

Subject: Compound assignment/operation generics would be useful
From: Van Snyder
Reference: 03-258r1, section 2.3.4

1 **Number**

2 TBD

3 **Title**

4 Compound assignment/operation generics would be useful.

5 **Submitted By**

6 J3

7 **Status**

8 For consideration.

9 **Basic Functionality**

10 Provide compound assignment/operation generics.

11 **Rationale**

12 Some applications have complicated derived-type objects on which one wishes to define operations and
13 assignment. In these cases, the result of the function that defines the operation will be an anonymous
14 object of a derived type. Finalizers help to get these to work correctly, but don't address the performance
15 problems that arise as a consequence of separating defined assignment from the defined operation, espe-
16 cially if assignment is a "deep copy." These could be ameliorated if a compound assignment/operation
17 generic interface could be defined.

18 **Estimated Impact**

19 Minor.

20 **Detailed Specification**

21 Define a new variety of *generic-spec* that specifies compounded assignment and operation, e.g., COM-
22 POUND(=, .MYMULT.). The first thing-o would have to be "=" so it may not be necessary to say so.
23 On the other hand, saying so leaves room to extend it to pointer assignment. The *generic-spec* could be
24 used in an interface block or a GENERIC statement in a type definition.

25 These defined operations compounded with assignment would be used in statements of the form *variable*
26 = *expr* .MYMULT. *expr* or *variable* = .MYUNARY. *expr*.

27 Require all of the procedures named or described by the interface to be subroutines with two or three
28 arguments, with the first becoming associated with the *variable* and the second (and third) becoming
29 associated with the *expr*(s). Also see the proposals for partial application and optional arguments for
30 subprograms that define operations or assignment, which would have impact on this specification.

31 **History**