

NAME

gelf_getsym, **gelf_update_sym** - read and update symbol information

LIBRARY

ELF Access Library (libelf, -lelf)

SYNOPSIS

```
#include <gelf.h>
```

```
GElf_Sym *
```

```
gelf_getsym(Elf_Data *data, int ndx, GElf_Sym *sym);
```

```
int
```

```
gelf_update_sym(Elf_Data *data, int ndx, GElf_Sym *sym);
```

DESCRIPTION

These convenience functions are used to retrieve and update class-dependent *Elf32_Sym* and *Elf64_Sym* structures in an ELF object.

Argument *data* is an *Elf_Data* descriptor associated with a section of type SHT_SYMTAB, SHT_DYNSYM or SHT_GNU_versym. Argument *ndx* is the index of the symbol being retrieved or updated. The class-independent *GElf_Sym* structure is described in [gelf\(3\)](#).

Function **gelf_getsym()** retrieves class-dependent symbol information at index *ndx* in data buffer *data* and copies it to the destination pointed to by argument *sym* after translation to class-independent form.

Function **gelf_update_sym()** converts the class-independent symbol information pointed to by argument *sym* to class-dependent form, and writes it to the symbol entry at index *ndx* in the data buffer described by argument *data*. Function **gelf_update_sym()** signals an error if any of the values in the class-independent representation exceeds the representable limits of the target type.

RETURN VALUES

Function **gelf_getsym()** returns the value of argument *sym* if successful, or NULL in case of an error.

Function **gelf_update_sym()** returns a non-zero value if successful, or zero in case of an error.

ERRORS

These functions may fail with the following errors:

[ELF_E_ARGUMENT]

Arguments *data* or *sym* were NULL.

[ELF_E_ARGUMENT]

Argument *ndx* was less than zero or larger than the number of symbols in the data descriptor.

[ELF_E_ARGUMENT]

Data descriptor *data* was not associated with a section containing symbol information.

[ELF_E_RANGE]

A value was not representable in the target type.

[ELF_E_VERSION]

The *Elf_Data* descriptor denoted by argument *data* is associated with an ELF object with an unsupported version.

SEE ALSO

elf(3), elf_getdata(3), elf_getscn(3), gelf(3), gelf_getsyminfo(3), gelf_update_syminfo(3)