

Doc. No.: WG14/N1263
Date: 2007-09-10
Project: Programming Language C (TR 24732)
Subject: Comments on N1241

The following is a collection of comments on the Decimal TR document N1241.

1. Section 9.2 specifies five macros for rounding modes. Can an implementation support additional rounding modes for decimal float, as it can for generic floating point in C99 section 7.6 paragraph 7? This could be seen either as a new issue or as an omission that should obviously be fixed.
2. Page 13, 7.1.1 TTDT, item "1" and "4": "unsuffixed floating constant" -> "unsuffixed decimal floating constant". TTDT makes no sense for unsuffixed hexadecimal floating constants, and rest of paragraph 1 is talking about decimal digits. This would also apply to the Suggested changes to C99 on page 15.
3. Page 17, 8.2 Functions, `ldexp` and `frexp` are missing from the `<math.h>` list.
4. Page 27, `ldexp`: Is the quantum of a DFP result implementation defined?
5. Page 31, 9.6 `strtodNN`. It is confusing to have "n-char-sequence" have two different meanings. Better would be to use a new name (such as "d-char-sequence") for the decimal functions.

Item [5]: Should add that rounding happens after negation. Also, should swap items [5] and [6].
6. Page 34, 9.7 `wcstodNN`. It is confusing to have "n-wchar-sequence" have two different meanings. Better would be to use a new name (such as "d-wchar-sequence") for the decimal functions.

Item [5]: Should add that rounding happens after negation. Also, should swap items [5] and [6].
7. Page 35, 9.8 `<tgmath.h>`: I am under the impression that mixing binary FP and decimal FP is a no-no; I do not see any such constraint. That is, `pow(3.f, 5.DD)` should be invalid.

Comments from Abstain ballot from Fred Tydeman:

1. Printed Page 27, Replace 7.12.6.6#2 item. Need to change that text so that it mentions `_x_` being a decimal floating-point number, then it is multiplied by a power of 10, otherwise, by a power of 2.

That is in keeping with the change to `_frexp_`.
2. Printed Page 26, Add to end of 7.12.14#1 item. There is no mention of how an implementation's `_isgreater_` macro is supposed to tell a 32-bit float from a 32-bit `_Decimal32`, nor a 64-bit double from a 64-bit `_Decimal64`. This issue was raised at the last WG14 meeting. It even had its own paper (by an implementor).
3. Printed Page 19, 7.12.3 mentioned in passing in section 9.3. Where is the text of the change to C99 to have the 7.12.3

Classification macros work with decimal floating-point? How is an implementation supposed to do this? It is the same issue as 7.12.14. The Rationale mentions 'compiler magic'.