

NAME

`elf_flagelf`, `elf_flagdata`, `elf_flagehdr`, `elf_flagphdr`, `elf_flagscn`, `elf_flagshdr` – set or clear flags on ELF descriptors and components

SYNOPSIS

```
#include <libelf.h>
```

```
unsigned int elf_flagelf(Elf * elf, Elf_Cmd cmd, unsigned int flags);
unsigned int elf_flagehdr(Elf * elf, Elf_Cmd cmd, unsigned int flags);
unsigned int elf_flagphdr(Elf * elf, Elf_Cmd cmd, unsigned int flags);
unsigned int elf_flagscn(Elf_Scn * scn, Elf_Cmd cmd, unsigned int flags);
unsigned int elf_flagshdr(Elf_Scn * scn, Elf_Cmd cmd, unsigned int flags);
unsigned int elf_flagdata(Elf_Data * data, Elf_Cmd cmd, unsigned int flags);
```

DESCRIPTION

These functions allow the caller to set or clear flags on various ELF components managed by **libelf**. They are typically used to indicate that a component has been modified (dirty), or to control layout behavior during output.

elf_flagelf() modifies flags on the top-level **Elf** descriptor.

elf_flagdata() modifies flags on an **Elf_Data** descriptor, typically retrieved using **elf_getdata(3)** or created with **elf_newdata(3)**.

elf_flagehdr() modifies flags on the ELF header for a ELF descriptor (e.g., the result of **gelf_getehdr(3)**).

elf_flagphdr() modifies flags on the program header table of an ELF descriptor.

elf_flagscn() modifies flags on an **Elf_Scn** (section) descriptor.

elf_flagshdr() modifies flags on the section header of a given **Elf_Scn**.

The *cmd* argument determines whether the given flags should be set (**ELF_C_SET**) or cleared (**ELF_C_CLR**).

The following flag values may be passed via the *flags* parameter:

ELF_F_DIRTY

Marks the object as modified. This ensures changes to the component is written out during **elf_update(3)**.

ELF_F_LAYOUT

Indicates that the layout of the ELF component is managed manually. Offsets and sizes will be preserved as provided. This flag is only used with **elf_flagelf()** and is ignored when used with other functions. If this flag is set the user must set Ehdr *e_phoff*, *e_shoff*, as well as **Elf_Data** *d_off* and Shdr *sh_offset*. If not set, then these offsets will be recalculated when **elf_update** is called. If **ELF_F_LAYOUT** is set then changes to phdr offsets will prevent **elf_update** from automatically adjusting shdr offsets.

ELF_F_PERMISSIVE

Enables relaxed validation of object structure. This flag permits nonstandard layouts without strict checks. Specifically this removes a check in **elf_update()** that the section *sh_size* is a multiple of the section *sh_entsize* (if not zero). This flag is only used with **elf_flagelf()** and is ignored when

used with other functions. **ELF_F_PERMISSIVE** is an elfutils libelf extension and may not be available in other libelf implementations.

PARAMETERS

obj The object to be modified: For **elf_flagelf**, **elf_flagehdr**, **elf_flagphdr**, this is an **Elf *** descriptor. For **elf_flagscn** and **elf_flagshdr**, this is an **Elf_Scn *** descriptor. For **elf_flagdata**, an **Elf_Data *** buffer.

cmd Must be either: **ELF_C_SET** to set the given flags, or **ELF_C_CLR** to clear them.

flags Bitmask of flags to modify. May include any combination of **ELF_F_DIRTY**, **ELF_F_LAYOUT**, and **ELF_F_PERMISSIVE**.

Note that **ELF_F_LAYOUT** and **ELF_F_PERMISSIVE** are only used with **elf_flagelf()** and are ignored for other functions.

RETURN VALUE

Returns the new value of the flags after the set/clear operation.

Returns 0 on error, and sets an error code retrievable via **elf_errmsg(3)**. If the first argument is NULL then 0 is return and no error code is set.

If the last flag is cleared, 0 will be returned as the new flag value. In this case 0 is not an error indicator and no error code is set.

SEE ALSO

elf_errmsg(3), **elf_update(3)**, **libelf(3)**, **elf(5)**

ATTRIBUTES

Interface	Attribute	Value
elf_flagelf() , elf_flagehdr() , elf_flagphdr() , elf_flagscn() , elf_flagshdr() , elf_flagdata()	Thread safety	MT-Unsafe race

REPORTING BUGS

Report bugs to <elfutils-devel@sourceware.org> or <https://sourceware.org/bugzilla/>.