Subject: Minor change in M4 to allow different-rank views of array objects
From: Van Snyder
References: 00-255, 00-337

1 Edits

Edits refer to 00-007r3. 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 immediately following text, while a page and line number followed by + (-) indicates that immediately following text is to be inserted after (before) the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

R736 *bounds-spec* **is** *lower-bound* : [*upper-bound*] 139:5

Constraint: If *upper-bound* is specified for any dimension of *pointer-object* it shall be specified for all dimensions. 139:12+

[Editor: "The" ⇒ "If an *upper-bound* is specified, the *target* shall have rank one; otherwise, the".] 139:17

If an *upper-bound* is specified, the size of the *target* shall not be less than the size of the *pointer-object*. The elements of the target of *pointer-object*, in array element order (6.2.2.2), are the first SIZE(*pointer-object*) elements of the *target*. 140:11+ New ¶

[Editor: "The" ⇒ "If no *upper-bound* is specified, the".] 140:12

NOTE 7.49$\frac{1}{3}$ 140:37+

It is possible to obtain high-rank views of (parts of) rank-one objects by specifying upper bounds in pointer assignment statements. Consider the following example, in which a matrix is under consideration. The matrix is stored as a rank-one object in MYDATA because its diagonal is needed for some reason – the diagonal cannot be gotten as a single object from a rank-two representation. The matrix is represented as a rank-two view of MYDATA.

```
real, target :: MYDATA ( NR*NC )      ! An automatic array
real, pointer :: MATRIX ( :, : )      ! A rank-two view of MYDATA
real, pointer :: VIEW_DIAG ( : )
MATRIX( 1:NR, 1:NC ) => MYDATA        ! The MATRIX view of the data
VIEW_DIAG => MYDATA( 1::NR+1 )        ! The diagonal of MATRIX
```

Rows, columns or blocks of the matrix can be accessed as sections of `MATRIX`.