

Subject: Comments on Clause 5  
 From: Van Snyder

## 1 Edits

- 2 Edits refer to 06-007r1. 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: “**attribute**” ⇒ “**attributes**” (but put “attribute” in the index).] 83:4
- 
- 7 [Make sentence more precise. Editor: “The ... list” ⇒ “The *declaration-type-spec* in a type declaration 83:15,17  
 8 statement specifies the type of the entities in the entity declaration list”. Then move [84:2-4] to be  
 9 within the paragraph at [83:17+].]
- 
- 10 [Doesn’t work for BLOCK constructs. Editor: “*specification-expr*” ⇒ “expression”.] 84:36
- 
- 11 [Why does this paragraph only address type parameters, and not bounds? Editor: Insert “or bound” 85:1-2  
 12 after “parameter”, then “the type parameter” ⇒ “its”.]
- 
- 13 [Editor: Insert “or construct” after “procedure” twice.] 85:3,4
- 
- 14 [Make it more precise: Editor: “unless the variable” ⇒ “for a nonallocatable variable unless it”.] 85:8
- 
- 15 [Editor: Insert “or construct” after “unit”.] 85:18
- 
- 16 [Make it more precise: Editor: Insert “, the values of deferred type parameters are specified, or the 86:19  
 17 dynamic type of a polymorphic object is specified’ after “allocated.”]
- 
- 18 [The last sentences of the two paragraphs conflict. Editor: Either delete the last sentence of the second 87:4+6-7  
 19 paragraph, or replace “facilitate” by “disable”.]
- 
- 20 [Editor: Exchange items (4) and (5) to bring “not assumed-shape” and “assumed-shape” together.] 88:2-3
- 
- 21 [Editor: “with the following properties” ⇒ “provided” at [88:6]. Then begin each item in lower case, 88:6-14+3  
 22 replace the full stop at the end of each item by a comma, and insert “and” at the end of item (e) at  
 23 [88:13]. Then delete UTI011.]
- 
- 24 [Sounds like it allows a subset of those that are consecutive, even if the subset isn’t consecutive.] 88:9-10
- 25 (c) The elements of the section are elements of a subset of the base object whose array element-  
 26 order (6.2.2.2) positions within the base object are consecutive.
- 27 [Needs to be revised to conform to ISO guidelines when UTI 011 is cleared.]
- 
- 28 [The term “allows” incorrectly implies that these optimizations are prohibited without the CONTIGU- 88:22+6  
 29 OUS attribute. Editor: “The ... processor” ⇒ “A processor might use the CONTIGUOUS attribute”]
- 
- 30 [Careless reading of C528 might lead one to believe that allocatable co-arrays or arrays declared in the 89:11+11+  
 31 main program cannot have the SAVE attribute. Editor: Add a new paragraph at the end of Note 5.11:]
- 32 An allocatable co-array that is not a dummy argument, or is declared in the main program, is allowed  
 33 to have the SAVE attribute.
- 34 [Alternatively, replace C528:]
- 35 C528 The SAVE attribute shall be specified for a co-array that is not a dummy argument, not declared  
 36 in the main program, and not allocatable.
- 
- 37 [Doesn’t work for BLOCK constructs. Editor: “*specification-expr*” ⇒ “*expr*” twice. Then add two 90:5-6,8+  
 38 constraints:]
- 39 C530a (R512) If an *explicit-shape-spec* appears other than in a BLOCK construct the bounds shall be

