Contents

1 Routine/Function Prologues

1.1 Fortran: Module Interface vmix_kinds_and_types

This module contains the declarations for all required vertical mixing data types. It also contains several global parameters used by the vmix package, such as kind numbers and string lengths.

REVISION HISTORY:

```
SVN:$Id: vmix_kinds_and_types.F90 39012 2012-07-28 19:50:12Z mlevy@ucar.edu $
SVN:$URL: https://svn-ccsm-models.cgd.ucar.edu/pop2/branches/vmix_project/source/vmix/vmix
```

USES:

```
uses no other modules
```

DEFINED PARAMETERS:

```
KIND TYPES:


The vmix package uses double precision for floating point computations.

integer, parameter, public ::              &
    vmix_r8 = selected_real_kind(13)

MIXING TYPES:


Want to store type of mixing as an integer, not a string (this saves time
on logical comparisons).

MNL Note: I don't think these values will actually be used in the vmix
          package, it will probably be returned to POP by the next tag.

integer, parameter, public ::   &
    VMIX_USE_CONST = 1          , &
    VMIX_USE_RICH  = 2          , &
    VMIX_USE_KPP   = 3
```

PUBLIC TYPES:

```
vmix_input_type contains every possible necessary input field for all
supported types of mixing.

type, public :: vmix_input_type
    integer                             :: vmix_type   ! Enumerate mixing type
    integer                             :: nlev        ! Number of levels
    integer                             :: deepcell_id ! Index of deepest cell
```

```
    real(vmix_r8)                        :: prandtl     ! Prandtl number

    real(vmix_r8), allocatable    :: viscosity(:)     ! nlev
    real(vmix_r8), allocatable    :: diffusivity(:,:) ! nlev+1 x 2
    real(vmix_r8), allocatable    :: density1(:)      ! nlev
    real(vmix_r8), allocatable    :: density2(:)      ! nlev
end type vmix_input_type
```

vmix_output_type contains tracer diffusion and momentum viscosity for a
single column.

```
type, public :: vmix_output_type
    integer                        :: nlev           ! Number of levels
    real(vmix_r8), allocatable    :: viscosity(:)     ! nlev
    real(vmix_r8), allocatable    :: diffusivity(:,:) ! nlev+1 x 2
end type vmix_output_type
```

vmix_bkgnd_params_type contains the necessary parameters for static
mixing. Background mixing fields can vary from level to level, but for
now we expect constant mixing.

```
type, public :: vmix_bkgnd_params_type
    real(vmix_r8), allocatable :: static_visc(:) ! viscosity
    real(vmix_r8), allocatable :: static_diff(:,:) ! diffusivity
    logical                        :: lconvective_diff
    real(vmix_r8)                :: convect_diff
    real(vmix_r8)                :: convect_visc
end type vmix_bkgnd_params_type
```