



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

\$ man strfromd.3

strfromd(3) Library Functions Manual strfromd(3)

NAME

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

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

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

```
int strfromd(size_t n;
```

```
    char str[restrict n], size_t n,
```

```
    const char *restrict format, double fp);
```

```
int strfromf(size_t n;
```

```
    char str[restrict n], size_t n,
```

```
    const char *restrict format, float fp);
```

```
int strfroml(size_t n;
```

```
    char str[restrict n], 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, other?

[illegible]

Note: these attributes are preliminary.

STANDARDS

ISO/IEC TS 18661-1.

VERSIONS

strfrommd()

strfromf()

strfroml()

glibc 2.25.

NOTES

These 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)

Linux man-pages 6.17

2026-02-08

strfromd(3)