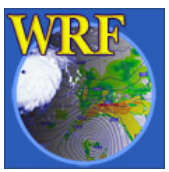
WRF ARW
Runtime Options (namelists)

Wei Wang



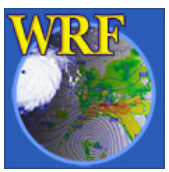
namelist.input

Six namelists:

- **`&time_control`**
- **`&domains`**
- **`&physics`**
- **`&dynamics`**
- **`&bc_control`**
- **`&namelist_quilt`**

As a general rule:

- Multiple columns: domain dependent
- Single column: value valid for all domains



&time_control

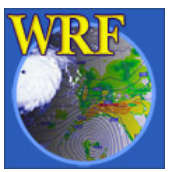
Run time control:

`run_days,run_hours, run_minutes, run_seconds` (WRF coarse grid only)

`start_year, start_day, start_hour, start_minute, start_second, end_year, end_day, end_hour, end_minute, end_second` (real/ndown and WRF, esp. for nest)

Input data interval control:

`interval_seconds` (real/ndown only)



&time_control

Output control:

`history_interval:` output frequency in minutes

`frame_per_outfile:` used to split output files

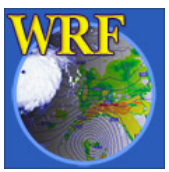
`restart:` whether this is a restart run

`restart_interval:` used to write restart file

`io_form_history/restart/initial/boundary:`

IO format (mostly set to 2 for netCDF; Other options: 1 – binary; 4 – PHDF5; 5 – GriB 1)

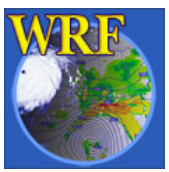
Special MPI output option: 100+io format number – allows one to write one output file per processor. Useful for restart run



&time_control

For a restart run, set:

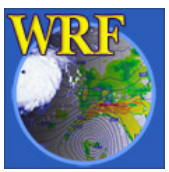
```
start_year, start_day, start_hour,
  start_minute, start_second, end_year
restart = .true.
io_form_restart = 2
```



&time_control

Using new real for SI output, set:

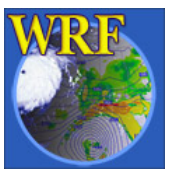
```
auxinput1_inname =
  "wrf_real_em_input.d<domain>.<date>"
```



Sample of Registry.EM

Example 1: time control

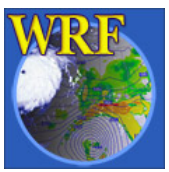
```
rconfig integer history_interval_mo namelist,time_control
   max_domains 0 h "history_interval_mo" "" "MONTHS"
rconfig integer history_interval_d namelist,time_control
   max_domains 0 h "history_interval_d" "" "DAYS"
rconfig integer history_interval_h namelist,time_control
   max_domains 0 h "history_interval_h" "" "HOURS"
rconfig integer history_interval_m namelist,time_control
   max_domains 0 h "history_interval_m" "" "MINUTES"
rconfig integer history_interval_s namelist,time_control
   max_domains 0 h "history_interval_s" "" "SECONDS"
```



Sample of Registry.EM

Example 2: time control

```
rconfig integer history_begin_y namelist,time_control max_domains 0
   h "history_begin_y" "" "YEARS from start of run"
rconfig integer history_begin_mo namelist,time_control max_domains
   0 h "history_begin_mo" "" "MONTHS from start of run"
rconfig integer history_begin_d namelist,time_control max_domains 0
   h "history_begin_d" "" "DAYS from start of run"
rconfig integer history_begin_h namelist,time_control max_domains 0
   h "history_begin_h" "" "HOURS from start of run"
```



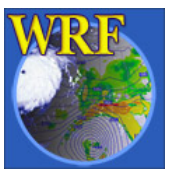
&time_control

Nest input control:

`input_from_file:` whether one would use wrfinput_d0n (n>1) as input.

`fine_input_stream:` how nest domain input are used: = 0 – all input used; = 2 – only static input and masked fields are used

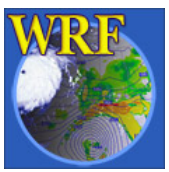
Hint: **`fine_input_stream = 2`** option allows a nest to start at a later time



&time_control

Debug option:

`debug_level:` values from 100 – 500 gives increasing amount of prints



&domains

Time step control:

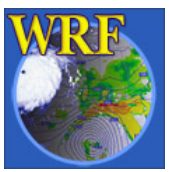
`time_step:` integer

`time_step_fract_num :` numerator for fractional time step

`time_step_fract_den:` denominator for fractional time step

Example: if one would specify time step of 15.5 sec, set

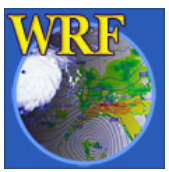
```
time_step = 15
time_step_fract_num = 1
time_step_fract_den = 2
```



