Subject: Comments on Annex A, issue 283
From: Van Snyder

1 Edits

Edits refer to 01-007. 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 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.

[These could be simplified to:] 12:5-8
(2) An interface body, excluding any scoping units in it, or
(3) A program unit or subprogram, excluding any scoping units in it.
[It isn't necessary to exclude scoping units that can't be there by mentioning only the other ones. Also see the edit for [408:11-13] below.]

J3 internal note 18:41+

Unresolved issue xxx The type parameter keyword definition doesn't include structure constructors. A glossary entry is also needed.

[There is no subclause reference for the term "bit field." The only other place it appears is in the definition of the MVBITS intrinsic subroutine. The first sentence contributes nothing to this standard; the second contradicts it. Editor: Delete the entry.] 400:18-19

[Editor: Delete unresolved issue note 283, instead of modifying it and moving it as 01-126r1 advocates.] 400:41-45

[The term "compatible type" does not appear anywhere except here. There is no subclause reference here. There seems to be no need for this glossary entry. Editor: Delete it.] 401:13-14

---

component name keyword (2.5.1) : A component name. It may be used in a structure constructor to indicate the component for which a value is provided. 401:15+

[The ASSOCIATE construct isn't in the list and the ones that are here are in random order – neither alphabetical nor in the order they appear in the normative text. Editor: "SELECT CASE ... FORALL" ⇒ "ASSOCIATE, DO, FORALL, IF, SELECT CASE, SELECT TYPE". Note: SELECT CASE was moved *after* the vote on this paper.] 401:31-32

[Editor: "which" ⇒ "that".] 402:28

[Editor: "A CASE ... (R216)" ⇒ "An action statement (R216), or an ASSOCIATE, CASE, DO, FORALL, IF, SELECT TYPE or WHERE construct".] 403:12-13

[Editor: "A main ... module subprogram" ⇒ "A scoping unit that contains another scoping unit is called the **host** of the contained scoping unit" works, now that we have the IMPORT statement.] 404:21-23

[Editor: "an internal ... definition" ⇒ "a contained scoping unit" (to account for the IMPORT statement).] 404:24-25

[The term "incomplete type" does not appear anywhere except here. There is no subclause reference here. There seems to be no need for this glossary entry. Editor: Delete the glossary 404:30-31

entry.]

[Editor: After "Components" insert "or procedure bindings".]	404:32
keyword (2.5.2) : Statement keyword, argument keyword, type parameter keyword or component name keyword.	405:23
[Editor: Insert a comma after "arithmetic"; "representing" ⇒ "which represents".]	406:16
[Editor: After "**mode**" insert "(15.3)".]	408:1
[These could be simplified to:] (2) An interface body, excluding any scoping units in it, or (3) A program unit or subprogram, excluding any scoping units in it. [It isn't necessary to exclude scoping units that can't be there by mentioning only the other ones. Also see the edit for [12:5-8] above.]	408:11-13
[Editor: "which" ⇒ "that".]	408:28
structure constructor (4.5.8) : A syntactic mechanism for constructing a value of derived type.	409:5+
[Editor: "*co*" ⇒ "object designator". (Is this already fixed by another paper?)]	409:6
[Editor: "component ... and" ⇒ "derived-type component that".]	410:1
[Editor: Before "in" insert "defined".]	410:4
[Editor: Delete "'There ... Fortran."]	410:5

2 Do we still need comparison to Fortran 77?

Delete the sentences that begin "Note that in Fortran 77 ..." at:
399:32-33 403:39-40 405:39-40 406:8 406:14-15 409:10-11 409:16-17 410:15

3 Not sure what to do

Some of these may need new unresolved issue notes.

Is the sentence "For an intrinsic type, the set of data values depends on the values of the type parameters" still adequate? I.e., do we need to say something about parameterized derived types, or do we assert that their values are ultimately a composite of intrinsic types?	402:3
Is this accurate now that we have stream access?	403:33
Should a C-based definition be here too?	404:10-11
Is the sentence "A parameter whose values label the available kinds of an intrinsic type" still adequate? I.e., do we need to say something about parameterized derived types, or do we assert that their values are ultimately a composite of intrinsic types?	405:24-25
A procedure isn't a computation; it specifies one.	407:17
The list doesn't seem to conform to ISO guidelines – i.e. item (1) ends with a period. Is this OK?	408:18-19
(3) The object whose type determines which block of a SELECT TYPE construct is executed. [And maybe the editor needs to fiddle with sentence structure (or lack thereof) so that it fits into ISO guidelines for lists.]	408:19+
It wouldn't hurt to have a glossary entry for **stream**.	409:2+