Subject: Comments on Section 13
From: Van Snyder

1 Edits

Edits refer to 01-007r3. 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 immediately following text, while a page and line number followed by + (-) indicates that immediately following text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

Optional arguments whose presence or absence affects the rank of the result, frequently named DIM, are prohibited to be optional dummy arguments.	272:4+ in note 13.3
[It is confusing to list all of the reasons that the shape might be defined,and only one that it wouldn't be. Either list none of the reasons it might be undefined, or list them all. Listing none: Editor: Delete "It shall not be an assumed shape array." Listing all: Editor: Insert ", an unallocated allocatable array or a pointer that is not associated with a target".]	272:16-19
[Sounds like only assumed-size arrays can be arguments. Either list all of the arguments, or none. Listing none: Editor: "Assumed ... they" ⇒ "They". Listing all: Editor: "Assumed-size" ⇒ "In addition to arguments allowed for transformational array intrinsic functions, assumed-size" and "these" ⇒ "inquiry intrinsic functions".]	272:21
[Simplification: Editor: Insert "of type logical" after "argument" and delete [272:29].]	272:24, 29
[How can procedures be a set? Editor: Insert "set of" before "bit", "consist" ⇒ "consists", and delete "a set of".]	272:31
[Duplicates note 13.2 on page 272. Editor: Delete.]	274:4-6
[Editor: Move [275:31-32] to here, to put the list into alphabetical order.]	275:26+
[Where are the "other" type conversion functions that these are different from? Editor: "Other" ⇒ "Miscellaneous".]	275:33
[Number doesn't agree. Editor: "an actual argument" ⇒ "actual arguments".]	279:29
[Editor: "has value" ⇒ "has the value".]	281:6
[The term *target* has been split into *data-target* and *proc-target*, and an article is needed. Editor: "*target*" ⇒ "the *data-target* or *proc-target*".]	283:23
[It is inconsistent to describe the result in terms of a prohibited pattern. Editor: insert "X were real with the value REAL(X[,KIND]) and" before "Y".]	286:30
[Editor: "*i*" ⇒ "*ii*" (probably requires changing the paragraph tag).	288:7
[Since the units of CPU_TIME aren't specified, one can't compare the timing of different algorithms on the same computer if different compilers are used. Editor: "computers" ⇒ "systems". Where else than "on a computer" would it be interesting to invoke CPU_TIME to measure the time taken by a computation? Editor: Delete "on a computer".]	288:30
[Editor: Remove the first "and".]	290:32
[Improve precision of cross references: Editor: "7.1.4"⇒ "7.1.4.2" twice.]	291:31, 33

[Editor: Insert another sentence: “The product is formed as if X and Y were first converted to double precision”.] 292:10

[Editor: Insert a comma before “which”.] 293:5

[The dynamic type of *every* disassociated pointer and unallocated allocatable is the declared type. Editor: “If ... type” ⇒ “The dynamic type of a disassociated pointer or unallocated allocatable is the declared type”.] 294:24

[Editor: Insert “a” between “has” and “value”.] 294:34

[Editor: “teh”⇒ “the”.] 296:43

[Simplification (of the same sort as used elsewhere): Editor: “INT(A) ... A” ⇒ “INT(A) = INT (REAL (A, KIND(A)))”.] 300:34-35

[It’s confusing to say “1 if A else *n* if B else 1”.] 302:23-29

> *Case (i):* If A is a whole array or structure component and either dimension DIM of array does not have extent zero or ARRAY is an assumed-size array of dimension DIM, LBOUND (ARRAY, DIM) has a value equal to the lower bound of subscript DIM of ARRAY. Otherwise the result value is 1.

[Simplification (of the same sort as used elsewhere):] 303:18-20

> **Result Value.** LEN(TRIM(STRING)).

[Doesn’t work because MATMUL is not an *op* listed in 7.1.4.2. If it were, this could be simplified. It can be simplified (and made to work) anyway: Editor: at [306:11] “according to 7.1.4.2” ⇒ “as specified in 7.1.4.2 for the * operator”; at [306:13] “according to 7.1.4.2” ⇒ “as specified in 7.1.4.2 for the .AND. operator”.] 306:10-14

[“extended” is used in 7.2.3. Editor: “padded” ⇒ “extended”.] 307:11

[How can an array have *a* declared lower bound if it has several dimensions? Editor: “A” ⇒ “some dimension of A” at [308:26] and “B” ⇒ “some dimension of B” at [308:30].] 308:26,30

[Editor: Delete the extraneous blank after “zero”.] 309:3

[Editor: “)” ⇒ “))”.] 309:10

[“extended” is used in 7.2.3. Editor: “padded” ⇒ “extended”.] 310:13

[Editor: Insert “or allocatable” after “pointer” because it’s not allowed to have a pointer actual argument associated with an allocatable dummy argument.] 314:38

[Number doesn’t agree. Editor: “type parameters” ⇒ “a type parameter”.] 315:9

[A scalar is defined to have rank zero at [17:35], which allows a simplification here: Editor: Delete “or ... one”, “is an array of” ⇒ “has” at [317:8]; delete the first “of” at [317:9].] 317:8-9

