

Subject: Minor things? left over from 00-144r1, and a few more
 From: Van Snyder
 References: 00-103 00-136r1 00-144r1

1 Introduction

The issues mentioned in 00-103 that didn't get addressed were collected into 00-144r1. Part of 00-144r1 was studied at meeting 152, and part was not. Of the part that was studied, several edits were passed. I collected these, together with the part that was not studied, into 00-144r1. Of the part that was not studied, I thought that several were probably typos, and part probably had technical content. I divided these into separate sections (2 and 3) in 00-144r1.

The editor (taking a suggestion I wrote into 00-144r1) has studied the ones I thought were typos, accepted some, and rejected others (some on stylistic grounds, some because they had technical content). Four of those are included here, with more explanation, for consideration at meeting 153.

There were eight edits that were identified in 00-144r1 by "may have technical content." Of those, six are repeated here for consideration at meeting 153.

I've also added a few.

2 Edits

Edits refer to 00-007r1. 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 immediately following text, while a page and line number followed by + indicates that immediately following text is to be inserted after the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

2.1 From part 2 of 00-144r1

[Editor: The part from "A derived-type object" to the end of the note applies to normative text at [183:36]. Split the note into two parts, and move the identified part to [183:36+], leaving it as a nonnormative note.] 183:8-12

[Editor: We are careful in other places, e.g. (7.5.2) and (5.1.2.11), not to suggest that all intrinsic functions can be used as actual arguments. E.g., even though AMAX1 is a specific name of an intrinsic function, it cannot be used as an actual argument. So as not to contradict other sections, make the wording here the same as is used in (7.5.2) and (5.1.2.11), i.e. replace "function" by "procedure listed in 13.15 and not marked with a bullet (●)" after "function" (but don't do this if 00-187 passes).] 253:46

[The invoking program unit may not be a subprogram. Editor: Replace "subprogram" by "scoping unit." The term "invoking scoping unit" is used frequently in (5.1.2.3), and possibly elsewhere.] 261:4

2.2 From part 3 of 00-144r1

[Editor: Add “and type parameters” after “array bounds.”]	105:44
[Editor: Replace “is determined by the constructor name” by “and type parameters are as described in 4.5.6”.]	115:11-12
[Editor: After “are” add “those of the specific function referenced,”; after “arguments” add “, as specified in 14.1.2.3”.]	115:16-17
[Editor: Before “from” insert “, accessed by use association”. After “IEEE_ARITHMETIC” add “(15)”.]	119:14-15
[Editor: Remove “or” at [259:48]. Insert “, or a reference to the NULL() intrinsic” after “pointer” at [260:1].]	259:48-1
[Editor: Replace “the vendor ... support” by “the supported subset of features is processor dependent.” These kinds of things are usually (always?) described by reference to the processor, not the vendor.]	361:35-36

2.3 New ones

[The example is incorrect. The second argument does not have the same characteristics as the corresponding argument in the procedure POINT_3D_LENGTH overrides. Editor: Replace by the following:] <pre> CLASS (POINT_3D), INTENT(IN) :: A CLASS (POINT_2D), INTENT(IN) :: B </pre>	54:38
[Editor: Add PARAMETER to the list. Also see edit for [81:29] below.]	64:23
[I couldn't find a constraint against ALLOCATABLE and EXTERNAL being specified for the same entity. Replace the one about ALLOCATABLE and POINTER by:] Constraint: If an entity has the ALLOCATABLE attribute it shall not have the POINTER or EXTERNAL attribute.	64:37-38
[Editor: Delete “PARAMETER,” and “DIMENSION,”. The PARAMETER attribute already can't be applied to dummy arguments, which is the only place VALUE can be used. The DIMENSION attribute is prohibited by the constraint at [65:21-22]. But don't bother with this if 00-170 passes.]	65:15,16
[Editor: Capitalize “external” twice to be consistent with usage in the other three paragraphs in this section.]	76:31, 33
[Editor: Delete – Duplicates the constraint at [71:11-12]. See [79:21-23].]	79:27
[Editor: Delete – Duplicates the constraint at [64:31-32]. See [79:21-23].]	80:26-28
[Editor: Delete – Duplicates the constraint at [64:18-20]. See [79:21-23].]	81:17-18
[Editor: Delete – Duplicates the constraint at [64:18-20]. See [79:21-23].]	81:27-28
[Editor: Delete – Duplicates the constraint at [64:23], as modified above. See [79:21-23].]	81:29
[Editor: Delete – Duplicates the constraint at [64:25-26]. See [79:21-23].]	81:38
[Editor: Delete – Duplicates the constraint at [65:32]. See [79:21-23].]	83:1-2

[Editor: Delete – Duplicates the constraint at [64:19-20]. See [79:21-23].]	91:15-16
[This repeats material at [134:40], but the intent here appears to be list all the circumstances in which things get deallocated.]	107:27+
When an intrinsic assignment statement (7.5.1.5) is executed, allocatable components of the <i>variable</i> are deallocated before the assignment takes place.	
[Editor: Insert “is” after the first “and”, delete the second “and”.]	118:13
[Editor: “When” ⇒ “If”.]	189:31
[Editor: Start a new paragraph]	192:27+
The err , eof , and eor arguments correspond, respectively, to the ERR= (9.9.3), END= (9.9.4) and EOR= (9.9.5) specifiers in a data transfer input/output statement.	
[Editor: “hargument” ⇒ “argument”.]	237:22
[Editor: Replace “OPTIONAL ... PUBLIC” by “or OPTIONAL” – Duplicates the constraint at [70:40]. See [79:21-23].]	242:2
[Editor: Insert “(functions only)” at the end – for consistency with the other ones in the list.]	245:27
[Editor: Insert “7.1.3” before “7.1.8.7”. 7.1.3 seems to be the reference for defined operations that would be most interesting here.]	249:23
[Editor: Insert “and type-bound procedure bindings” at the end.]	251:37
[The <i>proc-decl-list</i> isn’t optional. Editor: Replace “at most” by “exactly”.]	252:42
[Editor: Simplify by inserting “those” after “are” and deleting “and are ... interface”.]	253:9
[Editor: After “argument” insert “other than the passed-object dummy argument”.]	255:10
[Editor: After “argument” insert “that does not have INTENT(IN)”.]	257:11
[Editor: After “pointers” insert “that do not become undefined and are”. Otherwise, the statement implies that local pointers without the SAVE attribute retain their values if they happen to be associated with a dummy argument that has the characteristics discussed at this point.]	257:41
[Editor: After “dummy argument” insert “type”. A type can’t be an extension of a dummy argument.]	258:30
[There is a constraint at [255:28] that is nearly identical to the sentence “The label ... reference.” Here, it says “executable construct” while in the constraint at [255:28] it says “branch target.” I don’t know if there’s a real inconsistency, but the difference in wording is confusing. The simplest solution is to remove the sentence here. Editor: Insert “(12.4)” after “specifier” and delete the sentence “The ... reference.”]	260:19-21
[Editor: Delete “that” (finish – hopefully – work begun in 00-136r1).]	352:18