data area. It's very hard to decide whether makernote is IFD format or not, be careful to coding program. See Appendix for information of Makernote.

```
0000: 49 49 2A 00 08 00 00 00-02 00 1A 01 05 00 01 00
0010: 00 00 26 00 00 00 69 87-04 00 01 00 00 00 11 02
0020: 00 00 40 00 00 00 48 00-00 00 01 00 00 00 00
```

If the first part of TIFF data is above, it can read as:

Address	Hex data	Explanation
0x0000~0x0001	0x4949	the first 2bytes are 'I I', byte align is 'Intel'
0x0002~0x0003	0x2A00	
0x0004~0x0007	0x08000000	IFD0 starts from address '0x00000008'
0x0008~0x0009	0x0200	number of directory entry of IFD0 is '2'
0x000a~0x000b	0x1A01	number for Tag: X-Resolution (0x011A), it contains a horizontal resolution of image
0x000c~0x000d	0x0500	format is 0x0005, unsigned rational
0x000e~0x0011	0x01000000	number of components is '1'
.	.	Unsigned rational data size is 8 bytes/components, so total data length is 1 x 8 = 8 bytes Total data length is larger than 4 bytes, so next 4 bytes contains an offset to the data
0x0012~0x0015	0x26000000	X-Resolution data is stored to address 0x00000026

0x0016~0x0017	0x6987	next Tag is ExifOffset (0x8769), its value is an offset to Exif SubIFD
0x0018~0x0019	0x0400	format is 0x0004, unsigned long integer.
0x001a~0x001d	0x01000000	one component. Unsigned long integer's data size is 4 bytes/components, so total data size is 4 bytes
.	.	Total data size is equal to 4 bytes, so next 4 bytes contains the value for ExifSubIFD offset
0x001e~0x0021	0x11020000	ExifSubIFD starts from address '0x00000211'
.	.	This was the last directory entry, next 4 bytes is an offset to next IFD
0x0022~0x0025	0x40000000	next IFD starts from address '0x00000040'
0x0026~0x0029	0x48000000	numerator is 72
0x002a~0x002d	0x01000000	denominator is '1'
.	.	So the value of X-Resolution is 72/1

4.4 Thumbnail image

Exif format contains a thumbnail of the image (except Ricoh RDC-300Z). Usually it is located next to the IFD1. There are 3 formats for thumbnails; JPEG format (JPEG uses YCbCr), RGB TIFF format, YCbCr TIFF format. It seems that JPEG format and 160x120 pixels of size are recommended thumbnail format for Exif2.1 or later. By the DCF specification, thumbnail image **MUST** use JPEG format and image size is fixed to 160x120 pixels.

4.4.1 JPEG format thumbnail

If the value of **Compression (0x0103)** Tag in IFD1 is '6', thumbnail image format is JPEG. Most of Exif image uses JPEG format for thumbnail. In that case, you can get offset of thumbnail from **JpegIFOffset (0x0201)** Tag in IFD1, size of thumbnail from **JpegIFByteCount (0x0202)** Tag. Data format is ordinary JPEG format, starts from 0xFFD8 and ends by 0xFFD9.

4.4.2 TIFF format thumbnail

If the value of **Compression (0x0103)** Tag in IFD1 is '1', thumbnail image format is no compression (called TIFF image). Start point of thumbnail data is **StripOffset (0x0111)** Tag, size of thumbnail is the sum of **StripByteCounts (0x0117)** Tag.

If thumbnail uses no compression and **PhotometricInterpretation (0x0106)** Tag in IFD1 has a value '2', thumbnail uses RGB format. In that case, you can see thumbnail image by simply copy data to computer's RGB format (such as BMP format, or copy to VRAM directory). Kodak DC-210/220/260 uses this format. Be mention that at TIFF stores pixel data as 'RGB' order, but BMP stores 'BGR' order.

If that tag has a value '6', thumbnail uses YCbCr format. If you want to see thumbnail, you must convert it to RGB. Ricoh RDC4200/4300, Fuji DS-7/300 and DX-5/7/9 use this format (newer RDC5000/MX-X00 series use JPEG). Next section is brief description to conversion of Fuji DS's thumbnail. For more details, refer to [TIFF6.0 specification](#).

At DX-5/7/9, YCbCrSubsampling(0x0212) has values of '2,1', PlanarConfiguration(0x011c) has a value '1'. So the data align of this image is below.

Y(0,0),Y(1,0),Cb(0,0),Cr(0,0), Y(2,0),Y(3,0),Cb(2,0),Cr(3,0), Y(4,0),Y(5,0),Cb(4,0),Cr(4,0). . . .

The numeric in parenthesis is pixel coordinates. DX series' YCbCrCoefficients(0x0211) has values '0.299/0.587/0.114', ReferenceBlackWhite(0x0214) has values '0,255,128,255,128,255'. Therefore to convert from Y/Cb/Cr to RGB is;

$$B(0,0)=(Cb-128)*(2-0.114*2)+Y(0,0)$$

$$R(0,0)=(Cr-128)*(2-0.299*2)+Y(0,0)$$

$$G(0,0)=(Y(0,0)-0.114*B(0,0)-0.299*R(0,0))/0.587$$

Horizontal subsampling is a value '2', so you can calculate B(1,0)/R(1,0)/G(1,0) by using the Y(1,0) and Cr(0,0)/Cb(0,0). Repeat this conversion by value of ImageWidth(0x0100) and ImageLength(0x0101).