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 'mbtowc.3p' command

$ man mbtowc.3p

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

mbtowc ? convert a character to a wide-character code

SYNOPSIS

#include <stdlib.h>

int mbtowc(wchar_t *restrict pwc, const char *restrict s, size_t n);

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 not a null pointer, mbtowc() shall determine the number of bytes that constitute the character pointed to by s. It shall then de? termine the wide-character code for the value of type wchar_t that cor? responds to that character. (The value of the wide-character code cor? responding to the null byte is 0.) If the character is valid and pwc is not a null pointer, mbtowc() shall store the wide-character code in the object pointed to by pwc.

The behavior of this function is affected by the LC_CTYPE category of the current locale. For a state-dependent encoding, this function is placed into its initial state by a call for which its character pointer argument, s, is a null pointer. Subsequent calls with s as other than a null pointer shall cause the internal state of the function to be al? tered as necessary. A call with s as a null pointer shall cause this function to return a non-zero value if encodings have state dependency, and 0 otherwise. If the implementation employs special bytes to change the shift state, these bytes shall not produce separate wide-character codes, but shall be grouped with an adjacent character. Changing the LC_CTYPE category causes the shift state of this function to be unspec? ified. At most n bytes of the array pointed to by s shall be examined.

The implementation shall behave as if no function defined in this vol? ume of POSIX.1?2017 calls mbtowc().

The mbtowc() function need not be thread-safe.

RETURN VALUE

If s is a null pointer, mbtowc() shall return a non-zero or 0 value, if character encodings, respectively, do or do not have state-dependent encodings. If s is not a null pointer, mbtowc() shall either return 0 (if s points to the null byte), or return the number of bytes that con? stitute the converted character (if the next n or fewer bytes form a valid character), or return -1 and shall set errno to indicate the er? ror (if they do not form a valid character).

In no case shall the value returned be greater than n or the value of the {MB_CUR_MAX} macro.

ERRORS

The mbtowc() 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 following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

mblen(), mbstowcs(), wctomb(), wcstombs()

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