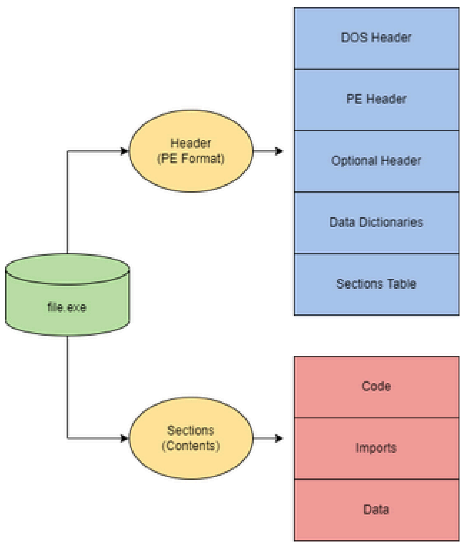
4 - Portable Executable Format

Task 6 Portable Executable Format



Executables and applications are a large portion of how Windows internals operate at a higher level. The PE (Portable Executable) format defines the information about the executable and stored data. The PE format also defines the structure of how data components are stored.

The PE (Portable Executable) format is an overarching structure for executable and object files. The PE (Portable Executable) and COFF (Common Object File Format) files make up the PE format.

PE data is most commonly seen in the hex dump of an executable file. Below we will break down a hex dump of calc.exe into the sections of PE data.

The structure of PE data is broken up into seven components,

The **DOS Header** defines the type of file

The `MZ` DOS header defines the file format as `.exe`. The DOS header can be seen in the hex dump section below.

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00  ............è...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
```

The **DOS Stub** is a program run by default at the beginning of a file that prints a compatibility message. This does not affect any functionality of the file for most users.

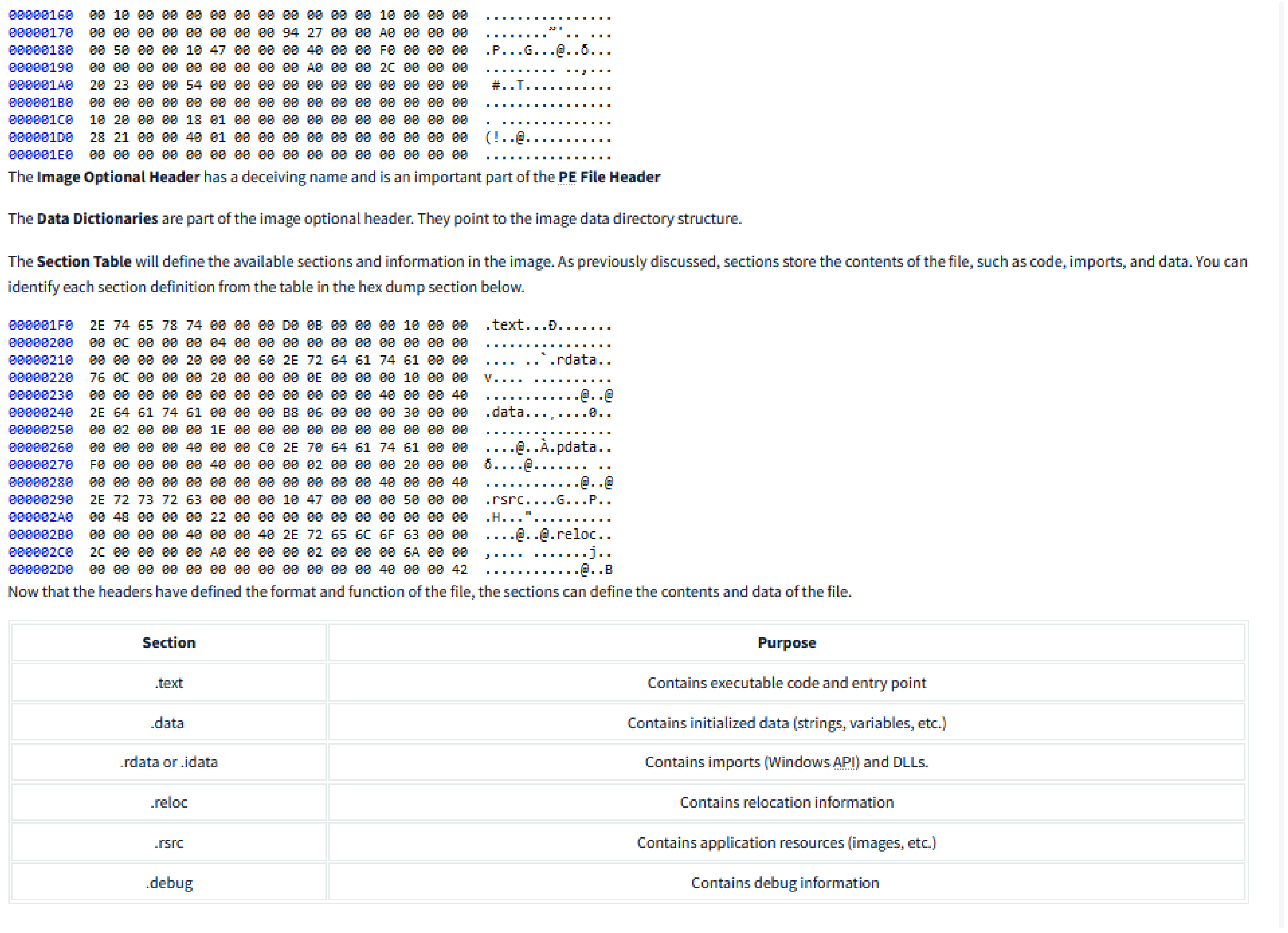
The DOS stub prints the message `This program cannot be run in DOS mode`. The DOS stub can be seen in the hex dump section below.