40	specification expressions .	
41	C530b (R512) Within a BLOCK construct, every variable in a bound expression shall be previously	
42	declared or accessed by use or host association.	
43	[Editor: Delete “, but ... association” because it’s said at [91:12-13] (not entirely accurately if the	91:1-2
44	“argument association” part is correct, but that will be corrected below).]	
45	[Editor: Delete “Its ... target.” for the same reason.]	91:3-4
46	[Editor: Insert “, and hence its shape,” after “array”. Then insert “or, if it is a dummy argument, when	91:12-13
47	it is argument associated with an allocated actual argument” after “allocated”.]	
48	[Editor: Insert “, and hence its shape,” after “pointer” and delete “in two ways”. Replace the semicolon	91:14-16
49	in item (1) by a comma, and replace the full stop in item (2) by “, or”, then insert]	
50	(3) if it is a dummy argument, by argument association with a nonpointer actual argument or	
51	an associated pointer actual argument.	
52	[Editor: “the” ⇒ “an” because the remainder of the current subclause uses the indefinite article, as do	91:17
53	similar paragraphs in related subclauses (see, e.g., [92:11]).]	
54	[Editor: Insert commas before and after “of any ... character” so it’s clear that “with a subscript order	91:34
55	value” applies when the type is <i>not</i> default character, not <i>only</i> when it <i>is</i> of type default character.]	
56	[If we’re going to repeat things here normatively about what happens to INTENT(OUT) arguments,	94:8-10
57	we ought to repeat it all. Editor: Insert “data” after the first “dummy”, then insert “that a finalizable	
58	associated actual argument is finalized on invocation of the procedure (4.5.6.3), that an associated	
59	allocatable argument is deallocated on invocation of the procedure (6.3.3.1), ” after “specifies”. Then	
60	replace “except ... that” by “and any subcomponents that have default initialization”. Alternatively,	
61	delete “, except ... (4.5.4.5)” because that’s said in Note 5.17, don’t do edits below for [94:13-95:0+23],	
62	and move the second paragraph of Note 5.17 into Note 5.13.]	
63	[Editor: Delete Note 5.13.]	94:13+1-3
64	[Editor: Delete the second sentence (“An actual argument...”) of Note 5.14, or at least move it into	94:17+3-4
65	Note 5.13 if the other edits for [94:8-17] are not done.]	
66	[Editor: Delete the final sentence of the second paragraph of Note 5.17 (“Because an...”).]	95:0+21-23
67	[Editor: Move “An intrinsic... argument” to a new note after C542.]	95:5-6,11+
68	[Editor: Insert “(13.7.138) after “function”, then move “The PRESENT... actual argument.” to a new	95:14-96:1,2+
69	note after C543.]	
70	[Editor: Replace “, converted... entity” by more precise text at [104:10-12]: “; if necessary, the value	96:5-6
71	is converted according to the rules of intrinsic assignment (7.4.1.3) to a value that agrees in type, type	
72	parameters, and shape with the named constant”.]	
73	Elsewhere within Clause 5 where we say “previously” we also say “within the same scoping unit”. Editor:	96:9
74	Insert “within the same scoping unit or construct” after “statement”.]	
75	[Editor: Insert “within the same scoping unit or construct” after the first “declaration” at [100:26] and	100:26,28
76	after “statement” at [100:28].]	
77	[Editor: Insert “or construct” after “unit”.]	101:35
78	[Editor: Insert “or construct” after “unit”.]	102:1
79	[Editor: Insert “within the same scoping unit or construct” after “declared”.]	104:6
80	[Editor: “This” ⇒ “The POINTER”, for consistency with most other subclauses within 5.4.]	104:17
81	[Editor: Insert “or construct” after “unit” twice.]	104:27-28

82	[Editor: Insert “or construct” after “unit”.]	105:2
83	[Editor: Insert “or construct” after “unit” twice.]	105:23-24
84	[Editor: Insert “or construct” after “unit” twice.]	106:2-3
85	[Editor: Insert “or construct” after “procedure”.]	106:7
86	[Editor: Insert “or construct” after “unit” twice.]	106:14,15
87	[Editor: Insert “or construct” after “unit”.]	108:5
88	[Editor: Insert “or construct” after “unit” twice.]	108:9
89	[Editor: Insert “or construct” after “unit”.]	108:16
90	[The middle four paragraphs of Note 5.40 duplicate the constraints, and don’t explain them any better	109:18+5-15
91	than the constraints do themselves. Editor: Delete them.]	
92	[The first sentence repeats the constraints. Editor: Delete it.]	110:3-5
93	[Editor: Insert “or construct” after “unit” twice.]	112:12-13
94	[Editor: Insert “or constructs” after “units”.]	112:24
95	[Editor: Insert “or construct” after “unit”.]	112:25
96	<b>2 Questions without edits</b>	
97	Why is finalization a problem, since it’s the associated actual argument that gets finalized? What if	91:27
98	the dummy argument is not assumed size, but the associated actual argument is? We probably need to	
99	say something about the part of an actual argument that is associated with a dummy argument being	
100	finalized.	
101	Why is “co-bounds” in bold-face type? These aren’t definitions.	92:33,93:9
102	Are non-initialization-expression co-bounds allowed for anything other than dummy arguments, or are	93:18-19
103	automatic co-arrays allowed? Subclause 5.2.2 doesn’t mention co-bounds.	
104	Shouldn’t 5.7.1.5 be constraints?	110:17-19
105	Common association, as such, is not mentioned in 16.5. Should it be?	5.7.2.4
106	What kind of association should be mentioned concerning BLOCK constructs?	112:24