

## Proposal for C2X WG14 N2490

**Title:** Why no wide string `strfrom` functions – updates N2475  
**Author, affiliation:** C FP group  
**Date:** 2020-02-21  
**Proposal category:** Editorial  
**Reference:** N2400, N2454, N2475

### Problem:

There are wide string analogs for the `strtod` function family, but not the `strfromd` family. It would be helpful to say how to easily achieve conversions to wide strings, and hence why additional functions to do the job are not provided.

N2400 suggested adding a paragraph in 7.29.4.1. At its fall 2019 meeting, WG14 requested this be redone as a footnote with a recipe for wide string analogs of `strfromd`.

Where would be a good place to attach the footnote? It pertains specifically to 7.29.4.1, but there is no text there, and footnotes aren't attached to subclause titles (that we see).

The suggested change below adds an introductory sentence in 7.29.4.1 and attaches the footnote to the new text. Note that other subclauses of 7.29.4 don't reference the single-byte versions this way.

We also considered attaching the footnote to the first sentence of 7.29.4, but that seemed less helpful to the reader.

### Suggested change:

In 7.29.4.1, insert the paragraph:

[1] This subclause describes wide string analogs of the `strtod` family of functions (7.22.1.5, 7.22.1.6).

In 7.29.4.1, attach a footnote to the wording:

the `strtod` family of functions (7.22.1.5, 7.22.1.6)

where the footnote is:

[1] Wide string analogs of the `strfromd` family of functions. (7.22.1.3, 7.22.1.4) are not provided because those conversions can be done by using `mbstowcs` (7.22.8.1) to convert the result of `strfromd`, `strfromf`, etc., to wide string. For example, the following converts `double d` to wide string `ws`

with at most **n**-1 non-null wide characters, using style **g** formatting, and computes the number **nc** of wide characters that would have been written had **n** been sufficiently large, not counting the terminating null wide character.

```
#include <stdlib.h>
const size_t n = 20;
double d;
...
// convert d to single-byte character string s
char s[n];
int nc = strfromd(s, n, "%g", d);
// convert s (regarded as a multi-byte character
// string) to wide string ws
wchar_t ws[n];
(void)mbstowcs(ws, s, n);
```