Subject: Comments on Clause 8
From: Van Snyder

1 Edits

Edits refer to 07-007r1. 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.

[Editor: Delete "It is usually a transfer of control".]	178: Note 6.3
[Editor: Move [178:7-9[to [178:10+]. At [178:10] replace "**Form**" by "**Purpose and form**".]	178:7-10
["execution of its *associate-stmt*" is not defined. Editor: Replace "execution of its *associate-stmt*" by "evaluation of every expression within every *selector* and evaluation of every *selector* that is not a *variable*,".]	178:30
[Editor: Exchange "SELECT TYPE" and "ASSOCIATE" since we not describe them in subclauses in the exchanged order.]	179:2
[Editor: Replace "**Form**" by "**Purpose and form**".]	180:14
[Editor: Replace "of the ... statements" by "selector of a CASE statement in a CASE construct".]	180:34
[We don't usually restate a constraint within itself. Editor: Delete "the *case-value-range*s ... that is,".]	181:5
[Editor: Replace "this" by "that".]	183:13
["succeeds" could be read as "not fails". Editor: Replace "succeeds" by "follows" on the second line. In the penultimate line replce "will always precede' by "always precedes".]	183: Note 8.9
	184:2
urpose and forms of the DO construct	P
[Editor: Delete.]	184:13
An *index-variable* in a DO CONCURRENT construct has a scope of the construct (16.4). The type and kind of the index variables and their values for the iterations of the construct"	186:23
[Editor: Delete.]	186:25-27
[Editor: Insert "simultaneously or" after "occur".]	187:9
[Editor: Replace "belonging" by "that belongs to" for consistency with the next list item".]	187:36
C827 A procedure referenced within a DO CONCURRENT construct shall be pure.	188:9
[Editor: Replace "in" by "during".]	188:14
[Editor: Delete "A variable ... terminate." because it's said at 16.6.6(16) which is the correct place.]	188:16-18
[Editor: Replace "in" by "during".]	188:19
[Editor: Replace "in" by "during".]	188:23
[Editor: Replace "in" by "during" twice.]	188:27-28
[Editor: Replace "in" by "during".]	188: Note 8.12
[Editor: Replace "`SUM(A(I,J,:)*B(:,I,J))`" by "`DOT_PRODUCT ( A(I, J, :), B(:, I, J) )`".]	189: Note 8.14
[Editor: In the second line replace "will continue" by "continues".]	189: Note 8.15
[Editor: Replace "example represents" by "silly example illustrates".]	189: Note 8.17

[Editor: Replace "**Form**" by "**Purpose and form**".] 190:2

[Editor: Replace "**Form**" by "**Purpose and form**".] 191:12

[Editor: Replace "whose ... evaluated" by "causes evaluation of every expression within the selector, and evaluation of the selector if it is not *variable*.] 192:29

(5) Otherwise, no block is executed. 193:6+

[Editor: Replace "the EXIT ... it" by "a construct name appears the EXIT statement".] 194:6-7

[Editor: Replace "contains" by "invokes"; delete "reference to a"; replace second "reference" by "invocation".] 197:1-2

[Editor: Delete "known as".] 197:5

[Editor: Delete "that was".] 199:8-9

[Editor: Replace "executed" by "execute".] 201:9

[Editor: In the second paragraph of Note 8.38, replace "will wait" by "waits".] 201: Note 8.38

[Editor: Replace "will be waiting" by "wait"; replace first "will finish" by "finishes"; delete "will" in second "will finish".] 202: Note 8.39

[Editor: Replace "execution" by "executed"; insert "executed" after "QUERY statement".] 202:18

[Editor: Insert "execution of" after "follow".] 202:24

[Editor: Delete "in some user-defined way".] 203:3

[Editor: Replace "will not necessarily" by "might not".] 204:Note 8.4

[Editor: Delete "the" in "from the a synchronization".] 205: Note 8.45

2 Questions without edits

Was there a conscious decision taken that variables defined in more than one iteration are undefined after the construct terminates, while pointers whose pointer association status is changed in more than one iteration have a processor-dependent association status? Could we make them both either undefined or processor-dependent? If we make the pointers undefined, we should say so in 16.5.2.2.3, not here. 188:19-22

Do we need to constrain against abstract types? If not, should we insert a note explaining how pointless it would be? 192:10-11

3 Feature creep or plugging a hole?

[Editor: Insert "if that dimension of the *selector* has an extent; otherwise that dimension has no upper bound" to take care of the possibility the *selector* is a whole assumed-size dummy argument.] 179:5

[Editor: Insert a new paragraph:] 123:8+

An *associate-name* that is construct associated in an ASSOCIATE or SELECT TYPE construct with a *selector* that is a whole assumed-size array is a whole assumed-size array.

4 Use synax instead of constraint

R214	*action-stmt*	**is**	*if-consequent-action-stmt*	11:32
		or	*end-function-stmt*	
		or	*end-mp-subprogram-stmt*	
		or	*end-program-stmt*	
		or	*end-subroutine-stmt*	

or *if-stmt*
R214a *if-consequent-action-stmt* **is** *allocate-stmt*

[Editor: Delete.] 11:41-44,50

[Editor: Replace "*action-stmt*" by "*if-consequent-action-stmt*" and delete C830.] 191:3-5