

Subject: Allow a polymorphic allocatable *variable* in intrinsic assignment
 From: Van Snyder
 Reference: 04-354r1

1 Number

2 TBD

3 Title

4 Allow a polymorphic allocatable *variable* in intrinsic assignment.

5 Submitted By

6 J3

7 Status

8 For consideration.

9 Basic Functionality

10 Allow a polymorphic allocatable *variable* in intrinsic assignment.

11 Rationale

12 We allow intrinsic assignment to change the extent or length type parameters of an allocatable *variable*
 13 in an assignment. Why not the type if it's also polymorphic? Surely that would be equally valuable.
 14 After all, it can be done with a DEALLOCATE statement followed by an ALLOCATE statement with
 15 a SOURCE= specifier. Why not allow the clearer intrinsic assignment statement?

16 Estimated Impact

17 Trivial for the standard, probably in the trivial-to-small range for processors. Estimated at meeting 169
 18 to be at 4 on the JKR scale.

19 Detailed Specification

20 Allow a polymorphic allocatable *variable* in intrinsic assignment. Require *expr* to have the same rank
 21 as a polymorphic *variable*. If *variable* is allocated and polymorphic, deallocate it (as we do now if any
 22 bounds or length type parameters differ). Then allocate it with the same dynamic type, type parameters
 23 and bounds as *expr*. Or, we could say "if the dynamic type of *variable* is different from the dynamic type
 24 of *expr*, deallocate *variable*." It is necessary to specify which one because if we do the first, processors
 25 could not choose to do the second as a polite optimization without making finalization indeterminate.
 26 There could be a middle ground that mumbles about finalizers.

27 Here are illustrative edits, to give an idea of the effect on the standard.

28 [Replace "*variable* shall not be polymorphic" by "if *variable* is polymorphic it shall be allocatable".] 138:18

29 [Replace the first line of the "derived type" row and "Type of *expr*" column of Table 7.8 with "if *variable* 139:2+8
 30 is polymorphic it shall be TKR compatible (5.1.1.2) with *expr*; if *variable* is not polymorphic *expr* shall
 31 have the same declared type and same kind type parameter values as *variable*;".]

32 [Replace "or" by a comma; insert ", or if *variable* has a dynamic type different from the dynamic type 139:23
 33 of *expr*" after "differ".]

- 1 [Delete “and”.] 139:25

- 2 [Insert “, and with the same dynamic type as *expr* if *variable* is polymorphic” after “LBOUND(*expr*)”.] 139:26
- 3 **History**