```
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
```

The **PE File Header** provides PE header information of the binary. Defines the format of the file, contains the signature and image file header, and other information headers.

The PE file header is the section with the least human-readable output. You can identify the start of the PE file header from the `PE` stub in the hex dump section below.

```
000000E0  00 00 00 00 00 00 00 00 50 45 00 00 64 86 06 00  ........PE..d†..
000000F0  10 C4 40 03 00 00 00 00 00 00 00 00 F0 00 22 00  .Ä@.........ð.".
00000100  0B 02 0E 14 00 0C 00 00 00 62 00 00 00 00 00 00  .........b......
00000110  70 18 00 00 00 10 00 00 00 00 00 40 01 00 00 00  p..........@....
00000120  00 10 00 00 00 02 00 00 0A 00 00 00 0A 00 00 00  ................
00000130  0A 00 00 00 00 00 00 00 00 B0 00 00 00 04 00 00  .........°......
00000140  63 41 01 00 02 00 60 C1 00 00 08 00 00 00 00 00  cA....`Á........
00000150  00 20 00 00 00 00 00 00 00 00 10 00 00 00 00 00  . ..............
00000160  00 10 00 00 00 00 00 00 00 00 00 00 10 00 00 00  ................
00000170  00 00 00 00 00 00 00 00 94 27 00 00 A0 00 00 00  ........”'.. ...
00000180  00 50 00 00 10 47 00 00 00 40 00 00 F0 00 00 00  .P...G...@..ð...
00000190  00 00 00 00 00 00 00 00 00 A0 00 00 2C 00 00 00  ......... ..,...
```



```
00000160  00 10 00 00 00 00 00 00 00 00 00 00 10 00 00 00  ................
00000170  00 00 00 00 00 00 00 00 94 27 00 00 A0 00 00 00  ........”'.. ...
00000180  00 50 00 00 10 47 00 00 00 40 00 00 F0 00 00 00  .P...G...@..ð...
00000190  00 00 00 00 00 00 00 00 00 A0 00 00 2C 00 00 00  ......... ..,...
000001A0  20 23 00 00 54 00 00 00 00 00 00 00 00 00 00 00   #..T...........
000001B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001C0  10 20 00 00 18 01 00 00 00 00 00 00 00 00 00 00  . ..............
000001D0  28 21 00 00 40 01 00 00 00 00 00 00 00 00 00 00  (!..@...........
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

The **Image Optional Header** has a deceiving name and is an important part of the **PE File Header**

The **Data Dictionaries** are part of the image optional header. They point to the image data directory structure.

