Subject: Allow declarations of local entities within constructs
From: Van Snyder
Reference: 03-258r1, section 2.1.1

Number

TBD

Title

Allow declarations of local entities within constructs.

Submitted By

J3

Status

For consideration.

Basic Functionality

Allow local declarations within constructs.

Rationale

Some entities are needed only for local purposes. Unfortunately, they have to be declared at the scope of the subprogram — or larger. The fact that a variable with local temporary usage exists throughout a procedure increases maintenance costs. When one sees an assignment to it, one needs to ask "what references does this definition reach?" If the existence of the variable could be confined to a more local extent, the answer to the question would similarly be confined.

Estimated Impact

Small.

Detailed Specification

The simplest way to do this is to specify that constructs are scoping units, and to allow a *specification-part* in each one. The words in Section 16 then mostly take care of things.

In IF or SELECT constructs, each branch should be a separate scoping unit.

For DO or FORALL constructs, declaration of a variable having the same name as the DO induction variable or a FORALL index variable should be a declaration of that variable, not a new one.

The effect of the SAVE attribute on a variable declared in a construct can be decided later. Does it exist for the duration of the program's execution, or only for the duration of the procedure instance's execution? The former is probably more useful.

An additional felicity would be to allow a type specification for the induction variable for a DO construct within the DO statement, or for an index variable of a FORALL statement or construct within the *forall-header*, e.g.

```
integer, parameter :: KBI = selected_integer_kind(7)
do integer(kbi) :: i = 1, 9999999
...
end do
```

```
forall ( integer(kbi) :: i = 1, 9999999 )
...
end forall

forall ( integer(kbi) :: i = 1, 9999999, a(i) > 0.0 ) b(i) = log(a(i))
```

History