



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'scalbnf.3'***

**\$ man scalbnf.3**

SCALBLN(3) Linux Programmer's Manual SCALBLN(3)

NAME

scalbn, scalbnf, scalbni, scalbln, scalblnf, scalblni - multiply float?

ing-point number by integral power of radix

SYNOPSIS

```
#include <math.h>
```

```
double scalbln(double x, long exp);
```

```
float scalblnf(float x, long exp);
```

```
long double scalblni(long double x, long exp);
```

```
double scalbn(double x, int exp);
```

```
float scalbnf(float x, int exp);
```

```
long double scalbni(long double x, int exp);
```

Link with -lm.

Feature Test Macro Requirements for glibc (see feature\_test\_macros(7)):

```
scalbn(), scalblnf(), scalblni():
```

```
_ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
```

```
|| /* Since glibc 2.19: */ _DEFAULT_SOURCE
```

```
scalbn(), scalbnf(), scalbni():
```

```
_ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
|| /* Since glibc 2.19: */ _DEFAULT_SOURCE
|| /* Glibc versions <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

## DESCRIPTION

These functions multiply their first argument  $x$  by `FLT_RADIX` (probably 2) to the power of `exp`, that is:

$$x * FLT\_RADIX ** exp$$

The definition of `FLT_RADIX` can be obtained by including `<float.h>`.

## RETURN VALUE

On success, these functions return  $x * FLT\_RADIX ** exp$ .

If  $x$  is a NaN, a NaN is returned.

If  $x$  is positive infinity (negative infinity), positive infinity (negative infinity) is returned.

If  $x$  is +0 (-0), +0 (-0) is returned.

If the result overflows, a range error occurs, and the functions return `HUGE_VAL`, `HUGE_VALF`, or `HUGE_VALL`, respectively, with a sign the same as  $x$ .

If the result underflows, a range error occurs, and the functions return zero, with a sign the same as  $x$ .

## ERRORS

See `math_error(7)` for information on how to determine whether an error has occurred when calling these functions.

The following errors can occur:

Range error, overflow

An overflow floating-point exception (`FE_OVERFLOW`) is raised.

Range error, underflow

`errno` is set to `ERANGE`. An underflow floating-point exception (`FE_UNDERFLOW`) is raised.

## VERSIONS

These functions first appeared in glibc in version 2.1.

## ATTRIBUTES

For an explanation of the terms used in this section, see `attributes(7)`.

??

?Interface                    ? Attribute    ? Value    ?

??

?scalbn(), scalbnf(), scalbnl(),    ? Thread safety ? MT-Safe ?

?scalbln(), scalblnf(), scalblnl() ?                    ?                    ?

??

CONFORMING TO

C99, POSIX.1-2001, POSIX.1-2008.

NOTES

These functions differ from the obsolete functions described in scalb(3) in the type of their second argument. The functions described on this page have a second argument of an integral type, while those in scalb(3) have a second argument of type double.

If FLT\_RADIX equals 2 (which is usual), then scalbn() is equivalent to ldexp(3).

BUGS

Before glibc 2.20, these functions did not set errno for range errors.

SEE ALSO

ldexp(3), scalb(3)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.