NAME

elf_getident – retrieve file identification data

SYNOPSIS

cc [*flag* ...] *file* ... **–lelf** [*library* ...]

#include <libelf.h>

char *elf_getident(Elf **elf*, **size_t** **ptr*);

MT-LEVEL

Unsafe

DESCRIPTION

As **elf**(3E) explains, ELF provides a framework for various classes of files, where basic objects may have 32 bits, 64 bits, etc. To accommodate these differences, without forcing the larger sizes on smaller machines, the initial bytes in an ELF file hold identification information common to all file classes. Every ELF header's **e_ident** has **EI_NIDENT** bytes with the following interpretation.

e_ident Index	Value	Purpose
EI_MAG0 **EI_MAG1** **EI_MAG2** **EI_MAG3**	**ELFMAG0** **ELFMAG1** **ELFMAG2** **ELFMAG3**	File identification
EI_CLASS	**ELFCLASSNONE** **ELFCLASS32** **ELFCLASS64**	File class
EI_DATA	**ELFDATANONE** **ELFDATA2LSB** **ELFDATA2MSB**	Data encoding
EI_VERSION	**EV_CURRENT**	File version
7-15	**0**	Unused, set to zero

Other kinds of files (see **elf_kind**(3E)) also may have identification data, though they would not conform to **e_ident**.

elf_getident() returns a pointer to the file's ''initial bytes.'' If the library recognizes the file, a conversion from the file image to the memory image may occur. In any case, the identification bytes are guaranteed not to have been modified, though the size of the unmodified area depends on the file type. If *ptr* is non-null, the library stores the number of identification bytes in the location to which *ptr* points. If no data are present, *elf* is null, or an error occurs, the return value is a null pointer, with zero stored through *ptr*, if *ptr* is non-null.

SEE ALSO

elf(3E), **elf32_getehdr**(3E), **elf_begin**(3E), **elf_kind**(3E), **elf_rawfile**(3E)