Subject: Issue 259
From: Van Snyder

1 Background

J3 unresolved issue 259 is about the use of "nonconstant" in the specification for namelist and anywhere else it might appear. It appears in one other place. Studying its appearance in namelist made namelist look pretty lame.

The definition of namelist in F90/95 excludes arrays with nonstant bounds (and characters with nonconstant length) for no particularly good reason; it appears to the outsider that namelist was only grudgingly and minimally added due to public demand.

It is clear that the niggling constraint at 86:39-42 has outlived its usefulness. We should remove forthwith as much of it as we can. In the current draft, we've forgotten to do anything about type parameters other than character length, or polymorphism, so it is broken anyway.

Similarly, since user-defined derived-type input/output is allowed for namelist input/output, we should lift the prohibition against objects with pointer or allocatable components from appearing in namelist. We still need the one against actual pointers and allocatables though. Otherwise, to make it consistent with F90/95, we would need to add something terribly icky about nonkind type parameters that are used for array bounds or character lengths, for which the values are not specified by initialization expressions.

2 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.

Constraint: A *namelist-group-object* shall not be an assumed-size array, an assumed-shape array, a pointer, or an allocatable variable. — 86:39-42

[The constraint avoids objects of unknowable size and arrays with nonconsecutive elements.]

[Editor: Delete unresolved issue note 259.] — 87:1-7

If a *namelist-group-object* is polymorphic or has an ultimate component that is allocatable or a pointer, that object shall be processed by a user-defined derived-type input/output procedure (9.5.4.3.3). — 193:7+ New ¶

[Editor: "nonconstant ... bound" ⇒ "character length or array bound that is specified by an expression that is not an initialization expression".] — 400:10

3 More lenient alternative edits

Constraint: A *namelist-group-object* shall not be an assumed-size array, an assumed-shape array, or a pointer. — 86:39-42

[The constraint avoids objects of unknowable size and arrays with nonconsecutive elements.]

[Editor: Delete unresolved issue note 259.] 87:1-7

[Editor: After the comma insert "every allocatable *namelist-group-object* in the namelist group shall be allocated, and".] 193:7

If a *namelist-group-object* is polymorphic or has an ultimate component that is allocatable or a pointer, that object shall be processed by a user-defined derived-type input/output procedure (9.5.4.3.3). 193:7+ New ¶

[Editor: "nonconstant ... bound" ⇒ "character length or array bound that is specified by an expression that is not an initialization expression".] 400:10

4 Even more lenient alternative edits

Constraint: A *namelist-group-object* shall not be an assumed-size array. 86:39-42

[The constraint avoids objects of unknowable size.]

[Editor: Delete unresolved issue note 259.] 87:1-7

[Editor: After the comma insert "every allocatable *namelist-group-object* in the namelist group shall be allocated, every *namelist-group-object* that is a pointer shall be associated with a target, and".] 193:7

If a *namelist-group-object* is polymorphic or has an ultimate component that is allocatable or a pointer, that object shall be processed by a user-defined derived-type input/output procedure (9.5.4.3.3). 193:7+ New ¶

[Editor: "nonconstant ... bound" ⇒ "character length or array bound that is specified by an expression that is not an initialization expression".] 400:10

5 An even more lenient alternative without edits

It is possible to allow unallocated allocatable *namelist-group-object*s or pointer *namelist-group-object*s for namelist input, so long as no value is specified for them in the input record. I decided that this one is so unlikely to get adequate support that I didn't develop edits for it.