Subject: Revised specs and second draft of edits for size of parameter array
From: Van Snyder
Reference: 01-189, 04-101, 04-394r1, 05-129, 05-189, WG5/N1626-J3-023

1 Specifications as revised at Delft

Allow an array named constant to get its extents from the extents of its *initialization-expr*. The ranks of the named constant and its *initialization-expr* shall be the same.

2 Syntax

Use an asterisk for the upper bound of every dimension in the declaration of the named constant.

3 Edits

Edits refer to 04-007. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

[Editor: Insert an additional right-hand-side for *array-spec* (R510):] 78:13+

or *implied-shape-spec-list*

[Editor: Add an example to Note 5.11:] 78:14+8+

```
REAL, PARAMETER :: V(0:*) = [0.1, 1.1] ! Implied-shape array
```

[giving]

NOTE 5.11

Examples of DIMENSION attribute specifications are:

```
SUBROUTINE EX (N, A, B)
   REAL, DIMENSION (N, 10) :: W           ! Automatic explicit-shape array
   REAL A (:), B (0:)                     ! Assumed-shape arrays
   REAL, POINTER :: D (:, :)              ! Array pointer
   REAL, DIMENSION (:), POINTER :: P      ! Array pointer
   REAL, ALLOCATABLE, DIMENSION (:) :: E  ! Allocatable array
   REAL, PARAMETER :: V(0:*) = [0.1, 1.1] ! Implied-shape array
```

[Insert a new subclause immediately before **5.1.2.6 EXTERNAL attribute**:] 80:35+

5.1.2.5.5 Implied-shape array

An **implied-shape array** is a named constant that takes its shape from the *initialization-expr* in its declaration. An implied-shape array is declared with an *implied-shape-spec-list*.

R516$\frac{1}{2}$ *implied-shape-spec* **is** [*lower-bound*:]*

C544$\frac{1}{2}$ (R516$\frac{1}{2}$) An implied-shape array shall be a named constant.

The rank of an implied-shape array is the number of *implied-shape-spec*s in the *implied-shape-spec-list*.

The extent of each dimension of an implied-shape array is the same as the extent of the corresponding dimension of the *initialization-expr*. The lower bound of each dimension is *lower-bound*, if it appears, and 1 otherwise; the upper bound is one less than the sum of the lower bound and the extent.