Subject: Automatic polymorphic variables
From: Van Snyder

# 1 Introduction

In Fortran 2003 we left polymorphic assignment to Fortran 2008. We were fortunately able to develop a compatible extension. Automatic variables can automatically get values for length parameters or array bounds, but cannot get their type. Variables of the same type as the dynamic type of a dummy argument or a module variable would be useful. There is no obvious extension that is compatible with MOLD=, so if we anticipate doing this, now is the time to do it (assuming we don't want a kludge). As things stand, the only way to get them is to allocate them. Automatic variables are simpler to program, and might have a slight edge of performance over explicitly allocated ones because the processor might invoke the memory manager only once when the specifcation part is elaborated to allocate space for all local variables, automatic or otherwise.

# 2 Proposal

Allow TYPEOF(*variable*) in place of *declaration-type-spec* except in IMPLICIT, or in place of *type-spec* in an array constructor or ALLOCATE statement. Delete MOLD= from the ALLOCATE statement.

# 3 Edits

Edits refer to 07-007r1. 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 associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [ and ] in the text.

---

**or** TYPEOF ( *variable* ) 47:22+

---

**or** TYPEOF ( *variable* ) 47:28+

---

[Editor: Append a sentence to the paragraph:] 47:37
If TYPEOF(*variable*) appears the declared and dynamic types and type parameters specified are those of *variable*.

---

C440a (R441) If *declaration-type-spec* is TYPEOF(*variable*), *variable* shall not be polymorphic and its type parameter values shall be specified by initialization expressions. 65:7+

---

[Editor: Insert "If *type-spec* appears and is TYPEOF(*variable*) the array constructor is polymorphic if and only if *variable* is polymorphic. Every type parameter of *variable* shall have a defined value." before "If". Insert "and is not TYPEOF(*variable*)" after "appears".] 84:14

---

[Editor: Replace "this type and type parameters" by "the type and type parameters of the *array-constructor*".] 84:15-16

---

[Editor: Insert "If TYPEOF(*variable*) appears the entities are polymorphic if and only if *variable* is polymorphic. Every type parameter of *variable* shall have a defined value. If TYPEOF does not appear, the". before second "The"] 87:15-16

---

C503a (R503) If *declaration-type-spec* is TYPEOF(*variable*) and *variable* is polymorphic, *object-name* shall not be the name of a dummy argument or function result variable. 88:8+

---

[Editor: Insert ", or if *declaration-type-spec* is TYPEOF(*variable*) and *variable* is polymorphic or has a type parameter that is assumed, deferred, or specified by a specification expression" at the end of the sentence.] 88:36

---

C513a An automatic data object shall not have the ALLOCATABLE, EXTERNAL, INTRINSIC, PARAMETER, POINTER or SAVE attribute. 88:37+

---

C579a (R560) The *declaration-type-spec* shall not be TYPEOF(*variable*). 110:7+

[Editor: Delete.] 127:13

If *type-spec* appears and consists of TYPEOF(*variable*), the value of *variable* need not be defined, but the values of its type parameters shall be defined. 130:15

C1215a (R1212) If *declaration-type-spec* is TYPEOF(*variable*), *variable* shall not be polymorphic. 306:19+

C1248a (R1226) If *declaration-type-spec* is TYPEOF(*variable*), *variable* shall not be polymorphic. 326:30+