Subject: Comments on Section 4
From: Van Snyder
References: 02-122, 02-133, e-mail message j3.2002-91

1 Edits

Edits refer to 02-007. 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.

[What good does Note 4.1 do? Editor: Delete Note 4.1.] 31:8+ff

[What's the point of "in this standard?" Why so wordy?] 31:20-21
The syntax for literal constants of each intrinsic data type is specified in 4.4.

[Editor: "when" ⇒ "where".] 33:21

[Editor: "this set of" ⇒ "the numeric".] 34:3

[It isn't the *digits* but the *constants* that are represented in "their respective number systems" (and it isn't specified what that means).] 35:18-19
The binary, octal and hexadecimal constants are interpreted according to their respective number systems. The *hex-digits* A through F represent the numbers ten through fifteen, respectively; they may be represented by their lower-case equivalents.

C408 (R408) A *boz-literal-constant* shall appear only as a *data-stmt-constant* in a DATA statement, as the actual argument associated with the dummy argument A of the numeric intrinsic functions DBLE, REAL or INT, or as the actual argument associated with the X or Y dummy argument of the CMPLX function. 35:20-22

(2) as actual arguments to intrinsic procedures other than those for which it is explicitly specified that negative zero is distinguished, and 35:35

[The paragraph is wrong (one can specify double precision with the REAL keyword), it duplicates [36:3-6], and it uses the term "double precision," which hasn't been defined yet. Editor: "and ... uses the" ⇒ "(R503). The"; Insert "is an alternative specifier for one kind of real type" at the end of the sentence.] 36:1-2

[Editor: Convert the hyphens in note 4.9 to math-mode minus signs. Put the numbers in math mode, too; maybe it will improve spacing.] 35:36+ff

[Editor: Insert a comma after "example" in the fourth line of note 4.10.] 36:0+...

[Editor: "one of the following" ⇒ "defined as follows".] 38:27

[Sounds like there's a special character set for fixed form. Editor: Move "in fixed source form" to the beginning of the sentence, put a comma and "it is" after it, and adjust capitalization.] 38:28

[Sounds like there's a special character set for free form. Editor: Move "in free source form" to the beginning of the sentence, put a comma and "it is" after it, and adjust capitalization.] 38:30

[Editor: After the previous two edits are done, exchange items (1) and (2), to make free form look more important.] 38:28-30

[Editor: Insert a comma before "which".]	40:18
[The definition of the properties of sequence type doesn't belong here. Editor: "if ... **type**" ⇒ "a storage order is implied for a sequence type (4.5.1.9)"; Move the sentence that begins "The order..." to be the second sentence at [52:2], then wrap the "sequence type" that was already there with `\tdef`.]	41:13-16
[Paraphrased from 02-133 – Editor: "does ... or" ⇒ contains neither a reference to a specification function nor".]	43:18
[Mistakenly allows a *generic-name* to be the same as the name of a final binding. Editor: "specific" ⇒ "nongeneric".]	44:24
[From Malcolm's e-mail j3.2002-91:] Default initialization provides a means of automatically initializing pointer components to be disassociated (4.5.1.4), and nonpointer nonallocatable components to have a particular value. Allocatable components are always initialized to not allocated.	47:1+
[Editor: Insert "(4.5.3.1)" after "component".]	47:4
[Editor: Insert "(5.1.2.5.1)" after "shape".]	48:4
[Editor: Insert "explicit-shape" before "array" in the first line of Note 4.28.]	48:bottom
[Editor: Insert "nonpointer" before components in the seventh line of Note 4.29.]	49:7+8
[Editor: "If a" ⇒ "A"; "is" ⇒ "that is a *dtio-generic-spec* shall not be"; Delete ", an error condition occurs".]	50:6+3-4
[Specify that pointers, *per se*, aren't finalizable (it's their targets that are). Editor: "An" ⇒ "A nonpointer data".]	50:16
[Line 9 of Note 4.30: Editor: Delete commas before "X" and after "Y".]	51:19+11
[Editor: Delete "A derived-type definition..." to the end of the note – it has nothing to do with component accessibility, or anything else nearby that subgroup could spot.]	More Note 4.39
[Doesn't cover the case of accessing two different types that have the same name. Also doesn't cover lots of other prohibitions in 16.2, and doesn't add anything to 16.2. Editor: Delete "A ... unit."]	53:2
[Editor: "which" ⇒ "that".]	53:12
[Line 1 of Note 4.47 duplicates the normative text two lines above. Editor: Delete "is the name that ... This".]	54:5+2-3
[Note 4.49: Leaves out bindings and type parameters. Editor: Insert ", bindings, or parameters" after "components" twice; ", and" ⇒ ";".]	55:0+6-7
[Unnecessarily wordy compared to other ones. Editor: Delete "procedure of the" and "that of".]	55:11
[Editor: Insert a comma after "overriding" in the first line of Note 4.52.]	56:0+2
[Editor: Insert "are said to" before "correspond".]	56:9
[Editor: "when" ⇒ "where".]	56:14
[Editor: "acording" ⇒ "according".]	57:22

