

**NAME**

**wcsrtombs**, **wcsnrtombs** - convert a wide-character string to a character string (restartable)

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

**#include <wchar.h>**

*size\_t*

**wcsrtombs**(*char \* restrict dst*, *const wchar\_t \*\* restrict src*, *size\_t len*, *mbstate\_t \* restrict ps*);

*size\_t*

**wcsnrtombs**(*char \* restrict dst*, *const wchar\_t \*\* restrict src*, *size\_t nwc*, *size\_t len*,  
*mbstate\_t \* restrict ps*);

**DESCRIPTION**

The **wcsrtombs**() function converts a string of wide characters indirectly pointed to by *src* to a corresponding multibyte character string stored in the array pointed to by *dst*. No more than *len* bytes are written to *dst*.

If *dst* is NULL, no characters are stored.

If *dst* is not NULL, the pointer pointed to by *src* is updated to point to the character after the one that conversion stopped at. If conversion stops because a null character is encountered, *\*src* is set to NULL.

The *mbstate\_t* argument, *ps*, is used to keep track of the shift state. If it is NULL, **wcsrtombs**() uses an internal, static *mbstate\_t* object, which is initialized to the initial conversion state at program startup.

The **wcsnrtombs**() function behaves identically to **wcsrtombs**(), except that conversion stops after reading at most *nwc* characters from the buffer pointed to by *src*.

**RETURN VALUES**

The **wcsrtombs**() and **wcsnrtombs**() functions return the number of bytes stored in the array pointed to by *dst* (not including any terminating null), if successful, otherwise it returns (*size\_t*)-1.

**ERRORS**

The **wcsrtombs**() and **wcsnrtombs**() functions will fail if:

[EILSEQ]           An invalid wide character was encountered.

[EINVAL]           The conversion state is invalid.

**SEE ALSO**

mbsrtowcs(3), wctomb(3), wcstombs(3)

**STANDARDS**

The **wcsrtombs()** function conforms to ISO/IEC 9899:1999 ("ISO C99").

The **wcsnrtombs()** function is an extension to the standard.