

To: /B subgroup
 Subject: Comments on Section 12
 From: Van Snyder
 References: 01-271

1 Edits

Edits refer to 01-007r2. 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 (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

A subprogram defines a procedure for the SUBROUTINE or FUNCTION statement. If the subprogram also contains one or more ENTRY statements, it defines a procedure for each of them. 237:31-32

[Assumed size is not obsolescent. Editor: Set size in the normal font.] 238:21

[The phrase “whether it is allocatable or a pointer” could be read to require that it be one or the other. Editor: “or” ⇒ “, whether it is”.] 238:30

[Editor: “in an” ⇒ “an”; “in both” ⇒ “both, ”.] 239:39

[Duplicates [240:33]. Editor: Delete the constraint.] 240:21

[Editor: Move [230:33-38, 43] to here (after making the next change) and sort the resulting block of constraints according to the syntax rules to which they refer.] 240:23+

[Editor: “if the” ⇒ “in an”; “has” ⇒ “that has”.] 240:43

[The reference to “procedure declaration statement” is forward. Editor: Insert “(12.3.2.3)” after “statement”.] 241:19

[Editor: “its” ⇒ “a”; delete “, if it has one”.] 242:47

[The sentence of which this line is a part is a run-on sentence. Editor: “and the” ⇒ “. The”.] 243:32

[There is nothing that says what abstract interfaces are used for.] 245:16+

NOTE 12.10¹/₂

Abstract interfaces may be used to declare procedure pointers and external procedures (12.3.2.3).

[Editor: Delete “accessible”.] 247:23

[Editor: Insert “(15.2.3)” after “module”; “then” ⇒ “required to be”.] 247:24

[Editor: Insert “, a procedure pointer” after “procedure”.] 248:40

[Editor: “an allocatable or” ⇒ “allocatable or a” twice.] 250:24

[Duplicates note 5.16 at [75:1-22] almost exactly.] 252:23-44

NOTE 12.25

Argument intent specifications serve several purposes. See note 5.16.

[There is no reason to put these constraints away from the syntax rules to which they apply. 253:1-14

Editor: Delete [253:1] and move [253:2-14] to [249:12+].]

[These paragraphs could be read to say that a dummy argument that has the POINTER attribute, or has the TARGET attribute but not INTENT(IN), cannot be used. Editor: Insert “only” after “taken” at [255:5].] 255:3-10

[Editor: “an intent of IN” ⇒ “INTENT(IN)”.] 256:26

[Editor: For consistency with [255:4-5], “of any part of the entity” ⇒ “the entity or any sub-object of it”.] 256:32

[Editor: “is an asterisk” ⇒ “is an asterisk”.] 258:40

[There are two problems here: (1) It is no longer necessary for the function to be recursive for it to have explicit interface within itself, and (2) the paragraph duplicates [239:8-9], which is the proper place for it. Editor: Delete it (but if you must keep it, “both ... are” ⇒ “RESULT is”).] 260:5-6

[Editor: Move [261:18] to here.] 261:11+

[Duplicates [239:8-9], which is the proper place for it. Editor: Delete this one.] 261:17

[Editor: “and ... INTRINSIC” ⇒ “nor shall it appear in an EXTERNAL, INTRINSIC, or PROCEDURE”.] 262:6

[Duplicates [239:8-9], which is the proper place for it. Editor: Delete this paragraph (but maybe not, if you decide to keep the one at [260:5-6]).] 262:20-21

[Duplicates [239:8-9], which is the proper place for it. Editor: Delete this paragraph (but maybe not, if you decide to keep the one at [260:5-6]).] 262:25-26

[Editor: “program” ⇒ “subprogram”.] 262:29

[Editor: After “statement” insert “that does not have a *proc-language-binding-spec*”.] 263:38

[BIND is no longer a *prefix-spec*. Editor: “the BIND *prefix-spec*” ⇒ “a *proc-language-binding-spec*”.] 263:38-39

[Editor: Insert “*proc-*” immediately before “*language-binding-spec*”.] 264:44

2 Things to do at meeting 159

The /A subgroup ran out of time to consider the following issues at meeting 158. REMEMBER that these page and line numbers refer to 01-007r2!

