

**NAME**

**elftc\_bfd\_find\_target**, **elftc\_bfd\_target\_byteorder**, **elftc\_bfd\_target\_class**, **elftc\_bfd\_target\_flavor**,  
**elftc\_bfd\_target\_machine** - binary object descriptor handling

**LIBRARY**

library "libelftc"

**SYNOPSIS**

```
#include <libelftc.h>
```

```
struct Elftc_Bfd_Target;
```

```
Elftc_Bfd_Target *
```

```
elftc_bfd_find_target(const char *target_name);
```

```
unsigned int
```

```
elftc_bfd_target_class(Elftc_Bfd_Target *target);
```

```
unsigned int
```

```
elftc_bfd_target_byteorder(Elftc_Bfd_Target *target);
```

```
Elftc_Bfd_Target_Flavor
```

```
elftc_bfd_target_flavor(Elftc_Bfd_Target *target);
```

```
unsigned int
```

```
elftc_bfd_target_machine(Elftc_Bfd_Target *target);
```

```
unsigned int
```

```
elftc_bfd_target_osabi(Elftc_Bfd_Target *target);
```

**DESCRIPTION**

Function **elftc\_bfd\_find\_target**() locates a binary object descriptor corresponding to the descriptor name in argument *target\_name*. Binary object descriptors encapsulate properties of an object format such as its file representation, ELF class, and byte endianness.

Known descriptor names and their properties include:

<i>Name</i>	<i>Object Format</i>	<i>Byte Order</i>	<i>Bit Width</i>
binary	Binary	-	-
efi-app-ia32	PE	LSB	32

efi-app-x86_64	PE	LSB	64
elf32-avr	ELF	LSB	32
elf32-big	ELF	MSB	32
elf32-bigarm	ELF	MSB	32
elf32-bigmips	ELF	MSB	32
elf32-i386	ELF	LSB	32
elf32-i386-freebsd	ELF	LSB	32
elf32-ia64-big	ELF	MSB	32
elf32-little	ELF	LSB	32
elf32-littlearm	ELF	LSB	32
elf32-littlemips	ELF	LSB	32
elf32-powerpc	ELF	MSB	32
elf32-powerpc-freebsd	ELF	MSB	32
elf32-powerpcle	ELF	LSB	32
elf32-riscv	ELF	LSB	32
elf64-riscv	ELF	LSB	64
elf64-riscv-freebsd	ELF	LSB	64
elf32-sh	ELF	MSB	32
elf32-shl	ELF	LSB	32
elf32-sh-nbsd	ELF	MSB	32
elf32-shl-nbsd	ELF	LSB	32
elf32-shbig-linux	ELF	MSB	32
elf32-shl-linux	ELF	LSB	32
elf32-sparc	ELF	MSB	32
elf32-tradbigmips	ELF	MSB	32
elf32-tradlittlemips	ELF	LSB	32
elf64-alpha	ELF	LSB	64
elf64-alpha-freebsd	ELF	LSB	64
elf64-big	ELF	MSB	64
elf64-bigmips	ELF	MSB	64
elf64-ia64-big	ELF	MSB	64
elf64-ia64-little	ELF	LSB	64
elf64-little	ELF	LSB	64
elf64-littleaarch64	ELF	LSB	64
elf64-littlemips	ELF	LSB	64
elf64-powerpc	ELF	MSB	64
elf64-powerpc-freebsd	ELF	MSB	64
elf64-powerpcle	ELF	LSB	64
elf64-sh64	ELF	MSB	64
elf64-sh64l	ELF	LSB	64

elf64-sh64-nbsd	ELF	MSB	64
elf64-sh64l-nbsd	ELF	LSB	64
elf64-sh64big-linux	ELF	MSB	64
elf64-sh64-linux	ELF	LSB	64
elf64-sparc	ELF	MSB	64
elf64-sparc-freebsd	ELF	MSB	64
elf64-tradbigmips	ELF	MSB	64
elf64-tradlittlemips	ELF	LSB	64
elf64-x86-64	ELF	LSB	64
elf64-x86-64-freebsd	ELF	LSB	64
ihex	IHEX	-	-
pei-i386	PE	LSB	32
pei-x86-64	PE	LSB	64
srec	SREC	-	-
symbolsrec	SREC	-	-

Function **elftc\_bfd\_target\_byteorder()** returns the ELF byte order associated with target descriptor *target*.

Function **elftc\_bfd\_target\_class()** returns the ELF class associated with target descriptor *target*.

Function **elftc\_bfd\_target\_flavor()** returns the object format associated with target descriptor *target*. The known object formats are:

ETF_ELF	An ELF object.
ETF_BINARY	Raw binary.
ETF_IHEX	An object encoded in Intel hex format.
ETF_NONE	An unknown object format.
ETF_SREC	An object encoded as S-records.

## RETURN VALUES

Function **elftc\_bfd\_find\_target()** returns a valid pointer to an opaque binary target descriptor if successful, or NULL in case of an error.

Function **elftc\_bfd\_target\_byteorder()** returns the ELF byte order associated with the target descriptor; one of ELFDATA2MSB or ELFDATA2LSB.

Function **elftc\_bfd\_target\_class()** returns the ELF class associated with the target descriptor; one of ELFCLASS32 or ELFCLASS64.

Function **elftc\_bfd\_target\_machine()** returns the ELF architecture associated with the target descriptor.

Function **elftc\_bfd\_target\_flavor()** returns one of ETF\_BINARY, ETF\_ELF, ETF\_IHEX or ETF\_SREC if successful or ETF\_NONE in case of error.

### EXAMPLES

To return descriptor information associated with target name "elf64-big" use:

```
struct Elftc_Bfd_Target *t;

if ((t = elftc_bfd_find_target("elf64-big")) == NULL)
    errx(EXIT_FAILURE, "Cannot find target descriptor");

printf("Class: %s\n", elftc_bfd_target_class(t) == ELFCLASS32 ?
    "ELFCLASS32" : "ELFCLASS64");
printf("Byteorder: %s\n",
    elftc_bfd_target_byteorder(t) == ELFDATA2LSB ? "LSB" : "MSB");
printf("Flavor: %d\n", elftc_bfd_target_flavor(t));
```

### SEE ALSO

elf(3)