Subject: Comments on Section 2, Unresolved issue 349
From: Van Snyder

1 Edits

Edits refer to 02-007. 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: Should "Sections" be plural here?] 11:44

[Editor: Set the second "DATA statements" in obsolescent font.] 15, Table 2.1

[Editor: Insert "(8.2)" after "statements".] 15:5

[Executing an *end-subroutine-stmt* or *end-function-stmt* isn't equivalent to executing a *return-stmt* that specifies an alternate return. Editor: Insert "with no *scalar-int-expr*" after *return-stmt*.] 16:2

[Needs to mention SELECT TYPE. Editor: "and IF" ⇒ "IF constructs, and SELECT TYPE".] 16:14

[The "together with" part at [16:31] doesn't contribute anything. Editor: Delete it, and insert a comma after "values" at [16:32].] 16:31-32

[Editor: "struvtures" ⇒ "structures".] 17:9

[Run-on sentence. Editor: ", and" ⇒ "; it".] 18:17

[Run-on sentence. Editor: ", and" ⇒ ";"; Conversion defect: Editor: Insert "**size** is the total" after "its".] 19:6

[The term "scalar-like" suggests there's something not-quite-scalar like too. Editor: ""scalar-like"" ⇒ "scalar" (notice that the quotes disappear).] 19:16

[What's "certain" about the restrictions. Editor: "certain" ⇒ "the"; "constraints (16.4.3)" ⇒ "restrictions described in 16.4.3".] 19:36

[Everything is "in this standard" but of all the subclauses of 2.5, only 2.5.2 bothers to say so. Nothing's special here. Editor: Delete "in this standard".] 20:7

[Keywords aren't used in dummy argument lists. Editor: Insert "actual" before "argument".] 20:13

[Editor: Replace "The term ... **undefined**." with the following:] 20:24-27
The term **definition** is used in two ways.

(1) It refers to the specification of derived types and procedures.
(2) When an object is given a valid value during program execution, it is said to become **defined**. This is often accomplished by execution of an assignment or input statement. When a variable does not have a predictable value, it is said to be **undefined**. Similarly, when a pointer is associated with a target or nullified, its pointer association status is said to become **defined**. When the association status of a pointer is not predictable, its pointer association status is said to be **undefined**.

[Pointers can be defined or undefined, too. Editor: "two" ⇒ "three". Insert "When a pointer is associated to a target, or nullified, its association status becomes **defined**. When the association 20:24, 27

status of a pointer is not predictable, it is said to be **undefined**." before "Section". Insert "and the association status of pointers" after "variables". (The term "unpredicatable" may not be the most desirable one, but it is already used for nonpointer variable values.)]

[Why be coy? Editor: "Under certain circumstances," ⇒ "When"; "and" ⇒ "it".] 20:26-27

[Now that 2.5.4 discusses "association" it would be useful to have this subclause before it. Editor: Move to [20:20-].] 21:7-13

[Editor: Listify the subclause. Start the first list element by replacing "The first" by "The qualifier" at [21:15]. Start the second list element at [21:21] after deleting "second use of".] 21:15-23

2 Unresolved issue 349

The editor liked the former wording of [12:9-10]. The current wording replaces "any" with "a processor-dependent." The editor says the replacement is an invitation to incompatibilities. For example, a processor may require alphabetical order. But "any" order prohibits a processor from imposing the dependencies contemplated in Note 2.2.

The standard should not discuss ordering – especially in a sentence that begins "The set..." – because that's the province of the linker. If we just take out the part about ordering, the sentence becomes nearly identical to the first sentence of the paragraph. BUT, it's still defective in not mentioning that a program can have only one unnamed block data program unit.

[Editor: Delete "The set. . . ."] 12:9-10

[Editor: Delete unresolved issue note 349. At the third line of Note 2.2, "Since" ⇒ "This standard places no ordering requirement on the program units that constitute a program, but since".] 12:10+1ff

3 Remark for the editor

The two-column stuff that's done with the LaTeX `tabular` environment only gets one line number per `tabular`. I tried `tabbing` and it doesn't get any at all. Here's a macro that gets line numbers: 17:35,37

```
\newcommand{\twoup}[3][1.75in]{\hspace*{0.25in}\makebox[#1][l]{#2}#3}
```

It takes three arguments, the first of which is optional with default 1.75in. It indents 0.25in, puts its second argument left justified in a box having a width given by its first argument, then emits its third argument. Here's how it looks:

named scalar a scalar object
a named array (an array object)

...

an array element (a scalar subobject)
an array section (an array subobject)
a structure component (a scalar or an array subobject)
a substring (a scalar subobject)

The first one was set using `\twoup{named scalar}{a scalar object}`. Here's one set using `\twoup[2.5in]{named scalar}{a scalar object}`:

named scalar a scalar object

I didn't look to see how many of these things there are, but I suspect it's a lot. How many do you want to change?