[The phrase “and it is present (12.4.1.6)” is covered in 12.4.1.6 and 7.1.6 (C709 and item 2). By saying it here, it appears to imply that it need not be present in other circumstances. Editor: Delete it.] 262:44

[Editor: If the advice in 01-264 concerning the definition of what constitutes a branch is not taken, insert the following note:] 263:21+

NOTE 12.40¹/₂

A transfer that results from an alternate return is a branch.

[Editor: Insert “or procedure pointer arguments” after “arguments”.] 266:17

[Editor: Insert “or procedure pointer” after “procedure”.] 266:18

[Editor: Insert “, if” before “any”; “with” ⇒ “has”.]	266:26
[Editor: Insert “is” before “an”.]	266:28
[Editor: Insert “it” before “shall”.]	266:29
[Editor: Insert “ <i>wait-stmt</i> ,” before “ <i>backspace-stmt</i> ”.]	266:41
[Editor: Before “or” insert “that does not have an <i>io-unit</i> , or a <i>read-stmt</i> ”.]	266:43
[Editor: “used” ⇒ “possible”. Then close up some of the empty space before the next paragraph (if possible).]	267:16
[How else than by a generic name or a specific name could an elemental procedure be referenced? Do defined operations count? Editor: “If a generic name” ⇒ Regardless whether a <i>generic-spec</i> ”.]	268:27
[Editor: “block” ⇒ “body” twice.]	369:24,30

3 Don't know what to do (or too lazy to figure it out)

There are no edits in this section.

This constraint allows, by omission, that one could access an interface by use or host association, and then put a specific procedure into that interface that is already in it. Is that the intent?	240:44-45
Does this allow or prohibit the case that a procedure has several names (because of renaming during USE) in a scoping unit?	241:23-24
Do we want to require that they all have the same result type as the derived type name?	243:5
What attributes, if any, are prohibited for dummy arguments of procedures that define operations or assignment? Are TARGET, VALUE and VOLATILE permitted? Is a procedure dummy argument or a procedure pointer dummy argument allowed?	12.3.2.1.2
Should VOLATILE be in the list?	246:11-16
I have no idea what the phrase “the name of the last argument shall be PRINT” does here.	250:14-15
This paragraph appears to be incomplete. It doesn't address INTENT(INOUT) or unspecified intent.	250:23-28
Neither this paragraph, nor any other that I could find in 12.4.1.2, address the case that the dummy argument has assumed length.	250:33-38
This can't work for rank > 1 without more explanation. Suppose we have an array of rank 2, with N rows. Don't we need the dummy argument to have a multiple of N elements? If not, we need to say something to the effect that some of the last rows are not present in the last column (and shall not be accessed). I think it gets worse in higher rank, but I haven't given much thought to the question.	254:18-19
The “, but not both” part is not accurate if there are intervening ALLOCATE statements.	255:47
The term “procedure heading” isn't defined. Is it the <i>function-stmt</i> or the <i>specification-part</i> or both?	259:23-24
Does this paragraph need to address allocatable function results?	259:31-41
Is the ALLOCATABLE attribute permitted?	262:17-19
The term “procedure header” isn't defined. Is it the <i>function-stmt</i> or <i>subroutine-stmt</i> , or the <i>specification-part</i> , or both?	259:23-24

A procedure having a BIND attribute contradicts [70:10] (but that's the topic of 01-271).	263:32
It's not clear why we need BINDNAME= and these paragraphs. If the binder name is derived from the name of the external procedure by the companion processor, and the processor knows which companion processor(s) is(are) to be used, it can derive the binder name(s). Why ask every user to do the research? Are C programmers asked to provide binder names?	264:18-31
The term "base object" appears to be defined only for derived types.	266:26
The term "base object" appears to be defined only for derived types.	268:3
The "has only scalar arguments" part duplicates [262:42].	268:41
Should this "must" be "shall"?	269:5