cppreference.com

Log in

Search

View Edit History

Page Discussion

C++ Strings library Null-terminated wide strings

std::towlower

```
Defined in header <cwctype>
std::wint_t towlower( std::wint_t ch );
```

Converts the given wide character to lowercase, if possible.

If the value of [ch] is neither representable as a wchar_t nor equal to the value of the macro [WEOF], the behavior is undefined.

Parameters

ch - wide character to be converted

Return value

Lowercase version of ch or unmodified ch if no lowercase version is listed in the current C locale.

Notes

Only 1:1 character mapping can be performed by this function, e.g. the Greek uppercase letter $^{\circ}\Sigma^{\circ}$ has two lowercase forms, depending on the position in a word: ' σ ' and ' ς '. A call to std::towlower cannot be used to obtain the correct lowercase form in this case. specifies which pairs of Unicode characters are included in this mapping.

Example

```
#include <clocale>
#include <cwctype>
#include <iostream>
int main()
    wchar_t c = L'\u0190'; // Latin capital open E ('E')
    std::cout << std::hex << std::showbase;
std::cout << "in the default locale, towlower("</pre>
                << static_cast<std::wint_t>(c) << ") = "
                << std::towlower(c) << '\n';
    std::setlocale(LC_ALL, "en_US.utf8");
    std::cout << "in Unicode locale, towlower("</pre>
                << static_cast<std::wint_t>(c) << ") = "
                << std::towlower(c) << '\n';
}
```

Output:

```
in the default locale, towlower(0x190) = 0x190
in Unicode locale, towlower(0x190) = 0x25b
```

See also

towupper	converts a wide character to uppercase (function)
tolower(std::locale)	converts a character to lowercase using the <code>ctype</code> facet of a locale (function template)
tolower	converts a character to lowercase (function)

C documentation for towlower

```
Support us Recent changes FAQ Offline version
```

What links here Related changes Upload file Special pages Printable version Permanent link Page information

In other languages Deutsch Español Français Italiano 日本語 Português Русский 中文

This page was last modified on 15 September 2023, at 09:27.

Privacy policy About cppreference.com Disclaimers





