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 'mbsrtowcs.3p' command

$ man mbsrtowcs.3p

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

mbsnrtowcs, mbsrtowcs ? convert a character string to a wide-character string (restartable)

SYNOPSIS

```
#include <wchar.h>
size_t mbsnrtowcs(wchar_t *restrict dst, const char **restrict src,
    size_t nmc, size_t len, mbstate_t *restrict ps);
size_t mbsrtowcs(wchar_t *restrict dst, const char **restrict src,
    size_t len, mbstate_t *restrict ps);
```

DESCRIPTION

For mbsrtowcs(): 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.

The mbsrtowcs() function shall convert a sequence of characters, begin? ning in the conversion state described by the object pointed to by ps, from the array indirectly pointed to by src into a sequence of corre?

sponding wide characters. If dst is not a null pointer, the converted characters shall be stored into the array pointed to by dst. Conver? sion continues up to and including a terminating null character, which shall also be stored. Conversion shall stop early in either of the fol? lowing cases:

* A sequence of bytes is encountered that does not form a valid char? acter.
* len codes have been stored into the array pointed to by dst (and dst is not a null pointer).

Each conversion shall take place as if by a call to the mbrtowc() func? tion.

If dst is not a null pointer, the pointer object pointed to by src shall be assigned either a null pointer (if conversion stopped due to reaching a terminating null character) or the address just past the last character converted (if any). If conversion stopped due to reach? ing a terminating null character, and if dst is not a null pointer, the resulting state described shall be the initial conversion state.

If ps is a null pointer, the mbsrtowcs() function shall use its own in? ternal mbstate_t object, which is 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 mbsnrtowcs() function shall be equivalent to the mbsrtowcs() func? tion, except that the conversion of characters indirectly pointed to by src is limited to at most nmc bytes (the size of the input buffer), and under conditions where mbsrtowcs() would assign the address just past the last character converted (if any) to the pointer object pointed to by src, mbsnrtowcs() shall instead assign the address just past the last byte processed (if any) to that pointer object. If the input buf? fer ends with an incomplete character, it is unspecified whether con? version stops at the end of the previous character (if any), or at the end of the input buffer. In the latter case, a subsequent call to mb? snrtowcs() with an input buffer that starts with the remainder of the

incomplete character shall correctly complete the conversion of that character.

The behavior of these functions shall be affected by the LC_CTYPE cate? gory of the current locale.

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

The mbsnrtowcs() and mbsrtowcs() functions need not be thread-safe if called with a NULL ps argument.

The mbsrtowcs() function shall not change the setting of errno if suc? cessful.

RETURN VALUE

If the input conversion encounters a sequence of bytes that do not form a valid character, an encoding error occurs. In this case, these func? tions shall store the value of the macro [EILSEQ] in errno and shall return (size_t)-1; the conversion state is undefined. Otherwise, these functions shall return the number of characters successfully converted, not including the terminating null (if any).

ERRORS

These functions 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.

These functions 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

A future version may require that when the input buffer ends with an incomplete character, conversion stops at the end of the input buffer.

SEE ALSO

iconv(), mbrtowc(), mbsinit()

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

COPYRIGHT