DOS Executable file format

- All multi-byte values are stored LSB first. One block is 512 bytes, one paragraph is 16 bytes.
- If the word at offset 02h is 4, it should be treated as 00h, since pre-1.10 versions of the MS linker set it that way.
- If both minimum and maximum allocation (offset 0Ah/0Ch) are zero, the program is loaded as high in memory as possible (DOS only checks the maximum allocation, however).
- The maximum allocation is set to FFFFh by default.
- Additional data may be contained in the file beyond the end of the load image described by the .EXE header; this data may be overlays, the actual executable for newer-format executables, or debugging information (see #01600,#01624).
- Relocations entries need not be in any particular order, although they are typically stored in order from beginning to end of the load image.

Offset	Size	Name	Description
00h	WORD	e_magic	0x4d, 0x5a or 0x5a, 0x4d. This is the “magic number” of an EXE file
02h	WORD	e_cblp	The number of bytes in the last block of the program that are actually used. If this value is zero, that means the entire last block is used (i.e. the effective value is 512).
04h	WORD	e_cp	Number of blocks in the file that are part of the EXE file. If [02-03] is non-zero, only that much of the last block is used.
06h	WORD	e_crlc	Number of relocation entries stored after the header. May be zero.
08h	WORD	e_cparhdr	Number of paragraphs in the header. The program's data begins just after the header, and this field can be used to calculate the appropriate file offset. The header includes the relocation entries. Note that some OSs and/or programs may fail if the header is not a multiple of 512 bytes.
0Ah	WORD	e_minalloc	Number of paragraphs of additional memory that the program will need. This is the equivalent of the BSS size in a Unix program. The program can't be loaded if there isn't at least this much memory available to it.
0Ch	WORD	e_maxalloc	Maximum number of paragraphs of additional memory. Normally, the OS reserves all the remaining conventional memory for your program, but you can limit it with this field.
0EH	WORD	e_ss	Relative value of the stack segment. This value is added to the segment the program was loaded at, and the result is used to initialize the SS register.
10h	WORD	e_sp	Initial value of the SP register.
12h	WORD	e_csum	Word checksum. If set properly, the 16-bit sum of all words in the file should be zero. Usually, this isn't filled in.
14h	WORD	e_ip	Initial value of the IP register.
16h	WORD	e_cs	Initial value of the CS register, relative to the segment the program was loaded at.
18h	WORD	e_lfarlc	Offset of the first relocation item in the file.
1Ah	WORD	e_ovno	Overlay number. Normally zero, meaning that it's the main program.
1Ch	DWORD	e_res	
20h	WORD	e_oemid	
22h	WORD	e_oeminfo	
24h	24 WORD	e_res2	
2ch	DWORD	e_lfanew	

Relocation table entry

Offset	Size	Name	Description
00h	WORD	offset	
02h	WORD	segment	