

# The default C locale is not a very interesting one

 devblogs.microsoft.com/oldnewthing/20250206-00

February 6, 2025



Although the C and C++ languages provide facilities for localization, the default locale is the so-called “C” locale, which barely understands anything.

In the “C” locale, the uppercase characters are “A” through “Z”; the lowercase characters are “a” through “z”, the decimal separator is “.”, and there is no thousands separator.

The “C” locale is designed to be minimal. But it also means that unless you’ve taken special efforts to change your process’s locale to something else, functions like `towupper` and `_wcslwr` produce only extremely rudimentary results. All they know is the characters in the 7-bit ASCII set. They don’t even know that the uppercase version of ä is Ä.

Support for any locales beyond the “C” locale is implementation-defined, and the standard considers it a quality of implementation issue. Microsoft’s Visual C++ compiler uses BCP47 for locale names, like `sr-Cyrl-BA` for “Serbian, Cyrillic script, as used in Bosnia and Herzegovina.” The gcc library appears to use a custom format, such as `de_AT.iso885915@euro` for “German, as used in Austria, using the ISO-8859-15 character set and the Euro as the currency.”

This means that if you just dive in and call `tolower` without doing any locale preparation, all you’re going to get support for is characters U+0041 (LATIN CAPITAL LETTER A) through U+005A (LATIN CAPITAL LETTER Z) mapping to U+0061 (LATIN SMALL LETTER A) through U+007A (LATIN SMALL LETTER Z).

The Microsoft Visual C++ compiler standard library comes with bonus functions like `_strlwr` and `_wcslwr` for converting strings to lowercase. By default, these follow the current C runtime locale, so again, if you don’t do any locale preparation, you’re going to get the naïve case mapping.

```
wchar_t example[] = L"\x00C0" L"BC"; // ÀBC
_wcslwr_s(example); // Result: Àbc
```

Next time, we'll look at how to get `_wcslwr` to operate on more interesting locales than the C locale.