

Subject: Comments on Clause 4
 From: Van Snyder

1 Edits

2	Edits refer to 07-007r1. Page and line numbers are displayed in the margin. Absent other instructions, a	
3	page and line number or line number range implies all of the indicated text is to be replaced by associated	
4	text, while a page and line number followed by + (-) indicates that associated text is to be inserted after	
5	(before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.	
6	[Editor: Combine the paragraphs and simplify: Replace “is” by a comma, delete the period after “4.4”,	45:17-18
7	delete “The syntax for denoting a value”, and delete “the particular value” giving:]	
8	The syntax for literal constants of each intrinsic type, specified in 4.4, indicates the type, type parameters,	
9	and value.	
10	[Editor: Replaced “given a name” by “named”.]	45:19
11	[Editor: Delete “be used to”.]	45:20
12	[Editor: Delete “Such ... arrays.” The requirement belongs at [79:19].]	45:21-22
13	[Sounds like the function has the OPERATOR interface. Editor: Replace “with” by “using”.]	45:26
14	[Editor: Insert “, 7.1.6, 12.4.3.4.1” after “4.5.11”.]	45:29
15	[Editor: Move [46:1] to replace “A type might be parameterized.”]	45:31-46:1
16	[Editor: Replace “with” by “by”.]	46:11
17	[Editor: Replace “will be” by “are”.]	46:2+3
18	[Editor: Delete “The”.]	47:33
19	[Editor: Replace “have been” by “be”.]	48:4
20	[Editor: Replace “was” by “were”.]	48:8
21	[Editor: Before “is” insert “that is not an unlimited polymorphic object”.]	49:3
22	[Editor: Delete Note 4.6; this seems like a weird place to quote C611, since there’s nothing else here	49:6+1-3
23	about components.]	
24	[Editor: Delete C410, in light of C411. If we must have it, we need an identical one at C412a.]	50:11
25	[Simpler to refer to 7.1.9.2. Delete [53:1-6]. Replace the second sentence at [53:8-10] by the following:]	53:1-10
26	Otherwise, the kind type parameter value of the complex literal constant is the same as the kind type	
27	parameter of the result of a numeric operation in which x_1 and x_2 are the real and imaginary parts	
28	(7.1.9.2).	
29	[Editor: Add the list to Annex E.]	54:17-22
30	[Join the constraints: Replace “type. (R430) If SEQUENCE appears” by “type, and”.]	61:3-5
31	[Editor: In the third line replace “. Nor” by “, nor” (starting a sentence with a conjunction is bad style).	61:Note 4.23
32	In the sixth line replace “will always be” by “is” (the situation might be different in the year 2350).]	
33	[Editor: Insert “of type” before “default”.]	63:14
34	[Editor: Insert a right parenthesis after “7”.]	64:5
35	[The constraint doesn’t constrain anything, since it explicitly allows everything the syntax allows. Editor:	65:12-14
36	Delete C443. If we must keep the essence, make it a note.]	
37	[Editor: Insert \ before “si”.]	65:24
38	[Editor: Delete second “be”.]	65:25

39	[Editor: Replace “the POINTER attribute” by “POINTER attributes”.]	65:28
40	[Editor: Delete second and third “be”.]	66:1-2
41	[Editor: Insert “if DIMENSION(<i>component-array-spec</i>) appears in” before “its” and delete “contains ...	66:19-20
42	<i>component-array-spec</i> ”.]	
43	[We don’t have “pointers to”. Editor: Replace “pointer ... holding” by “character array pointer”.]	67:Note 4.34
44	[Overlooks unlimited polymorphic and CONTIGUOUS. Should be referring to pointer assignment to	68:15-17
45	avoid such problems. Editor: Replace “the pointer ... rank” by “the target would be permitted as a	
46	target of the pointer in a pointer assignment (7.2.2)”.]	
47	[Sounds like component keywords are not used in structure constructors. Editor: Replace “where” by	70:3
48	“if”.]	
49	[Editor: Insert “, and shall not appear unless the binding is within the definition of an abstract type”.]	73:11
50	[Editor: Delete “A deferred ... type.”]	73:18-19
51	[Can be simplified. Editor: Replace “dummy argument ... variable” by “nonoptional, nonpointer,	74:13-14
52	nonallocatable, nonpolymorphic dummy argument”.]	
53	[Editor: Insert “scalar” before “values”. See the edit for [45:21-22].]	79:19
54	[Editor: Replace “nonprocedure” by “data” (which is the term used in R441).]	80:4
55	[Editor: Delete “an allocatable entity of the same rank,” since that’s covered by the rest of the paragraph.]	81:6
56	[Editor: Insert “(7.1.6)” after “operation”.]	81:16
57	Editor: Replace “Each” by “A scalar object of the type and type parameters of each”, insert “scalar”	84:14-16
58	before “variable”, replace “this” by “the”, insert “specified by <i>type-spec</i> ” after “parameters”, and insert	
59	a new paragraph “The dynamic type of every <i>ac-value</i> expression shall be the same.”	

60 2 Questions without answers

61	Is the set of valid values for objects of type bits really specified by the standard?	45:12
62	Why did we choose 18, which seems entirely arbitrary? It would be better to tie the requirement to	49:18-19
63	DOUBLE PRECISION by way of NUMERIC_STORAGE_UNIT.	
64	Am I imagining things or does this contradict C420?	55:7-10
65	Why list the operation here?	55:34-56:1
66	Why list the operations here?	57:21-25
67	Doesn’t the “nonpointer, nonallocatable” part apply to the dummy argument, not the actual argument?	75:35-36
68	Should this be “... an actual argument that is associated with a nonpointer, nonallocatable dummy	
69	argument with INTENT(OUT)...?” See [98:15-17].	
70	Did we plan to allow construction of polymorphic arrays? If so insert a new paragraph “The dynamic type	84:14-16
71	of every <i>ac-value</i> expression shall be the same.” Otherwise, insert “nonpolymorphic” before “variable”	
72	in the edit for [84:14-16] in the previous section, and don’t insert the new paragraph.	

73 3 Spec change from 2003

74	The “if and only if” seems too strong: If a procedure that has a passed-object dummy argument is	68:9
75	bound to a type that is neither SEQUENCE nor BIND it has to suffer the performance penalty of a	
76	polymorphic argument. Couldn’t we put the requirement on inheritance, or maybe even overriding?	
77	What’s wrong with internal procedures?	72:16