Reply from Joseph Myers to an email from Clive Pygott, regarding GB comments on the FP annex pt1

On Mon, 30 Sep 2013, clive@pygott.demon.co.uk wrote:

> I'm at the WG14 meeting in Chicago, and we are currently discussing the
> responses to the UK comments on the floating point annex. Is this
> something that has been discussed with you (attached)?

I haven't generally so far studied the documents postdating the pre-meeting mailing. But some initial comments on this:

(scope description) This was an issue of Nick Maclaren's about avoiding implications that aspects of 60559:2011 making optimization worse than necessary for many users need to be implemented to implement this part of the TS.

(strictly conforming implementations) Again an issue from Nick; footnote 3 (not normative) seems (in all useful cases) to contradict the normative text; since even if a strictly conforming program can *use* a conditional feature, it can't produce output depending on it.

(__STDC_WANT_* macros) If WG14 direction is the practice in this document, appropriate edits should be raised against all the other documents listed to follow it (I'm not sure how to track edits for future revisions not coming from DRs/TCs, when there isn't an active edit process for those documents at present, so maybe it would need raising as DRs even without a proper DR log existing for all the documents).

(long double choice) Another issue of Nick's; giving options seems useless without a way to distinguish between them (and the suggested new macros should fully characterise the values of a type, when used in conjunction with the existing <float.h> macros).

(6.2.6.1 notwithstanding) iscanonical is not an operator, but the other point from the CFP group seems reasonable.

(exceptional conditions) The point (from Nick) is that "not mathematically defined" can be considered to apply to cases returning NaNs, for example. The general principle here (and in a few other comments) is that if some readers of the standard find a way to interpret it that is not as intended, or an ambiguity in the standard, we should amend the text to make things unambiguous - even if some other readers think the interpretation is clear, they should accept that other readers don't find it so clear, and a single good-faith reader finding ambiguity or two such readers disagreeing on the interpretation should be enough reason to amend the wording to clarify things - which means a normative change rather than a footnote or other non-normative text, as the normative text must be clear by itself and if you wish to convince an implementor that their implementation is defective, this needs doing based on normative text only.

If in doubt, the normative text should always aim to make things completely clear to all readers, even if things could be put together from widely scattered bits of normative text, or, worse, other sources of intent, to argue for a unique interpretation.

(encoding errors) No further comments.

(int/size_t) No further comments.

(character sets) This issue arose from a real user bug report against an implementation. My view is that this is a real issue of consistency of C11+TS with 60559:2011, because 60559:2011 added the specification of strings for infinities (not in the previous version) in a slightly different form from how those strings were specified in C99 and C11.

(dynamic floating-point environment) No further comments.

(inline functions) Issue from Al Grant and Nick Maclaren; *I* think it's clear that use of "inline" doesn't change semantics, but Nick at least thinks otherwise, so the principle above about making things clear if any reader thinks there's an issue applies.

(constant rounding modes including float / double) No further comments.

(ISO 24747 and constant rounding modes) No further comments.

(Three comments on constant rounding modes with "Suggest no change" listed) See above on making things clear for *all* readers; in particular, Al Grant was expecting e.g. &sin in a constant-rounding-mode region to produce a pointer bound to that constant rounding mode.

(width of type) No further comments.

(FP_INT_*) No further comments.

(integers of specified width) No further comments.

(fmaxmag / fminmag) No further comments.

(setpayload / setpayloadsig) Jim Thomas previously told me "These functions must not raise any floating-point exceptions, but it appears we haven't said that.". If the CFP view has changed, my only further comment is to make unspecified things explicitly unspecified rather than leaving the intended implementation latitude unclear to the reader.

> Also, we've been looking at venues for 2015 - how do you feel about WG14
> meeting in London in March/April (BSI will provide the facilities)?

BSI hosting a meeting then seems fine with me, although I don't expect to
know until around September 2014 what dates in March / April 2015 I'd be
able to be at such a meeting (I expect to be otherwise engaged for at
least a couple of weeks in that period, but won't have specific dates for
some time).

--
Joseph S. Myers
joseph@codesourcery.com