

MZ EXE Format
Intel byte order

Information from File Format List 2.0 by Max Maischein.

-----!-CONTACT_INFO-----

If you notice any mistakes or omissions, please let me know! It is only with YOUR help that the list can continue to grow. Please send all changes to me rather than distributing a modified version of the list.

This file has been authored in the style of the INTERxxy.* file list by Ralf Brown, and uses almost the same format.

Please read the file FILEFMTS.1ST before asking me any questions. You may find that they have already been addressed.

Max Maischein

Max Maischein, 2:244/1106.17
Max_Maischein@spam.fido.de
corion@informatik.uni-frankfurt.de
Corion on #coders@IRC

-----!-DISCLAIMER-----

DISCLAIMER: THIS MATERIAL IS PROVIDED "AS IS". I verify the information contained in this list to the best of my ability, but I cannot be held responsible for any problems caused by use or misuse of the information, especially for those file formats foreign to the PC, like AMIGA or SUN file formats. If an information it is marked "guesswork" or undocumented, you should check it carefully to make sure your program will not break with an unexpected value (and please let me know whether or not it works the same way).

Information marked with "???" is known to be incomplete or guesswork.

Some file formats were not released by their creators, others are regarded as proprietary, which means that if your programs deal with them, you might be looking for trouble. I don't care about this.

The old EXE files are the EXE files executed directly by MS-DOS. They were a major improvement over the old 64K COM files, since EXE files can span multiple segments. An EXE file consists of three different parts, the header, the relocation table and the binary code.

The header is expanded by a lot of programs to store their copyright information in the executable, some extensions are documented below.

The format of the header is as follows :

OFFSET	Count	TYPE	Description
0000h	2	char	ID='MZ' ID='ZM'
0002h	1	word	Number of bytes in last 512-byte page of executable
0004h	1	word	Total number of 512-byte pages in executable (including the last page)
0006h	1	word	Number of relocation entries
0008h	1	word	Header size in paragraphs
000Ah	1	word	Minimum paragraphs of memory allocated in addition to the code size
000Ch	1	word	Maximum number of paragraphs allocated in addition to the code size
000Eh	1	word	Initial SS relative to start of executable
0010h	1	word	Initial SP
0012h	1	word	Checksum (or 0) of executable
0014h	1	dword	CS:IP relative to start of executable (entry point)
0018h	1	word	Offset of relocation table; 40h for new-(NE,LE,LX,W3,PE etc.) executable
001Ah	1	word	Overlay number (0h = main program)

Following are the header expansions by some other programs like TLink, LZExe and other linkers, encryptors and compressors; all offsets are relative to the start of the whole header :

---new executable

OFFSET	Count	TYPE	Description
001Ch	4	byte	????
0020h	1	word	Behaviour bits ??
0022h	26	byte	reserved (0)
003Ch	1	dword	Offset of new executable header from start of file (or 0 if plain MZ executable)

---Borland TLINK

OFFSET	Count	TYPE	Description
001Ch	2	byte	?? (apparently always 01h 00h)
001Eh	1	byte	ID=0FBh
001Fh	1	byte	TLink version, major in high nybble
0020h	2	byte	??

---old ARJ self-extracting archive

OFFSET	Count	TYPE	Description
001Ch	4	char	ID='RJSX' (older versions) new signature is 'aRJsf'" in the first 1000 bytes of the file)

---LZEXE compressed executable

OFFSET	Count	TYPE	Description
001Ch	2	char	ID='LZ'
001Eh	2	char	Version number : '09' - LZExe 0.90 '91' - LZExe 0.91

---PKLITE compressed executable

OFFSET	Count	TYPE	Description
001Ch	1	byte	Minor version number
001Dh	1	byte	Bit mapped : 0-3 - major version 4 - Extra compression 5 - Multi-segment file
001Eh	6	char	ID='PKLITE'

---LHarc 1.x self-extracting archive

OFFSET	Count	TYPE	Description
001Ch	4	byte	unused???
0020h	3	byte	Jump to start of extraction code
0023h	2	byte	???
0025h	12	char	ID='LHarc's SFX '

---LHA 2.x self-extracting archive

OFFSET	Count	TYPE	Description
001Ch	8	byte	???
0024h	10	char	ID='LHa's SFX ' For version 2.10 ID='LHA's SFX ' (v2.13) For version 2.13

---LH self-extracting archive

OFFSET	Count	TYPE	Description
001Ch	8	byte	???
0024h	8	byte	ID='LH's SFX '

---TopSpeed C 3.0 CRUNCH compressed file

OFFSET	Count	TYPE	Description
001Ch	1	dword	ID=018A0001h
0020h	1	word	ID=1565h

---PKARC 3.5 self-extracting archive

OFFSET	Count	TYPE	Description
001Ch	1	dword	ID=00020001h
0020h	1	word	ID=0700h

---BSA (Soviet archiver) self-extracting archive

OFFSET	Count	TYPE	Description
001Ch	1	word	ID=000Fh
001Eh	1	byte	ID=A7h

---LARC self-extracting archive

OFFSET	Count	TYPE	Description
001Ch	4	byte	???
0020h	11	byte	ID='SFX by LARC '

After the header, there follow the relocation items, which are used to span multiple segments. The relocation items have the following format :

OFFSET	Count	TYPE	Description
0000h	1	word	Offset within segment
0002h	1	word	Segment of relocation

To get the position of the relocation within the file, you have to compute the

physical address from the segment:offset pair, which is done by multiplying the segment by 16 and adding the offset and then adding the offset of the binary start. Note that the raw binary code starts on a paragraph boundary within the executable file. All segments are relative to the start of the executable in memory, and this value must be added to every segment if relocation is done manually.

EXTENSION:EXE,OVR,OVL

OCCURENCES:PC

PROGRAMS:MS-DOS

REFERENCE:Ralf Brown's Interrupt List

SEE ALSO:COM,EXE,NE EXE