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

# Number

TBD

# Title

More general rank remappings.

# Submitted By

J3

# Status

For consideration.

# Basic Functionality

More general rank remappings.

# Rationale

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

# Estimated Impact

Minor.

# Detailed Specification

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

Example:

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

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

# History