Subject: Comments on Section 12
From: Van Snyder

1 Edits

Edits refer to 01-007r3. 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.

[The term "dummy procedure" is defined, but neither "dummy argument" nor "actual argument" is, except indirectly by tenuous connection to the syntax terms *dummy-arg-name* and *actual-arg*. Editor: Replace "The reference ... definition" by the following, which begins in the same paragraph:] 239:4-6

The sequence of computations encapsulated by a procedure has access to entities in the invoking scoping unit by way of argument association (12.4.1). A **dummy argument** is a name that appears in the SUBROUTINE, FUNCTION or ENTRY statement in the declaration of a procedure (R1226). Dummy arguments are also specified for intrinsic procedures and procedures in intrinsic modules in Sections 13, 14 and 15.

The entities in the invoking scoping unit are specified by actual arguments. An **actual argument** is an entity that appears in a procedure reference (R1221).

A procedure may also have access to entities in other scoping units, not necessarily the invoking scoping unit, by use association (16.7.1.2), host association (16.7.1.3), linkage association (16.7.1.4), storage association (16.7.3), or by reference to external procedures (5.1.2.6).

[Editor: Delete "The reference ... definition" because it's covered adequately in 12.4.1, which is now referenced here.] 239:4-6

[Editor: Insert "an" before "interface".] 241:38

[The term *target* has been split into *data-target* and *proc-target*. Editor: "*target*" ⇒ "*data-target* or *proc-target*".] 255:39

[Icky wording. Editor: "been declared a pointer" ⇒ "the pointer attribute".] 261:4

[Just to illustrate a new feature:] 261:38

```
IMPORT C_INT, C_FLOAT
```

[More complicated than necessary. Even if it's not simplified, the two extra brackets at the end should be deleted.] 263:8-11

R1234 *entry-stmt* **is** ENTRY *entry-name* [([*dummy-arg-list*])] ■
■ [, *proc-language-binding-spec*] ■
■ [RESULT (*result-name*)]

[The term *target* has been split into *data-target* and *proc-target*. Editor: "*target* of" ⇒ "*data-target* or *proc-target* in".] 267:37

[There is no term *forall-assignment*. Editor: "*forall-assignments*" ⇒ "*forall-assignment-stmts*" (observe that the final "s" is not in "syntax term" font).] 268:30

[Editor: Insert "12," before "12.4.1" (see edits for [239:4-6] above).] 387:8

[Editor: Insert "12," before "12.5.2.1" (see edits for [239:4-6] above).]	390:24

2 Not sure what we have in mind

Defined operations were introduced in Fortran 90, when INTENT implied the POINTER attribute wasn't present, and dummy arguments couldn't be ALLOCATABLE. Do we now want to say "nonpointer, nonallocatable" along with "nonoptional"?	245:31-32
Defined assignment was introduced in Fortran 90, when INTENT implied the POINTER attribute wasn't present, and dummy arguments couldn't be ALLOCATABLE. Do we now want to say "nonpointer, nonallocatable" along with "nonoptional"? If so, this also has implications at [413:25], where we probably should remove "intrinsic".	246:32-33