```
!-----------------------------------------------------------------------
&namctl        !   Control prints                                  (default: OFF)
!-----------------------------------------------------------------------
   sn_cfctl%l_glochk = .FALSE.    ! Range sanity checks are local (F) or global (T). Set T for debugging only
   sn_cfctl%l_allon  = .FALSE.    ! IF T activate all options. If F deactivate all unless l_config is T
     sn_cfctl%l_config = .TRUE.     ! IF .true. then control which reports are written with the following
       sn_cfctl%l_runstat = .FALSE. ! switches and which areas produce reports with the proc integer settings.
       sn_cfctl%l_trcstat = .FALSE. ! The default settings for the proc integers should ensure
       sn_cfctl%l_oceout  = .FALSE. ! that  all areas report.
       sn_cfctl%l_layout  = .FALSE. !
       sn_cfctl%l_prtctl  = .FALSE. !
       sn_cfctl%l_prttrc  = .FALSE. !
       sn_cfctl%l_oasout  = .FALSE. !
       sn_cfctl%procmin   = 0       ! Minimum area number for reporting [default:0]
       sn_cfctl%procmax   = 1000000 ! Maximum area number for reporting [default:1000000]
       sn_cfctl%procincr  = 1       ! Increment for optional subsetting of areas [default:1]
       sn_cfctl%ptimincr  = 1       ! Timestep increment for writing time step progress info
   nn_print     =    0      !  level of print (0 no extra print)
   nn_ictls     =    0      !  start i indice of control sum (use to compare mono versus
   nn_ictle     =    0      !  end   i indice of control sum        multi processor runs
   nn_jctls     =    0      !  start j indice of control               over a subdomain)
   nn_jctle     =    0      !  end   j indice of control
   nn_isplt     =    1      !  number of processors in i-direction
   nn_jsplt     =    1      !  number of processors in j-direction
   ln_timing    = .false.   !  timing by routine write out in timing.output file
   ln_diacfl    = .false.   !  CFL diagnostics write out in cfl_diagnostics.ascii
/
```

namelist 14.1.: `&namctl`

14.5. Model optimisation, control print and benchmark

Options are defined through the `&namctl` (namelist 14.1) namelist variables.

14.5.1. Vector optimisation

`key_vectopt_loop` enables the internal loops to collapse. This is very a very efficient way to increase the length of vector calculations and thus to speed up the model on vector computers.

14.5.2. Status and debugging information output

NEMO can produce a range of text information output either: in the main output file (ocean.output) written by the normal reporting processor (narea == 1) or various specialist output files (e.g. layout.dat, run.stat, tracer.stat etc.). Some, for example run.stat and tracer.stat, contain globally collected values for which a single file is sufficient. Others, however, contain information that could, potentially, be different for each processing region. For computational efficiency, the default volume of text information produced is reduced to just a few files from the narea=1 processor.

When more information is required for monitoring or debugging purposes, the various forms of output can be selected via the `sn_cfctl` structure. As well as simple on-off switches this structure also allows selection of a range of processors for individual reporting (where appropriate) and a time-increment option to restrict globally collected values to specified time-step increments.

Most options within the structure are influenced by the top-level switches shown here with their default settings:

```
   sn_cfctl%l_allon  = .FALSE.    ! IF T activate all options. If F deactivate all unless l_config is T
     sn_cfctl%l_config = .TRUE.     ! IF .true. then control which reports are written with the following
```

The first switch is a convenience option which can be used to switch on and off all sub-options. However, if it is false then switching off all sub-options is only done if `sn_cfctl%l_config` is also false. Specifically, the logic is:

```
  IF ( sn_cfctl%l_allon ) THEN
    !  set all suboptions .TRUE.
    !  and set procmin, procmax and procincr so that all regions are selected ([0,10000000,1], respectively)
  ELSEIF ( sn_cfctl%l_config ) THEN
    !  honour individual settings of the suboptions from the namelist
  ELSE
    !  set all suboptions .FALSE.
  ENDIF
```

Control print suboptions

```
sn_cfctl%l_glochk
```

Details of the suboptions follow but first an explanation of the stand-alone option: `sn_cfctl%l_glochk`. This option modifies the action of the early warning checks carried out in `stpctl.F90`. These checks detect probable numerical instabilites by searching for excessive sea surface heights or velocities and salinity values outside a sensible physical range. If breaches are detected then the default behaviour is to locate and report the local indices of the grid-point in breach. These indices are included in the error message that precedes the model shutdown. When true, `sn_cfctl%l_glochk` modifies this action by performing a global location of the various minimum and maximum values and the global indices are reported. This has some value in locating the most severe error in cases where the first detected error may not be the worst culprit.

