



Linux Ubuntu 26.04 Manual Pages on command 'strfmon_1.3'

\$ man strfmon_1.3

strfmon(3) Library Functions Manual strfmon(3)

NAME

strfmon, strfmon_l - convert monetary value to a string

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <monetary.h>

ssize_t strfmon(size_t size;
               char s[restrict size], size_t size,
               const char *restrict format, ...);

ssize_t strfmon_l(size_t size;
                 char s[restrict size], size_t size, locale_t l,
                 const char *restrict format, ...);
```

DESCRIPTION

The `strfmon()` function formats the specified monetary amount according to the current locale and format specification `format` and places the result in the character array `s` of size `size` bytes.

The `strfmon_l()` function performs the same task, but uses the locale specified by `l`. The behavior of `strfmon_l()` is undefined if `l` is the special locale object `LC_GLOBAL_LOCALE` (see `duplocale(3)`) or is not a valid locale object handle.

Ordinary characters in `format` are copied to `s` without conversion. Conversion specifiers are introduced by a `'%'` character. Immediately following it there can be zero or more of the following flags:

- =f The single-byte character f is used as the numeric fill character (to be used with a left precision, see below). When not specified, the space character is used.
- ^ Do not use any grouping characters that might be defined for the current locale. By default, grouping is enabled.
- (or + The (flag indicates that negative amounts should be enclosed between parentheses. The + flag indicates that signs should be handled in the default way, that is, amounts are preceded by the locale's sign indication, for example, nothing for positive, "-" for negative.
- ! Omit the currency symbol.
- Left justify all fields. The default is right justification.

Next, there may be a field width: a decimal digit string specifying a minimum field width in bytes. The default is 0. A result smaller than this width is padded with spaces (on the left, unless the left-justify flag was given).

Next, there may be a left precision of the form "#" followed by a decimal digit string. If the number of digits left of the radix character is smaller than this, the representation is padded on the left with the numeric fill character. Grouping characters are not counted in this field width.

Next, there may be a right precision of the form "." followed by a decimal digit string. The amount being formatted is rounded to the specified number of digits prior to formatting. The default is specified in the `frac_digits` and `int_frac_digits` items of the current locale. If the right precision is 0, no radix character is printed. (The radix character here is determined by `LC_MONETARY`, and may differ from that specified by `LC_NUMERIC`.)

Finally, the conversion specification must be ended with a conversion character. The three conversion characters are

- % (In this case, the entire specification must be exactly "%%".) Put a '%' character in the result string.
- i One argument of type double is converted using the locale's international currency format.
- n One argument of type double is converted using the locale's national currency format.

RETURN VALUE

The `strfmon()` function returns the number of characters placed in the array `s`, not including the terminating null byte, provided the string, including the terminating null byte, fits.

Otherwise, it sets `errno` to `E2BIG`, returns -1, and the contents of the array is undefined.

ATTRIBUTES

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

[illegible]

?	Interface	?	Attribute	?	Value	?
---	-----------	---	-----------	---	-------	---

[illegible]

? strfmon() ? Thread safety ? MT-Safe locale ?

[illegible]

? strfmon l()	? Thread safety	? MT-Safe	?
---------------	-----------------	-----------	---

[illegible]

STANDARDS

POSIX.1-2008.

HISTORY

POSIX.1-2001.

EXAMPLES

The call

```
strfmon(buf, sizeof(buf), "[%^=*#6n] [%=*#6i]",
```

```
1234.567, 1234.567);
```

outputs

[? **1234,57] [EUR **1 234,57]

in the nl_NL locale. The de_DE, de_CH, en_AU, and en_GB locales yield

```
[ **1234,57 ?] [ **1.234,57 EUR]
```

[Fr. **1234.57] [CHF **1'234.57]

[\$**1234.57] [AUD**1,234.57]

[**1234.57] [GBP**1,234.57]

SEE ALSO

duplocale(3), setlocale(3), sprintf(3), locale(7)

Linux man-pages 6.17

2026-02-08

strfmon(3)