Subject: Disassociated or deallocated actual argument associated with nonpointer nonallocatable
           optional dummy argument is considered not to be present
From: Van Snyder
References: J3-016, 03-258r1, section 2.4.3.1, 04-178, 04-385

1   Number

TBD

2   Title

Disassociated or deallocated actual argument associated with nonpointer nonallocatable optional dummy
argument is considered not to be present.

3   Submitted By

J3

4   Status

For consideration.

5   Basic Functionality

A disassociated or deallocated actual argument associated with a nonpointer nonallocatable optional
dummy argument is considered not to be present.

6   Rationale

I have a procedure that has numerous pointers that are associated, or not, depending on input parame-
ters. These are in turn passed to some subroutines that do (or do not do) some computations depending
on whether the dummy arguments are associated. This, of course, requires the dummy arguments to
be pointers, which degrades the generality of the procedure, and may have undesirable implications for
optimization.

It would be better if I could use optional nonpointer dummy arguments to decide whether to do these
computations. In order to achieve this effect by using n optional dummy arguments I need an IF ...
ELSE IF ... END IF construct with 2^n branches.

7   Estimated Impact

Nearly trivial — a minor change in definition of "present". Estimated at meeting 169 to be 4 on the
JKR scale.

8   Detailed Specification

Allow to associate a deallocated allocatable actual argument, or a disassociated pointer actual argument,
with an optional nonpointer nonallocatable dummy argument, in which case the dummy argument is
considered to be absent. This applies both to dummy data objects and dummy procedures. In Fortran
2003, this is an error.

8.1   Suggested edits

The following suggested edits illustrate the magnitude of the proposal. Change bars reflect differences
from 04-385.

8.1.1 Allow disassociated or deallocated actual data object with optional nonpointer nonallocatable dummy

[Edits in 04-385 led to saying "Except in references..." three times; which is tedious; factor it:]   269:20-24
Except in references to intrinsic inquiry functions:

     (1)   If a nonoptional nonpointer dummy argument corresponds to a pointer actual argument, the actual argument shall be associated with a target and the dummy argument becomes associated with that target.

     (2)   If an optional nonpointer dummy argument corresponds to a pointer actual argument, the pointer association status of the actual argument shall not be undefined. If the actual argument is associated with a target the dummy argument becomes associated with that target.

     (3)   If a nonoptional nonallocatable dummy argument corresponds to an allocatable actual argument the actual argument shall be allocated.

8.1.2 Allow disassociated actual procedure pointer with optional nonpointer dummy

[An oversight in 04-385: Replace "an associated" by "a".]   271:16

[An oversight in 04-385: Add a sentence at the end of the paragraph:]   271:19
Except in references to intrinsic inquiry functions, if the dummy argument is not optional and the corresponding actual argument is a procedure pointer, the actual argument shall be pointer associated with a procedure. If the dummy argument is optional and the corresponding actual argument is a procedure pointer, the pointer association status of the actual argument shall not be undefined.

8.1.3 Expand definition of "absent"

[Replace "A dummy" by "An optional dummy" and "a dummy" by "an optional dummy".]   272:26

[Delete "or".]   272:28

[Replace period by ", or".]   272:29

     (3)   does not have the ALLOCATABLE or POINTER attribute and is associated with an actual   272:29+
argument that

         (a)   has the ALLOCATABLE attribute and is not allocated, or
         (b)   has the POINTER attribute and is not pointer associated; the pointer association status of the associated actual argument shall not be undefined.

9 History

m166 03-258r1, section 2.4.3.1
m167 04-178
m170 04-385