Subject: Comments on section 5
From: Van Snyder
References: 01-138r1, 01-166, 01-198r1

This paper contains the comments from 01-198r1, updated for 01-007r2.

Only needed if the argument has a *language-binding-spec* 65:13-14

There appears to be no reason for the "that has a *language-binding-spec*" part. I don't see why VALUE wouldn't work just fine for Fortran subprograms. 65:16

"If the kind ... default integer" duplicates [32:12-13]. 66:29-30

"If the kind ... default real" duplicates [34:11-12]. 66:33-34

"The kind ... (0.0D0)" duplicates [34:14]. 66:37

"If the kind ... default complex" duplicates [35:11-12]. 66:43-44

"If the kind ... default character" duplicates [36:10-11]. 68:8-10

Duplicates [38:10-12]. 68:16-18

If we had a term for "type compatible and all the kind type parameters have the same value" the discussions of argument association and generic resolution would be simpler. 68:38-41

The term "base object" appears to be defined only for structures. If that's true, what is a "base object" here? If it needs to be here, insert "a" before "variable". 69:33

Where else might a *bind-spec-list* appear? 70:24-25

"Shape" should be "bounds". 70:36

"*explicit-shape*" and "*deferred-shape*" should be "*explicit-bounds*" and "*deferred-bounds*" here, everywhere else these syntax terms appear, and everywhere the non-syntax terms similar to them appear. 70:42,44

Is the concept of "defined" defined for anything other than a variable or a pointer association status? 72:12

The specs really said "disassociated"! This would be cool, but almost certainly "disassociated" should be "undefined". "disassociated" conflicts with [251:12] and [378:26]. 74:13

The essence of note 5.15 supports the answer to interpretation 31 proposed in paper 01-166. It is inconsistent with the answer adopted at meeting 157. Either the answer should be changed, or this passage needs editing to make the interpretation complete. 74:35-47

This only says when a pointer can't be referenced. Do we assume the contrapositive to be true? If so, this supports the answer to interpretation 31 proposed in paper 01-166. It is inconsistent with the answer adopted at meeting 157. Either the answer should be changed, or this passage needs editing to make the interpretation complete. 76:9-10

If the advice implied by the remark for [65:16] above is accepted, insert "and the procedure has a *language-binding-spec*" after the first "argument". 77:17

The difference between the effect of VOLATILE on allocatable entities and their allocation status should be described. 77:36+

Do we need to say anything about deferred or assumed type parameters? 83:40+

The term "base object" appears to be defined only for structures. If that's true, what does the constraint mean? 87:27