NAME

elf_getdata, **elf_newdata**, **elf_rawdata** - iterate through or allocate section data

LIBRARY

ELF Access Library (libelf, -lelf)

SYNOPSIS

#include <libelf.h>

Elf_Data *
elf_getdata(*Elf_Scn *scn*, *Elf_Data *data*);

Elf_Data *
elf_newdata(*Elf_Scn *scn*);

Elf_Data *
elf_rawdata(*Elf_Scn *scn*, *Elf_Data *data*);

DESCRIPTION

These functions are used to access and manipulate data descriptors associated with section descriptors. Data descriptors used by the ELF library are described in elf(3).

Function **elf_getdata**() will return the next data descriptor associated with section descriptor *scn*. The returned data descriptor will be setup to contain translated data. Argument *data* may be NULL, in which case the function returns the first data descriptor associated with section *scn*. If argument *data* is not NULL, it must be a pointer to a data descriptor associated with section descriptor *scn*, and function **elf_getdata**() will return a pointer to the next data descriptor for the section, or NULL when the end of the section's descriptor list is reached.

Function **elf_newdata**() will allocate a new data descriptor and append it to the list of data descriptors associated with section descriptor *scn*. The new data descriptor will be initialized as follows:

d_align	Set to 1.
d_buf	Initialized to NULL.
d_off	Set to (off_t) -1. This field is under application control if the ELF_F_LAYOUT flag was set on the ELF descriptor.
d_size	Set to zero.
d_type	Initialized to ELF_T_BYTE.
d_version	Set to the current working version of the library, as set by elf_version(3).

The application must set these values as appropriate before calling elf_update(3). Section *scn* must be associated with an ELF file opened for writing. If the application has not requested full control of layout

by setting the ELF_F_LAYOUT flag on descriptor *elf*, then the data referenced by the returned descriptor will be positioned after the existing content of the section, honoring the file alignment specified in member *d_align*. On successful completion of a call to **elf_newdata**(), the ELF library will mark the section *scn* as "dirty".

Function **elf_rawdata**() is used to step through the data descriptors associated with section *scn*. In contrast to function **elf_getdata**(), this function returns untranslated data. If argument *data* is NULL, the first data descriptor associated with section *scn* is returned. If argument *data* is not NULL, is must be a data descriptor associated with section *scn*, and function **elf_rawdata**() will return the next data descriptor in the list, or NULL if no further descriptors are present. Function **elf_rawdata**() always returns *Elf_Data* structures of type ELF_T_BYTE.

Special handling of zero-sized and SHT_NOBITS sections

For sections of type SHT_NOBITS, and for zero-sized sections, the functions **elf_getdata**() and **elf_rawdata**() return a pointer to a valid *Elf_Data* structure that has its *d_buf* member set to NULL and its *d_size* member set to the size of the section.

If an application wishes to create a section of type SHT_NOBITS, it should add a data buffer to the section using function **elf_newdata**(). It should then set the *d_buf* and *d_size* members of the returned *Elf_Data* structure to NULL and the desired size of the section respectively.

RETURN VALUES

These functions return a valid pointer to a data descriptor if successful, or NULL if an error occurs.

ERRORS

These functions may fail with the following errors:

[ELF_E_ARGUMENT]
                    Either of the arguments *scn* or *data* was NULL.

[ELF_E_ARGUMENT]
                    The data descriptor referenced by argument *data* is not associated with section
                    descriptor *scn*.

[ELF_E_ARGUMENT]
                    The section denoted by argument *scn* had no data associated with it.

[ELF_E_DATA]   Retrieval of data from the underlying object failed.

[ELF_E_RESOURCE]   An out of memory condition was detected.

[ELF_E_SECTION]　　　Section *scn* had type SHT_NULL.

[ELF_E_SECTION]　　　The type of the section *scn* was not recognized by the library.

[ELF_E_SECTION]　　　The size of the section *scn* is not a multiple of the file size for its section type.

[ELF_E_SECTION]　　　The file offset for section *scn* is incorrect.

[ELF_E_UNIMPL]　　　The section type associated with section *scn* is not supported.

[ELF_E_VERSION]　　　Section *scn* was associated with an ELF object with an unsupported version.

SEE ALSO

elf(3), elf_flagdata(3), elf_flagscn(3), elf_getscn(3), elf_getshdr(3), elf_newscn(3), elf_rawfile(3),
elf_update(3), elf_version(3), gelf(3)