

NAME

gelf_getrel, **gelf_update_rel** - read and update ELF relocation entries

LIBRARY

ELF Access Library (libelf, -lelf)

SYNOPSIS

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

```
GElf_Rel *
```

```
gelf_getrel(Elf_Data *data, int ndx, GElf_Rel *rel);
```

```
int
```

```
gelf_update_rel(Elf_Data *data, int ndx, GElf_Rel *rel);
```

DESCRIPTION

These convenience functions are used to retrieve and update class-dependent *Elf32_Rel* or *Elf64_Rel* structures in an ELF object.

Argument *data* is an *Elf_Data* descriptor associated with a section of type SHT_REL. Argument *ndx* is the index of the entry being retrieved or updated. The class-independent *GElf_Rel* structure is described in gelf(3).

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

Function **gelf_update_rel**() converts the class-independent entry pointed to by argument *rel* to class-dependent form, and writes it to the entry at index *ndx* in the data buffer described by argument *data*. Function **gelf_update_rel**() 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_getrel**() returns the value of argument *rel* if successful, or NULL in case of an error.

Function **gelf_update_rel**() 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 *rel* were NULL.

[ELF_E_ARGUMENT]

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

[ELF_E_ARGUMENT]

Data descriptor *data* was not associated with a section of type SHT_REL.

[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)