Subject: Variation on PACK
From: Van Snyder

Number

TBD

Title

Variation on PACK.

Submitted By

J3

Status

For consideration.

Basic Functionality

Provide a variation on PACK that packs in only one dimension, producing a result with the same rank as the primary argument.

Rationale

The PACK function doesn't always do the thing appropriate to a problem. The obvious implementation of an alternative to the function proposed here requires two extra anonymous array temps. A more efficient implementation requires a named array temp or function result, and a loop to fill it. If implemented by a function, a separate function is required for each type, kind and rank of the first argument, and each kind of the other two. Here's one for default REAL and rank 4, default LOGICAL and default INTEGER arguments:

```
  function PACK ( ARRAY, MASK, DIM ) RESULT ( R )
    real, intent(in) :: ARRAY(:,:,:,:)
    logical, intent(in) :: MASK(:)
    integer, intent(in) :: DIM
    real, allocatable :: R(:,:,:,:)
    integer :: I, J, S(4)
    ! Checking of the value of DIM and the extent of MASK omitted
    s = shape(array)
    s(dim) = count(mask)
    allocate ( r(s(1),s(2),s(3),s(4)) ) ! Status should be checked
    j = 0
    select case ( dim )
    case ( 1 )
      do i = 1, size(mask)
        if ( mask(i) ) then
          j = j + 1
          r(j,:,:,:) = array(i,:,:,:)
        end if
      end do
    case ( 2 )
      ! ... obvious implementation for dim == 2, 3, 4
```

```
    end select
  end function PACK
```

The loop for `case ( 1 )` could be written

```
  r = array(pack([(i,i=1,size(mask))],mask),:,:,:)
```

but skeptical users probably don't trust optimizers in all processors that might be used to compile the program to do as good a job as writing out the loop explicitly. In particular, to be prudent, one should assume that evaluating the subscript for the first dimension in this example will require two array temps.

Estimated Impact

Minor both for standard and implementors: One intrinsic function.

Detailed Specification

In addition to the description of the function, a summary is needed in 13.5.13.

13.7.89$\frac{1}{2}$ PACK(ARRAY, MASK, DIM)

Description. Pack an array in one dimension, under control of a mask, giving an array of the same rank.

Class. Transformational function.

Arguments.

ARRAY	may be of any type. It shall not be scalar.
MASK	shall be of type logical. It shall be an array of rank one with the same extent as the DIM dimension of ARRAY.
DIM	shall be a scalar of type integer with a value in the range $1 \leq \text{DIM} \leq n$, where n is the rank of ARRAY.

Result Characteristics. The result is an array of the same type, kind and rank as ARRAY. In every dimension except DIM its extent is the same as the extent in the corresponding dimension of ARRAY. In the DIM dimension its extent is COUNT(MASK).

Result Value. The result has the value ARRAY(:, :, ..., PACK([(I,I=1,SIZE(ARRAY,DIM))], MASK), ..., :, :), where the dimension subscripted by the result of PACK instead of a colon is DIM.

Examples. The value of PACK($\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$, [.TRUE., .FALSE., .TRUE., .FALSE.], 2) is $\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$. The value of PACK($\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$, [.TRUE., .FALSE.], 1) is RESHAPE([1, 2, 3, 4], [1,4]).

If 04-275 proceeds the **Result Value** clause can be simplified:

Result Value. The result has the value ARRAY(:, :, ..., WHERE(MASK), ..., :, :), where the dimension subscripted by the result of WHERE instead of a colon is DIM.

If 04-275 proceeds, the PACK function proposed herein is not as urgently needed. It would still be useful, because it would avoid the need to form the array result of WHERE.

History