

## PE related data structures

```
typedef struct __IMAGE_DOS_HEADER { // DOS .EXE header
    unsigned short e_magic;           // Magic number
    unsigned short e_cblp;           // Bytes on last page of file
    unsigned short e_cp;             // Pages in file
    unsigned short e_crlc;           // Relocations
    unsigned short e_cparhdr;        // Size of header in paragraphs
    unsigned short e_minalloc;       // Minimum extra paragraphs needed
    unsigned short e_maxalloc;       // Maximum extra paragraphs needed
    unsigned short e_ss;             // Initial (relative) SS value
    unsigned short e_sp;             // Initial SP value
    unsigned short e_csum;           // Checksum
    unsigned short e_ip;             // Initial IP value
    unsigned short e_cs;             // Initial (relative) CS value
    unsigned short e_lfarlc;         // File address of relocation table
    unsigned short e_ovno;           // Overlay number
    unsigned short e_res[4];         // Reserved unsigned shorts
    unsigned short e_oemid;          // OEM identifier (for e_oeminfo)
    unsigned short e_oeminfo;        // OEM information; e_oemid specific
    unsigned short e_res2[10];       // Reserved unsigned shorts
    unsigned int e_lfanew;           // 0x3C File address of new exe header
} _IMAGE_DOS_HEADER;
```

*pointer to X*

```
typedef struct __IMAGE_FILE_HEADER {
    unsigned short Machine;           //0
    unsigned short NumberOfSections;  //2
    unsigned int TimeDateStamp;        //4
    unsigned int PointerToSymbolTable; //8
    unsigned int NumberOfSymbols;      //12
    unsigned short SizeOfOptionalHeader; //16
    unsigned short Characteristics;    //18
} _IMAGE_FILE_HEADER;                //size 20
```

*Sig 4+*

```
typedef struct __IMAGE_DATA_DIRECTORY {
    unsigned int VirtualAddress;
    unsigned int Size;
} _IMAGE_DATA_DIRECTORY;
```

```
typedef struct __IMAGE_OPTIONAL_HEADER {
    unsigned short Magic;                //0
    unsigned char MajorLinkerVersion;    //2
    unsigned char MinorLinkerVersion;    //3
    unsigned int SizeOfCode;             //4
    unsigned int SizeOfInitializedData;  //8
    unsigned int SizeOfUninitializedData; //12
    unsigned int AddressOfEntryPoint;    //16
    unsigned int BaseOfCode;             //20
    unsigned int BaseOfData;             //24
    unsigned int ImageBase;              //28
    unsigned int SectionAlignment;       //32
```

```

unsigned int    FileAlignment;           //36
unsigned short  MajorOperatingSystemVersion; //40
unsigned short  MinorOperatingSystemVersion; //42
unsigned short  MajorImageVersion;       //44
unsigned short  MinorImageVersion;       //46
unsigned short  MajorSubsystemVersion;    //48
unsigned short  MinorSubsystemVersion;    //50
unsigned int    Win32VersionValue;        //52
unsigned int    SizeOfImage;              //56
unsigned int    SizeOfHeaders;            //60
unsigned int    CheckSum;                 //64
unsigned short  Subsystem;                //68
unsigned short  DllCharacteristics;       //70
unsigned int    SizeOfStackReserve;       //72
unsigned int    SizeOfStackCommit;        //76
unsigned int    SizeOfHeapReserve;        //80
unsigned int    SizeOfHeapCommit;        //84
unsigned int    LoaderFlags;              //88
unsigned int    NumberOfRvaAndSizes;      //92
    _IMAGE_DATA_DIRECTORY DataDirectory[16]; //96
} _IMAGE_OPTIONAL_HEADER32;               //size 224

```

X {

```

typedef struct __IMAGE_NT_HEADERS {
    unsigned int Signature; 4
    _IMAGE_FILE_HEADER FileHeader;
    _IMAGE_OPTIONAL_HEADER32 OptionalHeader;
} _IMAGE_NT_HEADERS;

```

```

typedef struct __IMAGE_SECTION_HEADER {
    unsigned char    Name[8];           //0
    unsigned int     VirtualSize;       //8
    unsigned int     VirtualAddress;    //12
    unsigned int     SizeOfRawData;     //16
    unsigned int     PointerToRawData;  //20
    unsigned int     PointerToRelocations; //24
    unsigned int     PointerToLinenumbers; //28
    unsigned short   NumberOfRelocations; //32
    unsigned short   NumberOfLinenumbers; //34
    unsigned int     Characteristics;   //36
} _IMAGE_SECTION_HEADER;               //size 40

```