

WG14 N2065
Meeting notes

C Floating Point Study Group Teleconference

2016-07-26
9 AM PDT / 12 PM EDT

Attendees: Rajan, Jim, Fred, David, Mike, Ian

New agenda items:

None

Last meeting action items:

Mike: Monitor the Part 5 DTS ballot to determine whether or not to put it in the IEEE-754 revision bibliography. - Done.

Jim: Check one of the files from the EDG backup for testing the off site backup. - Not done.

Jim: Correction of David H's name in the minutes from last meeting. - Done.

Work Item: Discuss writing up issue C email reflector message 14283 (evaluation macros) as a DR against C11. - New action item.

Jim: Create a DR for Part 1 for email reflector message 14280 (return = copy/convertFormat). - Done.

Fred's proposed footnote is not in the document sent out by Jim.

Work Item: DR for Part 3 with words needed for email reflector message 14285 (DECIMAL_DIG). - Done (<http://wiki.edg.com/pub/CFP/WebHome/DRs2-20160726.pdf>).

Work Item: DR for Part 3 with words needed for email reflector message 14282 first part. - Done (<http://wiki.edg.com/pub/CFP/WebHome/DRs2-20160726.pdf>).

Work Item: DR for Part 3 with words needed for email reflector message 14282 second part (tgmah). - Done (<http://wiki.edg.com/pub/CFP/WebHome/DRs2-20160726.pdf>).

Work Item: Consider changing the specification to reflect the C11 and IEEE mechanism of conversion to strings to see what it would look like (re Fred's DFP to character string email) in Part 2.

Jim sent email on 2016/07/20 with approaches. Follow on discussion was for examples.

New action items:

Jim: Writing up issue C email reflector message 14283 (evaluation macros) as a DR against C11.

Jim: Add the footnote text as per email on 2016/07/26 to Draft DR 1 ("Is return of same type convertFormat or copy?").

Jim: Part 2: Page 42: Put (significant) in front of 'number' in "integers (s, c, q), where n is the number of digits"

Rajan: Send out rules for C2X proposals to the group.

Next Meeting:

August 30th, 2016, 12:00 EDT, 9:00 PDT

Same teleconference number.

Next WG14 meeting (Pittsburgh, 2016/10/17) has a September 19th mailing deadline.

Discussion:

IEEE 754 revision:

Incremental progress being made.

Two-sum discussion.

Arith23:

David did a report on the 754 revision, Jim did a presentation on the TS's (sent via email to our group).

Questions of how the alternate exception handling works with parallelism like OpenMP, etc.

754 doesn't say anything there either so it is not a C specific problem.

If 'short float' gets in, perhaps add something to Annex F to have binding of short float to `_Float16`.

nVidia, ARM, Intel, AMD all are moving to supporting `_Float16`. The hardware should have support for it.

Machine learning is driving this (ex. photo analysis for facial recognition, etc.)

C++ liaison:

No report.

Part 5:

Publication draft was sent to ISO. ISO adds a cover page and end page (pricing).

That draft had 3 problems. They were reported back to ISO, but nothing back from them yet.

Email comments from Joseph Myers continued (sent on 2016/06/21 to WG14 reflector, numbers 14287-14289 and surrounding):

Part 5:

Draft DR1:

Fred asked for a footnote for `convertFormat` vs `copy`.

Jim sent the words via email.

*Jim: Add the footnote to Draft DR 1.

Draft DR2:

Option 1:

May lead to having more digits than you need.

It is a C DR, but only relevant to extensions to C (TS part 3). Seems strange.

Rajan: We can make this a change to Part 3 that changes C11 to avoid this problem.

Option 2:

Only applies to TS part 3.

Ian: Can do both option 1 and two.

May be an issue for mixed runtimes or hardware or implementations.

Jim: Option 2 is not a runtime determination, it is a `#ifdef` thing.

Ian: Doesn't it imply the runtime will need to know this value. Ex. `print` might need to know. For buffers, etc.

Separately compiled things mixing can have different rules. One CU with the `STDC_WANT` macro defined, another without it.

Jim: Option 2 makes the implementation required to deal with the largest types.

Normally `WANT` macros expose interfaces, but in this case it changes the interface.

Due to the multi-translation unit issue, leaning towards option 1.

Ian: Perhaps add in for option 1, "and no more than the maximum digits for any type the implementation supports". May not be a good idea since it limits expansion.

Draft DR3:

Note that the final change is a change to text that lists a change to C11.

Though the changes are substantive this does simplify it, as Joseph stated.

Like the general approach. Look at it more since it is complicated and in a tricky area.

Conversion of DFP to character strings (Fred's email) continued (Jim's 2016/07/20 email regarding the work item):

Trying to get IEEE conversion functionality:

Options 1 and 3 are not unreasonable, but 2 has the least impact on behaviour.

Option 1 seems easier from an implementation point of view, but option 3 is the easiest from that point of view.

For tabular printing, having the precision helps limit the output and can be useful.

General inclination for option 2.

Fred: Need an example of leading zero's.

Jim: n is number of digits in the coefficient. Leading zero's don't affect it.

Fred: Part 2 has an example with only 6 digits on printed page 43 example 2 (`_Decimal32 x = 6543.00DF` -> `_Decimal32 x = 06543.00DF`).

Fred: On page 42, n is listed as the number of digits instead of significant digits.

*Jim: Part 2: Page 42: Put (significant) in front of 'number' in "integers (s, c, q), where n is the number of digits"

What should be proposed for the C standard (<http://wiki.edg.com/pub/CFP/WebHome/DRs2-20160726.pdf>):

Proposal of both part 1 and 2 in total.

Propose part 4 in total.

Perhaps break it into the math functions and reduction functions.

Propose this as a conditionally normative annex?

Fred: Adding it to the main body would be too much for implementers to do.

Jim: Can be a part of the main body but still be conditional (ex. complex, atomics).

Can say upfront that this can be an annex.

Jim: If 754 adopts twoSum or twoProduct, it would be good to get it in.

For Part 5:

Propose the expression control pragmas, and the optimization pragmas as separate things.

For reproducible results, can propose this only after optimization and evaluation methods are accepted.

For alternate exception handling, should we consider the OpenMP concerns as part of the proposal?

Similar issues occurred before Part 5. Ex. Implementing it with flags in C99.

For 'short float', OK with making it `_Float16`? Yes.

Can C refer to the TS?

Ex. Refer to Part 3 in the main standard since Part 3 is a very large change to the standard and keeping it separate may be more palatable.

Question for David Keaton.

Jim: How do we submit proposals?

*Rajan: Send out rules for C2X proposals to the group.

Rajan Bhakta
z/OS XL C/C++ Compiler Technical Architect
ISO C Standards Representative for Canada
C Compiler Development