[HUGE and TINY aren’t defined for complex arguments. Editor: “X” ⇒ “REAL(X, KIND(X))” twice.] 319:10

[We don’t use this clumsy circumlocution anywhere else. Editor: “the processor-dependent ... the” ⇒ “that of”.] 319:22-23

[“a processor-dependent approximation” is inconsistent with the “same value” at [319:33] and the behavior of AIMAG. Editor: Insert “and KIND is present with a different value from KIND(A)” after “complex”] 319:30-31

Case (iii): If A is of type complex with the value (x, y), and KIND is not present or is present with the value KIND(A), the result has the value x.

[The dynamic type of *every* disassociated pointer and unallocated allocatable is the declared type. Editor: “If ... type” ⇒ “The dynamic type of a disassociated pointer or unallocated allocatable is the declared type”.] 321:11

[Editor: “criteria” ⇒ “criterion”.] 322:35

[Not quite right, now that we can detect negative zero.] 323:43

Description. Magnitude of A with the sign of B.

[Why is the term “arithmetic” suddenly introduced here after not being used anywhere else in Section 13? Editor: Delete “arithmetic”.] 327:3

[Editor: De-italicize the “)”.] 329:4

2 A clarification and simplification

Many intrinsic functions refer to “whose model is as at the end of 13.4” It’s note 13.5 to which reference is made. Why not say that? Editor: Replace “at the end of 13.4.” by “in Note 13.5” at the following places:

291:10 293:30 294:8 295:3 297:22-23 307:23-24 310:25-26 314:4 316:28
317:41 319:11 320:38-39 321:22 323:26-27 325:19 328:14

3 Increased functionality

3.1 Get_Command

[Editor: “Subroutine” ⇒ “Elemental subroutine”.] 295:6

3.2 Get_Command_Argument

[Editor: “Subroutine” ⇒ “Elemental subroutine”.] 295:27

3.3 Get_Environment_Variable

[Editor: “Subroutine” ⇒ “Elemental subroutine”.] 296:33

3.4 Huge

There is no way to know how many characters are in a character set.

[Editor: “or real” ⇒ “, real or character”.] 297:17

Result Value. 297:19-21

Case (i): X is of type integer: The result has the value $r^q - 1$ where r and q are as defined in 13.4 for the model representing numbers of the same type and kind type parameter as X.

Case (ii): X is of type real: The result has the value $(1 - b^{-p})b^{e_{\max}}$ where b, p and $e_{\max}$ are as defined in 13.4 for the model representing numbers of the same type and kind type parameter as X.

Case (iii): X is of type character: The result has the value CHAR$(n - 1)$ where n is the number of characters in the representation method having the same kind as X.

3.5 ICHAR

ICHAR doesn't work if C has a kind that occupies more storage space than a default integer, for example, if C has kind `Selected_Char_Kind ('ISO_10646')` and default integers have 16 bits.

[Editor: Insert "**[, KIND]**" after "**C**".] 299:8

[Editor: Make this the first element of a list of arguments.] 299:13-14

[Editor: Copy [294:31-33] to here.] 299:15

3.6 Merge

Several intrinsic functions permit their arguments not to be defined. The functionality of Merge would be increased if that were allowed here also. It is clear from [433:34-43] that it was intended to allow actual arguments not to be evaluated if the processor can determine that the value isn't necessary.

[Editor: Insert an additional sentence "It is not referenced and need not be defined if MASK is false."] 309:31

[Editor: Insert an additional sentence "It is not referenced and need not be defined if MASK is true."] 309:32

[Editor: Make the current example *Case(i).* Then add:] 309:36-42

Case (ii): The value of MERGE(ORDER,10,PRESENT(ORDER)) has the value of ORDER if ORDER is present and 10 otherwise. Notice that ORDER is not defined if it is not present.

3.7 Product

It is useful to be able to carry out products in higher precision than the operands. That's the reason for DPROD. Since DPROD has only two arguments, it wouldn't be a disaster to convert them to double precision before doing the multiply. With PRODUCT, one would need to convert the array argument to double precision. This would be expected to take a lot more time and space than converting the arguments of DPROD.

PRODUCT(ARRAY,DIM[,MASK,KIND]) or PRODUCT(ARRAY[,MASK,KIND]) 316:39

[Editor: Copy [294:31] to here.] 317:6+

Result Characteristics. The type is the same as ARRAY. If KIND is present, the kind type parameter is that specified by the value of KIND; otherwise the kind type parameter is the same as the kind type parameter of ARRAY. It 317:7

Result Value. The product is formed as though ARRAY were converted to the kind of the result. 317:11

4 Not sure what to do

I think this is the only description that doesn't have an example. We probably don't need one. 286:42+

Should "default" be inserted before "real"? Is it intentionally ambiguous so as to allow any kind of real? Does this require an interp? 288:13

What does "is within range" mean? 325:17