



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 'mbrtowc.3p' command

\$ man mbrtowc.3p

MBRTOWC(3P) POSIX Programmer's Manual MBRTOWC(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

mbrtowc ? convert a character to a wide-character code (restartable)

SYNOPSIS

```
#include <wchar.h>

size_t mbrtowc(wchar_t *restrict pwc, const char *restrict s,
               size_t n, mbstate_t *restrict ps);
```

DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

If `s` is a null pointer, the `mbrtowc()` function shall be equivalent to the call:

```
mbrtowc(NULL, "", 1, ps)
```

In this case, the values of the arguments `pwc` and `n` are ignored.

If `s` is not a null pointer, the `mbrtowc()` function shall inspect at most `n` bytes beginning at the byte pointed to by `s` to determine the

number of bytes needed to complete the next character (including any shift sequences). If the function determines that the next character is completed, it shall determine the value of the corresponding wide character and then, if `pwc` is not a null pointer, shall store that value in the object pointed to by `pwc`. If the corresponding wide character is the null wide character, the resulting state described shall be the initial conversion state.

If `ps` is a null pointer, the `mbrtowc()` function shall use its own internal `mbstate_t` object, which shall be initialized at program start-up to the initial conversion state. Otherwise, the `mbstate_t` object pointed to by `ps` shall be used to completely describe the current conversion state of the associated character sequence. The implementation shall behave as if no function defined in this volume of POSIX.1-2017 calls `mbrtowc()`.

The behavior of this function is affected by the `LC_CTYPE` category of the current locale.

The `mbrtowc()` function need not be thread-safe if called with a NULL `ps` argument.

The `mbrtowc()` function shall not change the setting of `errno` if successful.

RETURN VALUE

The `mbrtowc()` function shall return the first of the following that applies:

0 If the next `n` or fewer bytes complete the character that corresponds to the null wide character (which is the value stored).

between 1 and `n` inclusive

If the next `n` or fewer bytes complete a valid character (which is the value stored); the value returned shall be the number of bytes that complete the character.

$(\text{size_t})-2$ If the next `n` bytes contribute to an incomplete but potentially valid character, and all `n` bytes have been processed (no value is stored). When `n` has at least the value of the

{MB_CUR_MAX} macro, this case can only occur if s points at a sequence of redundant shift sequences (for implementations with state-dependent encodings).

(size_t)-1 If an encoding error occurs, in which case the next n or fewer bytes do not contribute to a complete and valid character (no value is stored). In this case, [EILSEQ] shall be stored in errno and the conversion state is undefined.

ERRORS

The mbrtowc() function shall fail if:

EILSEQ An invalid character sequence is detected. In the POSIX locale an [EILSEQ] error cannot occur since all byte values are valid characters.

The mbrtowc() function may fail if:

EINVAL ps points to an object that contains an invalid conversion state.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

mbsinit(), mbsrtowcs()

The Base Definitions volume of POSIX.1?2017, <wchar.h>

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the

event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

MBRTOWC(3P)