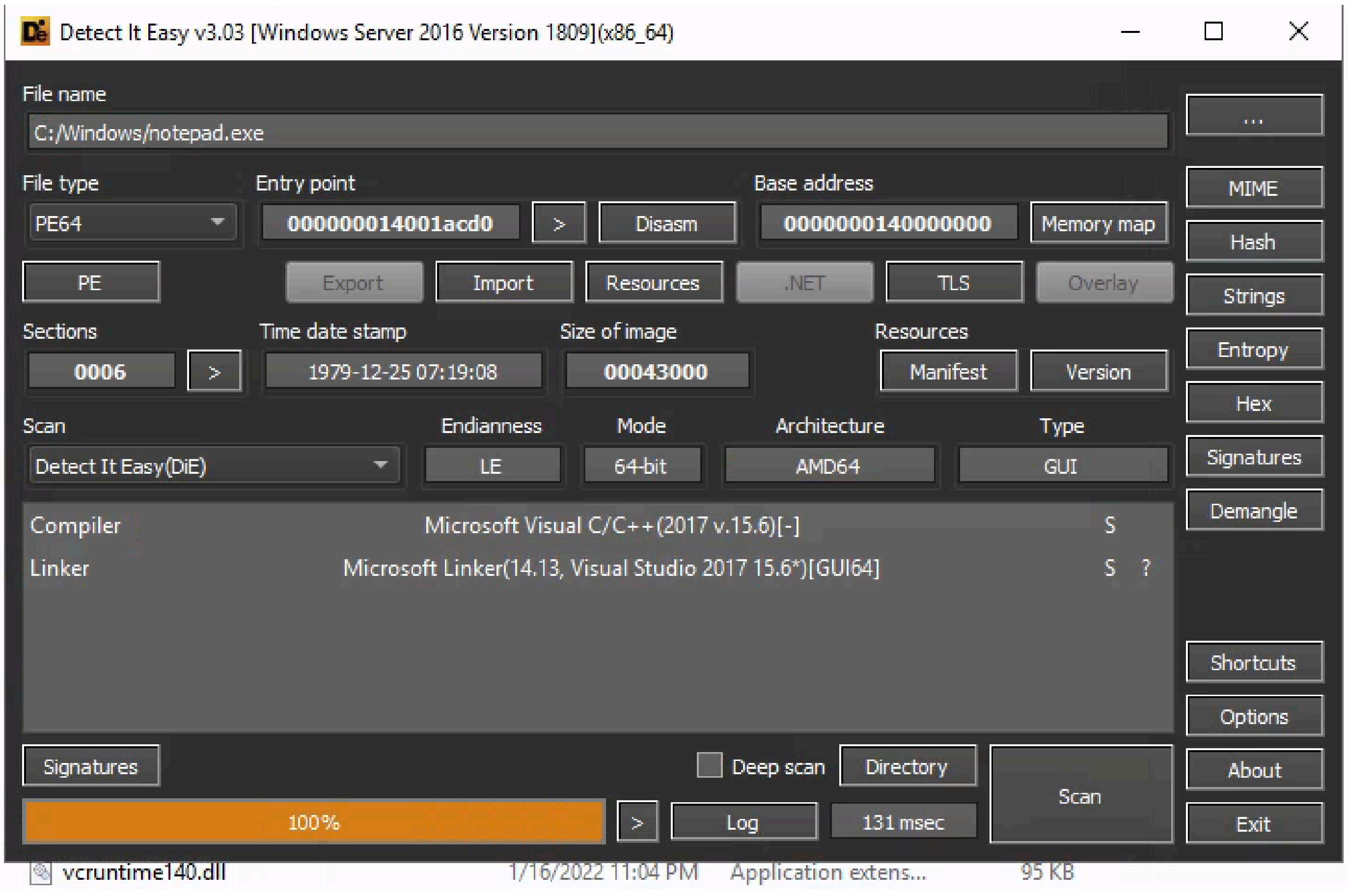
The **Section Table** will define the available sections and information in the image. As previously discussed, sections store the contents of the file, such as code, imports, and data. You can identify each section definition from the table in the hex dump section below.

```
000001F0  2E 74 65 78 74 00 00 00 D0 0B 00 00 00 10 00 00  .text...Ð.......
00000200  00 0C 00 00 00 04 00 00 00 00 00 00 00 00 00 00  ................
00000210  00 00 00 00 20 00 00 60 2E 72 64 61 74 61 00 00  .... ..`.rdata..
00000220  76 0C 00 00 00 20 00 00 00 0E 00 00 00 10 00 00  v.... ..........
00000230  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 40  ............@..@
00000240  2E 64 61 74 61 00 00 00 B8 06 00 00 00 30 00 00  .data...¸....0..
00000250  00 02 00 00 00 1E 00 00 00 00 00 00 00 00 00 00  ................
00000260  00 00 00 00 40 00 00 C0 2E 70 64 61 74 61 00 00  ....@..À.pdata..
00000270  F0 00 00 00 00 40 00 00 00 02 00 00 00 20 00 00  ð....@....... ..
00000280  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 40  ............@..@
00000290  2E 72 73 72 63 00 00 00 10 47 00 00 00 50 00 00  .rsrc....G...P..
000002A0  00 48 00 00 00 22 00 00 00 00 00 00 00 00 00 00  .H..."..........
000002B0  00 00 00 00 40 00 00 40 2E 72 65 6C 6F 63 00 00  ....@..@.reloc..
000002C0  2C 00 00 00 00 A0 00 00 00 02 00 00 00 6A 00 00  ,.... .......j..
000002D0  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 42  ............@..B
```

Now that the headers have defined the format and function of the file, the sections can define the contents and data of the file.

Section	Purpose
.text	Contains executable code and entry point
.data	Contains initialized data (strings, variables, etc.)
.rdata or .idata	Contains imports (Windows API) and DLLs.
.reloc	Contains relocation information
.rsrc	Contains application resources (images, etc.)
.debug	Contains debug information

To use DIE (Detect It Easy), we need to give the full path to the exe file.



Detect It Easy v3.03 [Windows Server 2016 Version 1809](x86_64)

File name: C:/Windows/notepad.exe

File type: PE64
Entry point: 000000014001acd0
Base address: 0000000140000000

Sections: 0006
Time date stamp: 1979-12-25 07:19:08
Size of image: 00043000

Scan: Detect It Easy(DiE)
Endianness: LE
Mode: 64-bit
Architecture: AMD64
Type: GUI

```
Compiler    Microsoft Visual C/C++(2017 v.15.6)[-]                          S
Linker      Microsoft Linker(14.13, Visual Studio 2017 15.6*)[GUI64]        S  ?
```

100% 131 msec

vcruntime140.dll   1/16/2022 11:04 PM   Application extens...   95 KB

To answer "What is the virtual address of "*.data*"?" we will need to click in the "PE" button, and that is going to open a new tab. We then go to "Sections" tab, and ".data" should be listed in there.