

Subject: Comments on Clause 5  
 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: Replace “[ , <i>attr-spec</i> ] ...” by “[ , <i>attr-spec-list</i> ]” if you agree it’s clearer.]	87:14
7	[Editor: Insert “or intrinsic” after “generic”.]	88:33
8	[Why pussy-foot around automatic co-arrays with the SAVE attribute in C528? Editor: Insert “a	88:37
9	co-array or” after “be”.]	
10	[Editor: Replace “the variable ... specified” by “default initialization is specified for a direct component	89:8-9
11	of the type of the variable”.]	
12	[Editor: Remove the parentheses around “9.5.2.5”.]	90:29
13	[Using “shall” in the list is wrong. Editor: Replace the list by the following:]	92:22-31
14	<ul style="list-style-type: none"> <li>• If its base object is a pointer or assumed-shape array, it has the CONTIGUOUS attribute.</li> </ul>	
15	<ul style="list-style-type: none"> <li>• It is not the real or imaginary part of an array of type complex.</li> </ul>	
16	<ul style="list-style-type: none"> <li>• Its designator does not contain a <i>substring-range</i>.</li> </ul>	
17	[Better yet:	
18	<ul style="list-style-type: none"> <li>• If its designator contains a <i>substring-range</i> the first <i>scalar-int-expr</i> either does not appear or is an</li> </ul>	
19	initialization expression with the value 1, and the second <i>scalar-int-expr</i> does not appear.]	
20	<ul style="list-style-type: none"> <li>• It does not have a vector subscript.</li> </ul>	
21	<ul style="list-style-type: none"> <li>• Only its final <i>part-ref</i> has nonzero rank.</li> </ul>	
22	<ul style="list-style-type: none"> <li>• If a <i>subscript-list</i> appears it satisfies the following conditions:</li> </ul>	
23	<ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>– A <i>stride</i> does not appear.</li> </ul> </li> </ul>	
24	[Better yet:	
25	<ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>– If <i>stride</i> appears it is specified by an initialization expression with the value 1.]</li> </ul> </li> </ul>	
26	<ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>– If any <i>section-subscript</i> is a <i>subscript</i>, it is not followed by a <i>subscript-triplet</i>.</li> </ul> </li> </ul>	
27	<ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>– Each <i>subscript-triplet</i> but the last consists of a single colon [, or two colons and a <i>stride</i>.] with</li> </ul> </li> </ul>	
28	no <i>subscripts</i> .	
29	[Editor: Insert “or submodule” after “module”.]	93:17
30	[Editor: If C513 is amended explicitly to prohibit automatic co-arrays, replace “This ... example, the”	93: Note 5.11
31	by “The”.]	
32	[C530 duplicates C513, doesn’t cover common (which is covered by C598), and prohibits using length	94:11-12
33	type parameters within a type definition. Editor: Delete C530.]	
34	[The sentence at [95:2] is clearer. It would also be clearer if it appeared before the lower bound value is	94:23-24
35	referenced. Editor: Delete “If the <i>lower-bound</i> ... is 1.” and copy the sentence “If <i>lower-bound</i> appears	
36	... is 1.” at [95:2] before “The value” at [94:19].]	

37	[The part about rank duplicates [93:2]. The part about bounds and shape duplicates [95:16-18]. Editor:	95:5-6
38	Delete “and a specified . . . association.”]	
39	[The sentence about bounds and shape is defective by not mentioning argument association, and dupli-	95:7-8
40	cates [95:19-23] anyway. Editor: Delete “Its bounds . . . target.”]	
41	[Editor: Replace “be written” by “appear” because there’s no WRITE statement under discussion.]	96:13
42	[There’s no need to put “bounds” into possessive case; it’s a perfectly good adjective. Editor: Delete	97:3
43	the apostrophe.]	
44	[It’s not obvious that the PROTECTED attribute “ensures” anything. Editor: Perhaps “specifies” is a	101: Note 5.20
45	better word than “ensures” in the penultimate line of the note.]	
46	[Editor: Delete “accessible”.]	102:10
47	[Editor: Replace “will have” by “has” in the first line. Pointers don’t “point to” they are “associated	102: Note 5.23
48	with”. Editor: Replace “point only to” by “only be associated with” in the third line.]	
49	[Editor: Delete “necessarily”.]	103:5
50	[Editor: Replace “If” by “A”. Replace “has been . . . definition” by “for which default initialization is	105:1-2
51	specified for any direct component”. Delete “, it”. Shouldn’t this be a constraint?]	
52	[Editor: Replace “has not been” by “is not”.]	105:4
53	[Editor: Replace “An” by “The array properties of an” and replace “have had its array properties” by	105:5-6
54	“be”.]	
55	[Editor: Replace “and” by “or”.]	105:23
56	[Editor: Replace “have been” by “be” and replace “made accessible” by “accessed”.]	106:2
57	[Editor: Replace “have been” by “be” and replace “made accessible” by “accessed”.]	106:12-14
58	[Editor: Replace “If” by “Otherwise if”. As it stands, it contradicts IMPLICIT NONE.]	110:16
59	[Editor: Replace “made accessible” by “accessed”, insert a comma after “host association” and replace	110:21,23
60	“has been” by “be”.]	
61	[Editor: Replace “have no” by “not have a direct component with”.]	115:24
62	[Editor: Replace “with” by “for which any direct component has”.]	117:32

## 2 Questions and comments without edits

64	Would it be clearer to put the prohibitions and constraints relating default initialization, explicit initial-	4.5.4.5
65	ization [89:8-9, 105:2], common [115:24], and equivalence [117:32] into subclause 4.5.4.5?	
66	How can the ASYNCHRONOUS attribute both disable and facilitate code motion optimizations?	91: Note 5.5
67	I can’t find normative support for the second sentence in note. If it exists it would be helpful to reference	93: Note 5.10
68	it.	
69	C537 appears to prevent using a length parameter within an expression that specifies a co-bound of a	97:14-15
70	co-array component of a parameterized derived type. How about copying C513:	
71	C537 (R520) A co-array with a <i>lower-co-bound</i> or <i>upper-co-bound</i> that is not an initialization expres-	
72	sion shall not be a local variable of a main program, module or submodule.	
73	There’s no prohibition against the product of the first rank–1 co-extents being greater than the number	5.3.7.8
74	of images. What happens in that case?	
75	It seems that one can declare a variable with an attribute statement, say DIMENSION or TARGET,	108:10, 112:20
76	and then declare it with a <i>type-declaration-stmt</i> that gives it a type different from the type the implicit	
77	rules would imply, unless the attribute statement is a PARAMETER statement. Shouldn’t this be done	

78 by a constraint in 5.2.1? Something like

79 C504a If an entity is declared by a statement other than a *type-declaration-stmt* the *type-declaration-*  
80 *stmt* shall confirm the type and type parameters the object would have according to the implicit  
81 typing rules (5.5).

82 Then delete “If ... implicitly (5.5).” at [108:10-12] and delete “If a namelist ... parameters.” at  
83 [112:20-22].

---

84 It would be easier to use clause 5 for reference, and it would be a tiny bit shorter, if the material on each 5.2.3, 5.3-5.4  
85 attribute declaration statement were in the same subclause as the attribute itself, and the material on  
86 the DATA statement were a subclause of 5.2.3. Most subclause headings would have “and statement”  
87 added, e.g. **ALLOCATABLE attribute and statement**.

---

88 Should 5.7.3 be constraints? 5.7.3