



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

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'strfromf.3' command

\$ man strfromf.3

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

NAME

strfromd, strfromf, strfroml - convert a floating-point value into a string

SYNOPSIS

```
#include <stdlib.h>
```

```
int strfromd(char *restrict str, size_t n,  
             const char *restrict format, double fp);
```

```
int strfromf(char *restrict str, size_t n,  
             const char *restrict format, float fp);
```

```
int strfroml(char *restrict str, size_t n,  
             const char *restrict format, long double fp);
```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

```
strfromd(), strfromf(), strfroml():  
    __STDC_WANT_IEC_60559_BFP_EXT__
```

DESCRIPTION

These functions convert a floating-point value, `fp`, into a string of characters, `str`, with a configurable format string. At most `n` characters are stored into `str`.

The terminating null byte ('\0') is written if and only if `n` is sufficiently large, otherwise the written string is truncated at `n` characters.

The `strfromd()`, `strfromf()`, and `strfroml()` functions are equivalent to

?	Interface	?	Attribute	?	Value	?
?	?	?	Thread safety	?	MT-Safe locale	?
?	strfromd()	?	?	?	?	?
?	strfromf()	?	Asynchronous signal safety	?	AS-Unsafe heap	?
?	strfroml()	?	?	?	?	?
?	?	?	Asynchronous cancellation safety	?	AC-Unsafe mem	?

Note: these attributes are preliminary.

CONFORMING TO

C99, ISO/IEC TS 18661-1.

NOTES

The strfromd(), strfromf(), and strfroml() functions take account of the LC_NUMERIC category of the current locale.

EXAMPLES

To convert the value 12.1 as a float type to a string using decimal notation, resulting in "12.100000":

```
#define __STDC_WANT_IEC_60559_BFP_EXT__
#include <stdlib.h>
int ssize = 10;
char s[ssize];
strfromf(s, ssize, "%f", 12.1);
```

To convert the value 12.3456 as a float type to a string using decimal notation with two digits of precision, resulting in "12.35":

```
#define __STDC_WANT_IEC_60559_BFP_EXT__
#include <stdlib.h>
int ssize = 10;
char s[ssize];
strfromf(s, ssize, "%.2f", 12.3456);
```

To convert the value 12.345e19 as a double type to a string using scientific notation with zero digits of precision, resulting in "1E+20":

```
#define __STDC_WANT_IEC_60559_BFP_EXT__
#include <stdlib.h>
```

```
int ssize = 10;
char s[ssize];
strfromd(s, ssize, "%.E", 12.345e19);
```

SEE ALSO

atof(3), snprintf(3), strtod(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/>.

GNU 2019-03-06 STRFROMD(3)