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?

2025-12-02 - 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 required when a program header of an ELF file specifies more memory than is actually present in the file (`p_memsz > p_filesz`). In this case, the additional memory region is expected to be initialized to zero. This initialization is usually performed by the start-up code of the target application.

  By default, TRACE32 does **not** clear this additional memory. In the `sYmbol.List.MAP` window, such memory ranges are marked with the information **“FILL DROP”**.

  When the `/CODEZERO` option is used, the debugger explicitly fills this memory region with zeroes. The affected ranges are then marked with **“FILL”** in the `sYmbol.List.MAP` window.
- **/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.