The options that can be individually selected fall into three categories:

1. **Time step progress information**

   ```
   sn_cfctl%runstat, sn_cfctl%trcstat and sn_cfctl%timincr
   ```

   This category includes `run.stat` and `tracer.stat` files which record certain physical and passive tracer metrics (respectively). Typical contents of `run.stat` include global maximums of ssh, velocity; and global minimums and maximums of temperature and salinity. A netCDF version of `run.stat` (`run.stat.nc`) is also produced with the same time-series data and this can easily be expanded to include extra monitoring information. `tracer.stat` contains the volume-weighted average tracer value for each passive tracer. Collecting these metrics involves global communications and will impact on model efficiency so both these options are disabled by default by setting the respective options, `sn_cfctl%runstat` and `sn_cfctl%trcstat` to false. A compromise can be made by activating either or both of these options and setting the `sn_cfctl%timincr` entry to an integer value greater than one. This increment determines the time-step frequency at which the global metrics are collected and reported. This increment also applies to the time.step file which is otherwise updated every timestep.

2. **One-time configuration information/progress logs**

   ```
   sn_cfctl%oceout, sn_cfctl%layout and sn_cfctl%oasout
   ```

   Some run-time configuration information and limited progress information is always produced by the first ocean process. This includes the `ocean.output` file which reports on all the namelist options read by the model and remains open to catch any warning or error messages generated during execution. A `layout.dat` file is also produced which details the MPI-decomposition used by the model. The suboptions: `sn_cfctl%oceout` and `sn_cfctl%layout` can be used to activate the creation of these files by all ocean processes. For example, when `sn_cfctl%oceout` is true all processors produce their own version of `ocean.output`. All files, beyond the the normal reporting processor (narea == 1), are named with a _XXXX extension to their name, where XXXX is a 4-digit area number (with leading zeros, if required). This is useful as a debugging aid since all processes can report their local conditions. Note though that these files are buffered on most UNIX systems so bug-hunting efforts using this facility should also utilise the FORTRAN:

   ```
   CALL FLUSH(numout)
   ```

   statement after any additional write statements to ensure that file contents reflect the last model state. Associated with the `sn_cfctl%oceout` option is the additional `sn_cfctl%oasout` suboption. This does not activate its own output file but rather activates the writing of addition information regarding the OASIS configuration when coupling via oasis and the sbccpl routine. This information is written to any active `ocean.output` files.

3. **Control sums of trends for debugging**

   ```
   sn_cfctl%prtctl, sn_cfctl%prttrc
   sn_cfctl%procmin, sn_cfctl%procmax and sn_cfctl%procincr
   nn_ictls, nn_ictle, nn_jctls, nn_jctle, nn_isplt, nn_jsplt
   ```

   NEMO includes an option for debugging reproducibility differences between a MPP and mono-processor runs. This is somewhat dated and clearly only useful for this purpose when dealing with configurations that can be run on a single processor. The full details can be found in this report: The control print option in NEMO* The switches to

*http://forge.ipsl.jussieu.fr/nemo/attachment/wiki/Documentation/prtctl_NEMO_doc_v2.pdf

activate production of the control sums of trends for either the physics or passive tracers are, in the modern code, the sn_cfctl%prtctl and sn_cfctl%prttrc suboptions, respectively.

In addition, the parameters nn_ictl[se] and nn_jctl[se] are the indices used to bound each 2D or 3D array for the sum control over a sub-domain. For instance if qt is a 2D array the sum will be done so that :

```
total = SUM ( qt(nn_ictls : nn_ictle, nn_jctls : nn_jctle) )
```

If these parameters are set to their default values of zero then the actual bounds used will be repectively - for a given processor area the first and the last point effectively computed by this processor in both direction, i.e without taking into account the any halo strips. If these parameters are different from zero, the control print will be done over the corresponding bounded area of the global domain and it will be usefull only for debugging a mono-processor run not a MPP one. Doing a MPP run in that case is a nonsense. nn_isplt and nn_njsplt correspond to the domain decomposition applied in each direction.

Although, perhaps, of limited use for its original intention, the ability to produce these control sums of trends in specific areas provides another tool for diagnosing model behaviour (or misbehaviour!). If only the output from a select few regions is required then additional options are available to activate options for only a simple subset of processing regions. These are: sn_cfctl%procmin, sn_cfctl%procmax and sn_cfctl%procincr which can be used to specify the minimum and maximum active areas and the increment. The default values are set such that all regions will be active. Note this subsetting can also be used to limit which additional ocean.output and layout.dat files are produced if those suboptions are active.