FAT Directory – The Entry

- Directory Entry for Cop.txt
  - Again for the purpose of this section, the directory entry is going to be considered a Record, with data at certain areas ~ offsets 1 through 32.
  - Clearly the first offset in a file would be 0, not 1 - each Entry will be broken down as offsets 1 – 32 as if it were a Record

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
43	4F	50	20	20	20	20	20	54	58	54	20	18	84	82	70	9F	2D	9F	2D	00	00	F1	71	9F	2D	16	02	FC	08	00	00
Status	Directory Name							Extension			Attributes	Reserved	Directory	Created				Accessed Date		Unused		Written				Cluster	Starting	Directory Size				
														Time		Date						Time		Date							

FAT Directory – Created Time & Date

- Created Time & Date – 4 Bytes
  - Broken down in 2 byte segments each or 16 Bits.
    - Created Time - Offsets 15 & 16
    - Created Date- Offsets 17 & 18

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
43	4F	50	20	20	20	20	20	54	58	54	20	18	84	82	70	9F	2D	9F	2D	00	00	F1	71	9F	2D	16	02	FC	08	00	00
Status	Directory Name							Extension			Attributes	Reserved	Directory	Created				Accessed Date		Unused		Written				Cluster Starting		Directory Size				
														Time		Date						Time		Date							

FAT Directory – Big or Little Endian

- Before we can dissect the Date & Time Information we must first know whether the data is stored Big or Little Endian.
- In our example, the Created Time information Hex Values are 82 70

FAT Directory – Big or Little Endian

- Intel Processors processed binary data “Little Endian”
  - This is the order in which the processor processes the data.
  - DOS, Windows and Linux on Intel processors will store data Little Endian.
  - Some files such as bitmaps images store data Big Endian regardless of the processor
  - Unix and Macintosh are both Big Endian systems

FAT Directory – Big or Little Endian

- Fortunately, The Processor knows how to interpret the data correctly, But…
  - When converting the data manually, you must input it in reverse order or Least Significant Byte First
    - Hex Values of 82 70 for Created Time must be input manually as 70 82.
    - EnCase will convert this data for you automatically using Date & Time Functions.
  - However to better understand this concept we will do the conversion manually using a Base or Hex Converter.

FAT Directory – Created Time

- File Created Time of 14:04:04 would display as 82 70 in Hex (16 bit Integer)
  - This must be converted into binary as Little Endian
    - Input 70 82 into Base Converter – Little Endian
    - The binary result is 0111000010000010
    - The First 5 Bits are Hours 01110000010000010
    - The Next 6 Bits are Minutes 01110000100000010
    - The Last 5 Bits are Seconds 01110000100000010
      - Seconds are Multiplied by 2, this is why all DOS dates are even number seconds.

FAT Directory – Created Time

- Created Time in binary (16 Bit Integer)
  - 0111000010000010
- The bits of the 16 bit field for the Directory Entry Creation Time are not read as one 16 bit number.
- Instead the 16 Bits of the 2 bytes are broken down and segmented.

FAT Directory – Created Time

0111000010000010 = Created Time in binary (16 Bit Integer)

The Bit Values are not interpreted as a Straight 16 Bit Value as shown here.															
32768	16384	8092	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0

The Values are adjusted for the appropriate field of data being interpreted															
16	8	4	2	1	32	16	8	4	2	1	16	8	4	2	1
0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0
Hours					Minutes						Seconds x2				

Next we add the values of the bits

FAT Directory – Created Time

- File Created Time in binary (16 Bit Integer)
  - 0111000010000010 = 14:04:04

Hours					Minutes						Seconds x2				
16	8	4	2	1	32	16	8	4	2	1	16	8	4	2	1
0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0
Total Value - 14					Total Value = 4						Total Value = 2 (x2) or 4				

- Hours =14 Minutes = 4 Seconds x2 = 4
- 14:04:04 is the Created File Time

FAT Directory – Created Date

- File Created Date of 12/31/2002 in Hex would display as 9F 2D (16 bit Integer)
  - 9F 2D > 2D 9F
    - The binary result is 0010110110011111
    - The First 7 Bits are the Year 0010110100111111
      - Add 1980
    - The Next 4 Bits are the Month 0010110110011111
      - Value of 1-12
    - The Last 5 Bits are the Day 0010110110011111
      - Value of 1-31

FAT Directory – Created Date

- Created Date in binary (16 Bit Integer)
  - 0010110110011111
- Just as Created Time Data, the 16 Bit data field for the Creation Time is not read as one 16 bit number.
- Instead the 16 Bits of the 2 bytes are broken down and segmented into Year Month & Day

64	32	16	8	4	2	1	8	4	2	1	16	8	4	2	1
Year							Month				Day				

FAT Directory – Created Date

- File Created Date in binary (16 Bit Integer)
  - 0010110110011111 = 12/31/2002

Year + 1980							Month				Day				
64	32	16	8	4	2	1	8	4	2	1	16	8	4	2	1
0	0	1	0	1	1	0	1	1	0	0	1	1	1	1	1
Total Value = 22 + 1980 or 2002							Total Value = 12				Total Value = 31				

- Year = 22 + 1980 = 2002 Month = 12 Day = 31
- 12/31/2002 is the Created File Date

Reference

- Al Hobbs, Understanding Impact of Individual Bits within a Byte, and working with Multi-byte values, Lake County States Attorney Investigations & Guidance Software Inc.