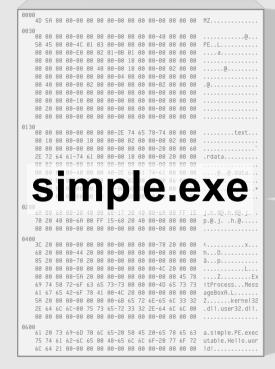
version 1, 3rd May 2012

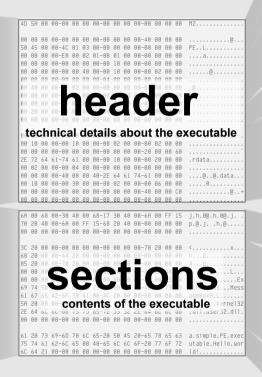


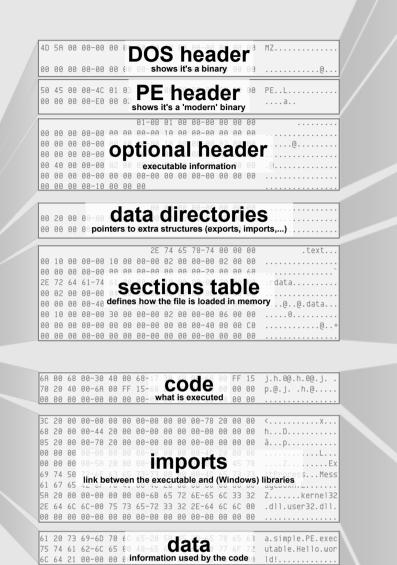
PE101 a windows executable walkthrough

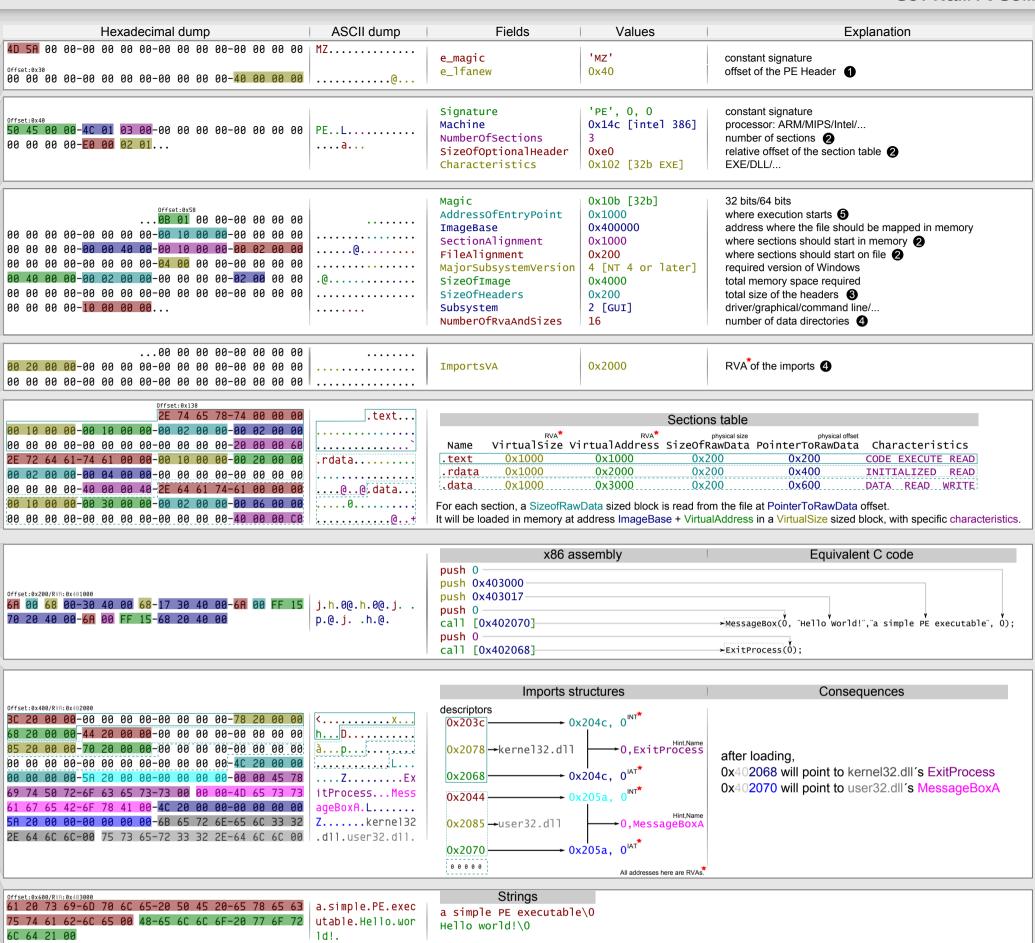
Dissected PE

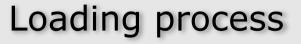










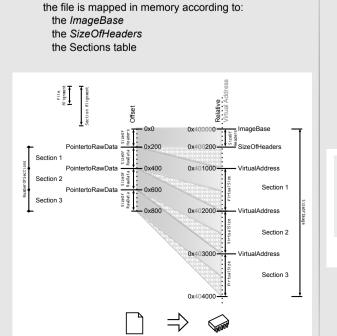


1 Headers

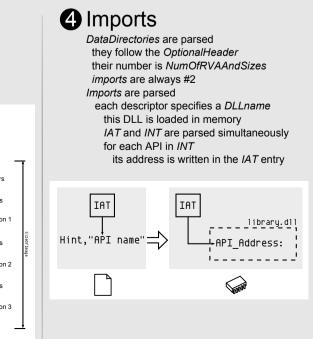
the PE Header is parsed (its offset is DOS Header's e Ifanew) (it follows the PE Header)

2 Sections table

it contains NumberOfSections elements it is checked for validity with alignments: FileAlignments and SectionAlignments



Mapping



5 Execution Code is called at the EntryPoint the calls of the code go via the IAT to the APIs a simple PE executa... Hello world! OK

This is the whole file, however, most PE files contain more elements. Explanations are simplified, for conciseness

Notes MZ HEADER aka DOS_HEADER

Starts with 'MZ' (initials of *Mark Zbikowski* MS-DOS developer) PE HEADER aka IMAGE_FILE_HEADERS / COFF file header Starts with 'PE' (Portable Executable) OPTIONAL HEADER aka IMAGE_OPTIONAL_HEADER Optional only for non-standard PEs but required for executables **RVA** Relative Virtual Address Address relative to ImageBase (at ImageBase, RVA = 0) Almost all addresses of the headers are RVAs In code, addresses are not relative. **INT** Import Name Table Null-terminated list of pointers to Hint, Name structures **IAT** Import Address Table Null-terminated list of pointers On file it is a copy of the INT After loading it points to the imported APIs Index in the exports table of a DLL to be imported Not required but provides a speed-up by reducing look-up