Subject: More general rank remappings
From: Van Snyder
Reference: 03-258r1, section 2.9.2.3

1 Number

2 TBD

3 Title

4 More general rank remappings.

5 Submitted By

6 J3

7 Status

8 For consideration.

9 Basic Functionality

10 More general rank remappings.

11 Rationale

12 Fortran 2003 allows the *data-pointer-object* in a pointer assignment statement to have higher rank than
13 the *data-target* provided both bounds are specified for every dimension of *data-pointer-object* and *data-*
14 *target* has rank one. It would be useful to extend this by allowing to specify both bounds for any
15 consecutive sequence of dimensions of *data-pointer-object* provided the number of dimensions for which
16 both bounds are not specified is one greater than the rank of the *data-target*.

17 Estimated Impact

18 Minor.

19 Detailed Specification

20 Extend pointer assignment by allowing to specify both bounds for any consecutive sequence of dimensions
21 of *data-pointer-object* provided the number of dimensions for which both bounds are not specified is one
22 greater than the rank of the *data-target*.

23 Example:

24 In one application, I have a 3×3 matrix at every point along a path of indeterminate length. For reasons
25 having to do with restrictions in the input/output package I am required to use, I have to store this as a
26 rank-2 array in which the first dimension has extent 9. When It's time to use it — usually in MATMUL —
27 I need to reshape it. It would be more convenient to write `P(:3,:3,:)  =>` `Q` or `P(:3,:3,:)  =>`
28 `Q(:9,:)`. Notice that I cannot write `P(:3,:3,:)  =>` `RESHAPE(Q(:9,:),[9*size(Q,2)])`

29 In conjunction with the proposal to allow any combination of explicit and assumed shape, if P and Q
30 were declared "`real, pointer :: P(3,3,:), Q(9,:)`" it would be nice if I could write simply `P =>`
31 `Q`.

32 History