

Subject: Comments on section 2

From: Van Snyder

1 Edits

Edits refer to 01-007r1. 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 (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

[Editor: alphabetize according to the right-hand sides.]	10:8-17
[Conflicts with [9:11-12] and [237:38-39]. Editor: Delete “unit”.]	12:9
[The <i>select-kind-construct</i> is gone. Editor: Delete “; this includes ... <i>select-kind-construct</i> ”.]	13:29-31
[The phrase “in a subprogram” leads one to ask “If it’s necessary to say that here, where else might a <i>return-stmt</i> be executed?” The answer is “nowhere else – it’s a red herring”. Editor: Delete “in a subprogram”.]	15:4
[Using the definite article in “the exception is an undefined variable” is inappropriate. Other cases include zero size arrays, disassociated pointers and deallocated allocatable variables. (Zero length characters apparently do have a value.) Editor: “the” ⇒ “an”.]	16:20
[Editor: Delete “particular”.]	20:20
[Editor: “Fortran contains” ⇒ “This standard specifies”.]	20:21
[Editor: Italicize “n”.]	20:26
[Editor: “function” ⇒ “procedure”.]	20:39

2 Potential problems with no edits offered

I don’t see how program units can simultaneously be “fundamental” and the largest scale parts of a Fortran program.	11:45
A subroutine can also be invoked by user-defined derived-type input/output.	§2.2.3.0
I thought that we did something having to do with the relation between EXTERNAL and “defined by a means other than Fortran” but I can’t find it. If I recollect correctly, a procedure defined by a means other than Fortran is an external procedure.	12:33-34, 401:33-34
Type-bound procedures and procedure pointers should be mentioned at this point.	§2.2.3.4+
The appearance of “to define” and “defining” in such intimate juxtaposition is uncomfortable.	§2.4.0
An expression or function reference may produce a pointer, which is not a data entity – at least not according to the definition in 2.4.3.	§2.4.3.2,3
Do we need 2.4.6$\frac{1}{2}$ Allocatable?	§2.4.6+
Should pointers be discussed in this subclause?	§2.5.4
Derived types and procedures are defined, not declared. That’s why the discussion is in this	19:36

subclause. Therefore, “declaration” isn’t the appropriate word here. Maybe “specification” would be better.

Do pointers need special treatment here?

19:38-39