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

Number

TBD

Title

Compound assignment/operation generics would be useful.

Submitted By

J3

Status

For consideration.

Basic Functionality

Provide compound assignment/operation generics.

Rationale

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

Estimated Impact

Minor.

Detailed Specification

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

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

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

History