std::wctomb

Defined in header <cstdlib>

```
int wctomb( char *s, wchar_t wc );
```

Converts a wide character wc to multibyte encoding and stores it (including any shift sequences) in the char array whose first element is pointed to by s. No more than `MB_CUR_MAX` characters are stored. The conversion is affected by the current locale's LC_CTYPE category.

If wc is the null character, the null byte is written to s, preceded by any shift sequences necessary to restore the initial shift state.

If s is a null pointer, resets the global conversion state and determines whether shift sequences are used.

Parameters

s	-	pointer to the character array for output
wc	-	wide character to convert

Return value

If s is not a null pointer, returns the number of bytes that are contained in the multibyte representation of wc or `-1` if wc is not a valid character.

If s is a null pointer, resets its internal conversion state to represent the initial shift state and returns `0` if the current multibyte encoding is not state-dependent (does not use shift sequences) or a non-zero value if the current multibyte encoding is state-dependent (uses shift sequences).

Notes

Each call to wctomb updates the internal global conversion state (a static object of type std::mbstate_t, only known to this function). If the multibyte encoding uses shift states, this function is not reentrant. In any case, multiple threads should not call wctomb without synchronization: std::wcrtomb may be used instead.

Example

Run this code

```
#include <iostream>
#include <iomanip>
#include <clocale>
#include <string>
#include <cstdlib>

void print_wide(const std::wstring& wstr)
{
    bool shifts = std::wctomb(nullptr, 0); // reset the conversion state
    std::cout << "shift sequences are " << (shifts ? "" : "not" )
              << " used\n" << std::uppercase << std::setfill('0');
    for (const wchar_t wc : wstr) {
        std::string mb(MB_CUR_MAX, '\0');
        const int ret = std::wctomb(&mb[0], wc);
        const char* s = ret > 1 ? "s" : "";
        std::cout << "multibyte char '" << mb << "' is " << ret
                  << " byte" << s << ": [" << std::hex;
        for (int i{0}; i != ret; ++i) {
            const int c = 0xFF & mb[i];
            std::cout << (i ? " " : "") << std::setw(2) << c;
        }
        std::cout << "]\n" << std::dec;
    }
}

int main()
{
    std::setlocale(LC_ALL, "en_US.utf8");
    // UTF-8 narrow multibyte encoding
    std::wstring wstr = L"z\u00df\u6c34\U0001d10b"; // or L"zß水𝄋"
    print_wide(wstr);
}
```

Output:

```
shift sequences are not used
multibyte char 'z' is 1 byte: [7A]
multibyte char 'ß' is 2 bytes: [C3 9F]
multibyte char '水' is 3 bytes: [E6 B0 B4]
multibyte char '𝄋' is 4 bytes: [F0 9D 84 8B]
```

See also

mbtowc	converts the next multibyte character to wide character (function)
wcrtomb	converts a wide character to its multibyte representation, given state (function)
do_out [virtual]	converts a string from internT to externT, such as when writing to file (virtual protected member function of std::codecvt<InternT,ExternT,State>)
C documentation for **wctomb**	

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/string/multibyte/wctomb&oldid=133936"