

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.

7	[What good does Note 4.1 do? Editor: Delete Note 4.1.]	31:8+ff
8	[What's the point of "in this standard?" Why so wordy?]	31:20-21
9	The syntax for literal constants of each intrinsic data type is specified in 4.4.	
10	[Editor: Insert "that has deferred the same type parameter" at the end of the sentence.]	33:7
11	[Editor: "when" ⇒ "where".]	33:21
12	[Editor: "this set of" ⇒ "the numeric".]	34:3
13	[It isn't the <i>digits</i> but the <i>constants</i> that are represented in "their respective number systems" (and it isn't specified what that means).]	35:18-19
14	The binary, octal and hexadecimal constants are interpreted according to their respective number systems. The <i>hex-digits</i> A through F represent the numbers ten through fifteen, respectively;	
15	they may be represented by their lower-case equivalents.	
18	C408 (R408) A <i>boz-literal-constant</i> shall appear only as a <i>data-stmt-constant</i> 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
19		
20		
21		
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
23		
24	[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
25		
26		
27		
28	[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
29		
30	[Editor: Insert a comma after "example" in the fourth line of note 4.10.]	36:0+...
31	[Editor: "one of the following" ⇒ "defined as follows".]	38:27
32	[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
33		
34	[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
35		

1	[Editor: After the previous two edits are done, exchange items (1) and (2), to make free form	38:28-30
2	look more important.]	
3	[Editor: Insert a comma before “which”.]	40:18
4	[The definition of the properties of sequence type doesn’t belong here. Editor: “if ... type ” ⇒	41:13-16
5	“a storage order is implied for a sequence type (4.5.1.9)”]; Move the sentence that begins “The	
6	order...” to be the second sentence at [52:2], then wrap the “sequence type” that was already	
7	there with \tdef.]	
8	[Paraphrased from 02-133 – Editor: “does ... or” ⇒ contains neither a reference to a specifi-	43:18
9	cation function nor”.]	
10	[Mistakenly allows a <i>generic-name</i> to be the same as the name of a final binding. Editor:	44:24
11	“specific” ⇒ “nongeneric”.]	
12	[From Malcolm’s e-mail j3.2002-91:]	47:1+
13	Default initialization provides a means of automatically initializing pointer components to be	
14	disassociated (4.5.1.4), and nonpointer nonallocatable components to have a particular value.	
15	Allocatable components are always initialized to not allocated.	
16	[Editor: Insert “(4.5.3.1)” after “component”.]	47:4
17	[Editor: Insert “(5.1.2.5.1)” after “shape”.]	48:4
18	[Editor: Insert “explicit-shape” before “array” in the first line of Note 4.28.]	48:bottom
19	[Editor: Insert “nonpointer” before components in the seventh line of Note 4.29.]	49:7+8
20	[Editor: “If a” ⇒ “A”; “is” ⇒ “that is a <i>dtio-generic-spec</i> shall not be selected”; Delete “, an	50:6+3-4
21	error condition occurs”.]	
22	[Specify that pointers, <i>per se</i> , aren’t finalizable (it’s their targets that are). Editor: Insert “it	50:16
23	is not a pointer and” after “if”.]	
24	[Line 9 of Note 4.30: Editor: Delete commas before “X” and after “Y”.]	51:19+11
25	[Editor: Delete “A derived-type definition...” to the end of the note – it has nothing to do	More Note
26	with component accessibility, or anything else nearby that subgroup could spot.]	4.39
27	[Doesn’t cover the case of accessing two different types that have the same name. Also doesn’t	53:2
28	cover lots of other prohibitions in 16.2, and doesn’t add anything to 16.2. Editor: Delete “A	
29	... unit.”]	
30	[Editor: “which” ⇒ “that”.]	53:12
31	[Line 1 of Note 4.47 duplicates the normative text two lines above. Editor: Delete “is the name	54:5+2-3
32	that ... This”.]	
33	[Note 4.49: Leaves out bindings and type parameters. Editor: Insert “, bindings, or parameters”	55:0+6-7
34	after “components” twice; “, and” ⇒ “,.”]	
35	[Unnecessarily wordy compared to other ones. Editor: Delete “procedure of the” and “that	55:11
36	of”.]	
37	[Editor: Insert a comma after “overriding” in the first line of Note 4.52.]	56:0+2
38	[Editor: Insert “are said to” before “correspond”.]	56:9
39	[Editor: “when” ⇒ “where”.]	56:14

1	[Editor: “acording” ⇒ “according”.]	57:22
2	[Awkward to say “the type parameters of the type parameters.” It’s also a bit coy, given that	57:23
3	the only type parameter is the kind type parameter. Editor: “agrees ... of” ⇒ “of the same	
4	kind as”.]	
5	[Editor: “actual-arg-spec-list” ⇒ “ <i>actual-arg-spec-list</i> ”. Don’t include the “-list” part in the	58:2
6	indexing. Here’s how to do it: “\si{actual-arg-spec}\st{-list}”.]	
7	Intrinsic assignment of derived-type entities is described in 7.5.1. This standard does not specify	59:12-
8	any intrinsic operations on derived-type entities.	Same ¶
9	[What’s nonintrinsic assignment? Editor: “nonintrinsic assignment” ⇒ “defined assignment	59:12
10	(7.5.1.2)”.]	
11	[Too wordy. Editor: “If ... but” ⇒ “Otherwise if”.]	59:18
12	[Too wordy. Editor: “If ... requirements” ⇒ “Otherwise”. It might be argued that this creates	59:20-21
13	the classic “dangling else” problem, but the consequent of the first “otherwise if” makes it clear	
14	that anything other than “otherwise applies to the nearest previous if” is nonsense.]	
15	[Reads at first as though array components are finalized differently from array objects that	59:22-23
16	aren’t components. I hope this helps. Editor: Insert “being finalized” after “entity”; “the	
17	components ... are” ⇒ “each finalizable component of each element of that entity is”.]	
18	[“At the same time” and “the order in which they are finalized” are incompatible. Editor: “at	59:26
19	the same time” ⇒ “as a consequence of an event specified in 4.5.10.1”.]	
20	[Editor: “A pointer or” ⇒ “The target of a pointer is finalized when the pointer is deallocated.	59:29
21	An”.]	
22	[Editor: In the first line of Note 4.63, “C” ⇒ “The C standard”.]	62:0+7
23	[The asterisk case is covered by a conspiracy of C470 and C540. The colon case, which one	63:25
24	might think ought to be covered here if the asterisk case is, is covered by C402. Editor: Delete.]	

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.

25 **2 Derived type constructors aren’t finished**

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

30	[Editor: “ <i>expr</i> ” ⇒ “ <i>component-data-source</i> ”.]	57:29
----	---	-------