[Awkward to say "the type parameters of the type parameters." It's also a bit coy, given that the only type parameter is the kind type parameter. Editor: "agrees ... of" ⇒ "of the same kind as".] 57:23

[Editor: "actual-arg-spec-list" ⇒ "*actual-arg-spec-list*". Don't include the "-list" part in the indexing. Here's how to do it: "`\si{actual-arg-spec}\st{-list}`".] 58:2

Intrinsic assignment of derived-type entities is described in 7.5.1. This standard does not specify any intrinsic operations on derived-type entities. 59:12-Same ¶

[What's nonintrinsic assignment? Editor: "nonintrinsic assignment" ⇒ "defined assignment (7.5.1.2)".] 59:12

[Too wordy. Editor: "If ... but" ⇒ "Otherwise if".] 59:18

[Too wordy. Editor: "If ... requirements" ⇒ "Otherwise". It might be argued that this creates the classic "dangling else" problem, but the consequent of the first "otherwise if" makes it clear that anything other than "otherwise applies to the nearest previous if" is nonsense.] 59:20-21

[Reads at first as though array components are finalized differently from array objects that aren't components. I hope this helps. Editor: Insert "being finalized" after "entity"; "the components ... are" ⇒ "each finalizable component of each element of that entity is".] 59:22-23

["At the same time" and "the order in which they are finalized" are incompatible. Editor: "at the same time" ⇒ "as a consequence of an event specified in 4.5.10.1".] 59:26

[Editor: "A pointer or" ⇒ "The target of a pointer is finalized when the pointer is deallocated. An".] 59:29

[Editor: In the first line of Note 4.63, "C" ⇒ "The C standard".] 62:0+7

[The asterisk case is covered by a conspiracy of C470 and C540. The colon case, which one might think ought to be covered here if the asterisk case is, is covered by C402. Editor: Delete.] 63:25

NOTE 4.71 64:bottom

Examples of zero-size array constructors are:
(/ INTEGER :: /)
(/ (I, I = 1, 0) /)

NOTE 4.72

An example of an array constructor that specifies a nonkind type parameter:
(/ CHARACTER(LEN=7) :: 'Takata', 'Tanaka', 'Hayashi' /)
In this constructor, without the type specification, it would have been necessary to specify all of the constants with the same character length.

2 Derived type constructors aren't finished

There is no provision to use a procedure name as a "value" for a procedure pointer component – a *procedure-name* is not an instance of an *expr*. Using *expr* in R450 covers the data target syntax, but not the necessary constraints. We add *proc-target* and *data-target* – they carry along constraints C718-C719 and C724-C726. [58:17-59:1] doesn't have the stature of a constraint.

[Editor: "*expr*" ⇒ "*component-data-source*".] 57:29

R450$\frac{1}{2}$ *component-data-source* **is** *expr* 57:29+

or *data-target*
or *proc-target*

C476$\frac{1}{3}$ (R450$\frac{1}{2}$) A *data-target* shall correspond to a nonprocedure pointer component; a *proc-target* shall correspond to a procedure pointer component. 58:3+

[Rank remapping can't work during construction, so [59:1-2] isn't adequate.]

C476$\frac{2}{3}$ (R450$\frac{1}{2}$) A *data-target* shall have the same rank as its corresponding component.

[Editor: "*expr*" ⇒ "*component-data-source*".] 58:4

[Editor: Insert "nonallocatable" after "nonpointer".] 58:8

[Editor: "*expr*" ⇒ "*component-data-source*".] 58:10

[Editor: "constructor expression" ⇒ "*component-data-source*".] 58:17

[Editor: "object" ⇒ "entity".] 59:1

3 Proposed spec change

Revival of pseudo-elementalism. Sometimes you are perfectly happy to have the same finalization done on every element of an array of derived type, but you can't use an elemental procedure because you want to do something forbidden therein, such as I/O.

[Editor: After "argument." insert a new sentence: "Otherwise, if there is a nonelemental final subroutine whose dummy argument is a scalar that has the same type and kind type parameters as the entity, it is invoked once for each element of the entity with that element as an actual argument; the elements are finalized in array element order."] 59:20

4 Question for J3 to ponder

Do we want to specify any accuracy requirements? How about the same as for NEAREST rounding for I/O at [218:29-31]? Notice that this question applies only to conversion of the textual form of constants to machine numbers, not to arithmetic, and that NEAREST is not the same as IEEE rounding. 36:25-26