



[Knowledgebase](#) > [TRACE32 PowerView](#) > [Does Data.LOAD.ELF use the ELF program or section header?](#)

Does Data.LOAD.ELF use the ELF program or section header?

2024-08-07 - [Comments \(0\)](#) - [TRACE32 PowerView](#)

The ELF file includes the following sources of code information:

- ELF program headers which include:
 - Virtual address and physical address
 - Memory size and file size
- ELF section headers which include:
 - Address with size

Using options of the **Data.LOAD.ELF** command, you can control which address and size information to download the code:

- **/CODEPROG** forces loading from the program table
- **/CODESEC** loads code from the section table. This option is useful in case the linker produces a buggy program table.
- **/CODEZERO** fills target with zeros when memory size is larger than file size. This option is needed if a program header of an ELF file indicates that it is indented for more memory than within the ELF file ($p_memsz > p_filesz$) than this additional memory should be set to zero. This is usually done by the start-up code of the target application. However, with option **/CODEZERO**, the debugger fills this memory with zeroes.
- **/LOGLOAD** takes the logical address (p_vaddr) of the program table to load the code (instead of the physical address).
- **/PHYSLOAD** uses the physical address (p_paddr) of the program table to load the program.
- **/CODESEC** and **/PHYSLOAD** load the code from the section table and the translation from the program table.

In most cases, it is enough to use the default Data.LOAD.ELF setup and none of these options. Using one or multiple of these options is however necessary if the linker produces wrong information in the program or section table.