NAME

wcstod, wstod, watof – convert wide character string to double-precision number

SYNOPSIS

cc [*flag* ...] *file* ... **–lw** [*library* ...]

#include <wchar.h>

double wcstod(const wchar_t **nptr,* **wchar_t** ***endptr***);**
double wstod(const wchar_t **nptr,* **wchar_t** ***endptr***);**
double watof(wchar_t **nptr***);**

MT-LEVEL

MT-Safe

DESCRIPTION

The **wcstod( )** and **wstod( )** functions convert the initial portion of the wide character string pointed to by *nptr* to **double** representation. They first decompose the input wide character string into three parts: an initial, possibly empty, sequence of white-space wide character codes (as specified by **iswspace**(3I)); a subject sequence interpreted as a floating-point constant; and a final wide-character string of one or more unrecognised wide-character codes, including the terminating null wide character code of the input wide character string. They then attempt to convert the subject sequence to a floating-point number, and return the result.

The expected form of the subject sequence is an optional '+' or '–' sign, then a non-empty sequence of digits optionally containing a radix, then an optional exponent part. An exponent part consists of 'e' or 'E', followed by an optional sign, followed by one or more decimal digits. The subject sequence is defined as the longest initial subsequence of the input wide character string, starting with the first non-white-space wide-character code, that is of the expected form. The subject sequence contains no wide-character codes if the input wide character string is empty or consists entirely of white-space wide-character codes, or if the first wide-character code that is not white space other than a sign, a digit or a radix.

If the subject sequence has the expected form, the sequence of wide-character codes starting with the first digit or the radix (whichever occurs first) is interpreted as a floating constant as defined in the C language, except that the radix is used in place of a period, and that if neither an exponent part nor a radix appears, a radix is assumed to follow the last digit in the wide character string. If the subject sequence begins with a minus sign (-), the value resulting from the conversion is negated. A pointer to the final wide character string is stored in the object pointed to by *endptr*, provided that *endptr* is not a null pointer.

The radix is defined in the program's locale (category **LC_NUMERIC**). In the POSIX locale, or in a locale where the radix is not defined, the radix defaults to a period (.).

In other than the POSIX locale, other implementation-dependent subject sequence forms may be accepted.

If the subject sequence is empty or does not have the expected form, no conversion is performed; the value of *nptr* is stored in the object pointed to by *endptr*, provided that *endptr* is not a null pointer.

watof(str) is equivalent to **wstod(str, (wchar_t **)NULL)**.

RETURN VALUES

wcstod( ) and **wstod( )** return the converted value, if any. If no conversion could be performed, **0** is returned, and **errno** may be set to **EINVAL**.

If the correct value is outside the range of representable values, ±**HUGE_VAL** is returned (according to the sign of the value), and **errno** is set to **ERANGE**.

If the correct value would cause underflow, **0** is returned, and **errno** is set to **ERANGE**.

ERRORS

wcstod( ) and **wstod( )** will fail if:

ERANGE The value to be returned would cause overflow or underflow.

wcstod( ) and **wcstod( )** may fail if:

EINVAL No conversion could be performed.

SEE ALSO

iswspace(3I), **localeconv**(3C), **scanf**(3S), **setlocale**(3C), **wcstol**(3I)

NOTES

Because **0** is returned on error and is also a valid return on success, an application wishing to check for error situations should set **errno** to **0**, call **wcstod( )** or **wstod( )**, then check **errno** and if it is non-zero, assume an error has occurred.