

Subject: Resolve generic without invoking a procedure or evaluating arguments
From: Van Snyder

1 **Number**

2 TBD

3 **Title**

4 Resolve generic without invoking a procedure or evaluating arguments

5 **Submitted By**

6 J3

7 **Status**

8 For consideration.

9 **Basic Functionality**

10 Given exemplars of actual arguments, resolve a generic name to a specific procedure without invoking
11 the procedure or evaluating its arguments.

12 **Rationale**

13 With care and diligence, one can develop a program so that related sets of variables, constants and
14 function results are parameterized by a single kind type parameter. In order to change the kind of that
15 set of entities, one need only change one named constant's definition — almost: Generic procedures
16 cannot be actual arguments or procedure pointer targets. Thus, if one needs to change the program, in
17 addition to changing the single named constant definition, one needs to find all places where a specific
18 procedure that operates on the entities in question is an actual argument or procedure pointer target,
19 and manually edit those appearances.

20 It would be helpful to have a facility to resolve a generic name to a specific procedure without evaluating
21 any arguments or invoking a procedure.

22 **Estimated Impact**

23 Minor. Processors already know how to do generic resolution.

24 **Detailed Specification**

25 Given exemplars of actual arguments, resolve a generic name to a specific procedure without invoking
26 the procedure or evaluating its arguments.

27 There are at least two ways to do this. One is to provide a syntax that is suggestive of procedure
28 reference, but does resolution instead. One possibility for this is to enclose an actual argument list in
29 square brackets or curly brackets instead of round brackets.

30 Another is to provide an entity that looks like an intrinsic function but that has the important distinction
31 that its arguments aren't evaluated. Indeed, this entity that has the appearance of a function reference
32 isn't even invoked during program execution. It is entirely resolved to a procedure by the processor
33 during translation.

34 No matter what syntax is used, it should be allowed to use the result either as an actual argument or a
35 procedure pointer target.

1 It is conceivable that a provision could be made to resolve a generic name from the context of its
2 appearance. This could work if it is an actual argument associated with a dummy procedure provided
3 that both the referenced procedure and the dummy procedure have explicit interface, or if it is a target
4 in a procedure pointer assignment and the pointer has explicit interface. This would still require some
5 means to cause resolution in the implicit interface cases, so it may not be worth contemplating.

6 **History**