Subject: Default initial values for optional dummy arguments
From: Van Snyder
References: 03-258r1, section 2.4.3.2, 04-179, 04-338, 04-353, 04-354

1 Number

TBD

2 Title

Default initial values for optional dummy arguments.

3 Submitted By

J3

4 Status

For consideration.

5 Basic Functionality

Default initial values for optional dummy arguments.

6 Rationale

A frequently requested feature is to be able to specify a default initial value for absent optional dummy arguments.

7 Estimated Impact

Minor, mostly in Section 12, unless the ramifications remarked below in Section 8.1 are attributed to this proposal. That would make this proposal a teensy bit bigger. Estimated at J3 meeting 169 to be at 4 on the JKR scale.

8 Detailed Specification

Provide a specification for a default initial value for absent optional dummy scalars, and arrays that are not assumed-size arrays. The specification has almost exactly the same syntax as an initialization, the differences being that the expression need not be an initialization expression, and that pointer initialization need not be NULL(); indeed, the target need not have the SAVE attribute.

If an optional dummy argument has a default initialization specified and the associated actual argument is absent, the effect is as if the initializer were evaluated in the context of the invocation of the procedure, and then became associated with the dummy argument as if it were the actual argument, and the dummy argument had the VALUE attribute as well. This implies that optional dummy arguments with initializers cannot have INTENT(INOUT) or INTENT(OUT). Two initializers shall not depend on each other, either directly or indirectly — because they can't be evaluated. In particular:

(1) Its assumed and deferred nonkind type parameters and extents, and dynamic type if it is polymorphic, are taken from the initializer. This is an extension beyond the behavior of VALUE.

(2) If it is not a pointer the value is assigned as if by intrinsic assignment.

(3) If it is allocatable, it is assumed to be unallocated before the magic happens. This is an extension beyond the behavior of VALUE.

(4) If it is a pointer, the default initializer shall have the TARGET attribute, and the association is established as if by pointer assignment. This is an extension beyond the behavior of VALUE.

(5) The PRESENT intrinsic never returns false, or maybe is prohibited, since the initializer is in effect an actual argument.

8.1 Remarks

Once this is done, the proposal in 04-338 to remove restrictions on VALUE might as well be implemented. Actually, 04-338 would need to be extended a little bit more, to allow ALLOCATABLE and POINTER to coexist with VALUE. The behavior of those extensions is obvious. 04-353 and 04-354 might as well be done, too. That way, default initialization for optional dummy arguments, intrinsic assignment and VALUE would all work consistently.

9 History