Subject: Use ALLOCATABLE and POINTER attributes in generic resolution
From: Van Snyder
Reference: 03-258r1, section 2.3.2

Number

TBD

Title

Use ALLOCATABLE and POINTER attributes in generic resolution.

Submitted By

J3

Status

For consideration.

Basic Functionality

Use ALLOCATABLE and POINTER attributes in generic resolution.

Rationale

I have a generic interface `Allocate_Test` that allocates an object, tests the status, and prints an error message if an error occurs. I cannot have a specific interface for a to-be-allocated argument that has the POINTER attribute, and another for a to-be-allocated argument with the same type, kind type parameters and rank that has the ALLOCATABLE attribute. So I need different generic names for allocatable objects and pointer objects. If I change an object from pointer to allocatable or vice-versa, I have to track down all of the `Allocate_Test` invocations for that variable and change them to the other one. Avoiding this labor was one of the justifications for the generic facility.

Estimated Impact

Minor. A few lines in 16.2.3.

Detailed Specification

Allow the POINTER and ALLOCATABLE attributes to be used for generic resolution. If the only difference between two specific interfaces is that one has neither the POINTER nor ALLOCATABLE attribute for some argument, and the other one has one of those attributes for the corresponding argument, the interface is ambiguous. If one has the POINTER attribute and the corresponding one has the ALLOCATABLE attribute the interface is not ambiguous, at least so far as that pair of specific procedures is concerned.

If a dummy argument has the POINTER or ALLOCATABLE attribute, the corresponding actual argument is required to have the same attribute. Therefore this change would not invalidate any existing program.

History