

Subject: Embed a decision within an expression  
From: Van Snyder  
Reference: 03-258r1, section 2.8.1; 04-192, 04-357r1 04-393

1 **Number**

2 TBD

3 **Title**

4 Embed a decision within an expression.

5 **Submitted By**

6 J3

7 **Status**

8 For consideration.

9 **Basic Functionality**

10 Provide a mechanism to embed a decision within an expression.

11 **Rationale**

12 If one needs to embed one of several values within an expression, depending upon a condition, the  
13 available methods are

- 14 (1) Nearly duplicate the statement containing the expression with a different entity within the  
15 different instances and select between them using the condition,
- 16 (2) Copy one of the values into a temporary variable depending upon the condition and use  
17 that temporary variable within the expression, or
- 18 (3) Associate a pointer with one of the variables depending on the condition — if the entities  
19 to be selected are variables.

20 In the first case, assuming  $m$  alternatives for each of  $n$  entities within the statement,  $m^n$  instances of  
21 the statement are needed. In the second case, the program must endure the expenses of an auxiliary  
22 variable and the time to copy it, both of which could be substantial in the array case. In the third case,  
23 which only works if the values are variables, the variables have to have the pointer or target attribute,  
24 which could have negative implications for optimization.

25 The same considerations apply if one needs to associate one of several entities as an actual argument.  
26 It's even worse in the case of using a temporary variable if the associated dummy argument does not  
27 have INTENT(IN), because one needs to do the selection and copying twice.

28 It would be useful to be able to embed a decision within an expression. This would avoid all of the  
29 problems outlined above.

30 In the case of an actual argument or a target in a pointer assignment, it would be useful to allow  
31 procedures or procedure pointers to be the alternative entities.

32 **Estimated Impact**

33 This is part of the proposal in J3 paper 04-393, whose antecedent was 04-357r1. At J3 meeting 169, the  
34 proposal in 04-357r1 was judged to be at 4 on the JKR scale. Surely this proposal, being only part of  
35 the previous one, is not larger.

## 1 Detailed Specification

2 Provide a mechanism, called here a “conditional form,” to embed a decision within an expression. The  
3 syntax shall provide for a condition and two alternative entities.

4 If an actual argument is a conditional form, and the alternatives are either variables or procedures,  
5 to be useful the selected entity itself, rather than the value of it, must become the actual argument.  
6 Indeed, in the case of an argument that is a procedure, it does not have a value. No matter what  
7 syntax is used, if there is no special description it cannot be called a function reference or an operation,  
8 because the result would be a value separate from the desired entity itself. One might think that it  
9 would be acceptable to call it a function reference or operation that returns a pointer associated with  
10 the appropriate alternative entity, but that won't work either, because of the absurd outcome of interp  
11 F95/74. That would also require the alternatives to have the target attribute, which would have negative  
12 implications for optimization. Maybe a new term that doesn't carry the baggage of “pointer” or “target,”  
13 such as “referent,” could be used to describe the result, allowing the form to be called a function reference  
14 or an operation. That seems like a lot of work to invest for one purpose.

15 It is important that the alternative that is not selected, and all expressions within it, are not evaluated.  
16 The condition that selects between the alternatives may be a proxy indicating the nonexistence of values  
17 necessary for these evaluations.

18 If one of the alternative entities has the pointer attribute and the other does not, the one that does not  
19 shall have the target attribute, and the result has the target attribute. If the alternative that has the  
20 pointer attribute is selected, the result is its target.

21 Otherwise, the alternative entities shall have the same characteristics. If one of them is the NULL  
22 intrinsic without an argument, the result characteristics are inferred from the other. If they are both  
23 the NULL intrinsic without an argument, the result characteristics are inferred from the context of the  
24 appearance of the conditional form, exactly as would be the case if the NULL intrinsic had appeared on  
25 its own instead.

## 26 Examples

27 Several syntaxes are possible. Two under consideration are similar to a function reference, and similar  
28 to an operator.

29 `A = B + IF ( X > 0.0, SQRT(X), 0.0 )`

30 or

31 `A = B + X > 0.0 ? SQRT(X) : 0.0`