Subject: Comments on clause 15
From: Van Snyder

1 Edits

Edits refer to 07-007. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

[Editor: “named … C_NULL_FUNPTR” ⇒ “the named constants C_NULL_PTR and C_NULL_FUNPTR and those with names listed in the first column of Table 15.1 and the second column of Table 15.2, and the types C_PTR and C_FUNPTR”.] 467:15-17

[Editor: Replace “CPTR” by “FPTR” thrice.] 470:18-471:2

Editor: Exdent the paragraphs at the following places:

470:18-471:1 * 471:2-5 471:15 * 471:22-24 471:25-27 472:1-3 *
* = join to previous line as well

[Elsewhere we say “1.2.3 of the C International Standard” instead of the way it’s done here. Editor: Insert “6.2.5, 7.17 and 7.18.1 of” after “in”, delete “, subclauses. . . .” Then move it to the end of the paragraph ahead of the table at 472:24 so the reader will know what the column heading means in advance of reading the table.] 474: Footer of Table 15.2

[Editor: “a … allocatable” ⇒ “allocatable or a pointer” (notice that “procedure” became ”pointer”), insert “the value of” after “but”, “hold” ⇒ “be”.] 475:Note 15.13

[Editor: Insert a blank probably “\ ”) before “does”.] 477:Note 15.22

[Presumably other C functions can have variables associated with a C variable that has external linkage in their usual way. Editor: Insert “all the Fortran program units of” after “within”.] 479:6

2 Companion processor isn’t necessarily C

[Editor: Delete “C”.] 467:26

[Editor: Replace “C” by “companion” at the following places]

467:27 467:29 467:30 467:31 468:1+2 468:9 468:10 472:1 472:4
474:Note 15.11 (twice)

3 Nouns and verbs

Most (but not all) of the descriptions of functions in clause 13 are nouns, while most but not all intrinsic subroutine descriptions are imperative sentences in active voice. Make the ones here consistent with the ones in Clause 13.

[Editor: “Indicates … whether” ⇒ “True if and only if C_PTR_1 is associated with an entity and C_PTR_2 is absent, or if”.] 469:6

[Editor: “Associates” ⇒ “Associate” and “specifies” ⇒ “specify”.] 469:19

[Editor: “Associates” ⇒ “Associate”.] 470:6

[Editor: Delete “Returns the”.] 470:12

[Editor: Delete “Returns the”.]	471:7
[Editor: “Returns the size” ⇒ “Size”.]	471:29

4 If and only if

Several function result descriptions use “if and only if” while others use “true if . . . and false otherwise” or a variation thereof. Settle consistently on “if and only if”.

Editor: Replace “if” by “if and only if” twice.	469:14, 16
[Editor: Delete “and true otherwise”.]	469:14
[It’s not clear to what “Otherwise” refers. Editor: “Otherwise” ⇒ “If C_PTR is not a C null pointer”.]	469:15
[Editor: Delete “, and false otherwise”.]	469:17

5 Questions without answers

Are the procedures in ISO_C_BINDING pure?	468:22 ff
Are the types C_PTR and C_FUNPTR extensible? Do they have any type parameters?	474:1 ff
Can an interoperable scalar be a co-array?	476:2-3
If both Fortran procedures and C functions perform input/output using direct access read and write operations, are the results processor dependent?	480:29-30