&domains

Domain dimension control:

- **`s_we:`** always set to 1
- **`e_we:`** domain dimension in x direction (non-staggered)
- **`s_sn:`** always set to 1
- **`e_sn:`** domain dimension in y direction (non-staggered)
- **`s_vert:`** always set to 1
- **`e_vert:`** domain dimension in z (full η levels)
- **`dx, dy: (dx=dy)`** grid distance in meters
- **`ztop:`** only used in idealized case to set model top



&domains

Nest control:

`max_dom:` how many domains to run

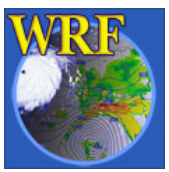
`grid_id, parent_id,`

`i_parent_start, j_parent_start,`

`parent_grid_ratio, parent_time_step_ratio`

`feedback = 0, 1`

`smooth_option = 0, 1 or 2` (applied to parent domain over nest area)

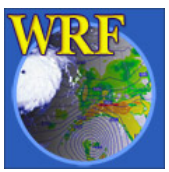


&domains

For new *real* that reads in WPS data:

`num_vert_levels:` how many vertical levels in the incoming data (GFS, NAM, etc.)

`eta_levels:` model vertical coordinate levels, from 1 to 0. This should match the number specified by **`e_vert`** variable. By the release time, option will be available for a user to specify only **`e_vert`**, and *real* will compute a set of reasonable vertical eta levels.



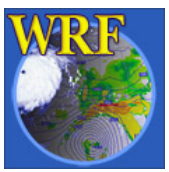
&domains

For new *real* that reads in WPS data:

`interp_type:` whether linear in p (1, default) or log in p (2)

`lagrange_order:` linear (1, default) or quadratic (2) vertical interpolation.

`lowest_lev_from_sfc`: whether the surface fields are forced to be the lowest eta level values. Will be available by release time.



&domains

Moving nest control: (*special compile required*)

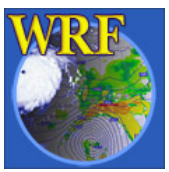
Two options available:

- specified move:

  `num_moves, move_id, move_interval, move_cd_x, move_cd_y`

- automatic move: use a vortex-following algorithm

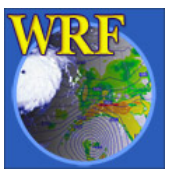
  `vortex_interval` (default 15 min)
  `max_vortex_speed` (default 40 m/s)
  `corral_dist` (default 8 coarse grid cells)



&physics

Seven major physics categories:

- **`mp_physics: 0,1,2,3,4,5,6,8,98,99`**
- **`ra_lw_physics: 0,1,99`**
- **`ra_sw_physics: 0,1,2,99`**
- **`sf_sfclay_physics: 0,1,2`**
- **`sf_surface_physics: 0,1,2,3`** (set before running read or ideal, together with **`num_soil_layers`**)
- **`bl_pbl_physics: 0,1,2,99`**
- **`cu_physics: 0,1,2,3,99`**
- (Note, GFS options not yet ported to ARW)



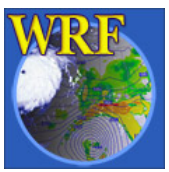
&physics

Physics call time control:

- **`radt:`** for radiation calls (typically 1 min per km)
- **`bldt:`** for surface and PBL calls (typically set to 0)
- **`cudt:`** for cumulus calls (typically every 5 min)

Negative moisture variable control:

- **`mp_zero_out`**: 0, 1, or 2
- **`mp_zero_out_thresh`**: 1E-8



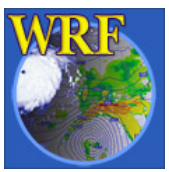
&physics

Other useful ones:

- **`surface_input_source:`** whether to use WPS landuse and soil cat data, or from GriB file
- **`num_soil_layers:`** different values for different sf_surface_physics options (must set before running **`real.exe`**)

Sea-ice temperature control:

- **`seaice_threashold:`** 271 K (default, used in **`real.exe`**)



&physics

Sea-surface temperature update control:

`sst_update:` 0 – no SST update

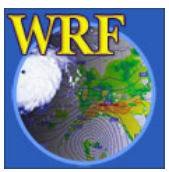
1 – update SST

Set before running **`real.exe`**, and this will create an additional output from **`real.exe`**: wrflowinp_d01

To use the file in **`wrf.exe`**, in **`&time_control`**, add

`auxinput5_inname` = “wrflowinp_d01”,

`auxinput5_interval` = 360,



&physics

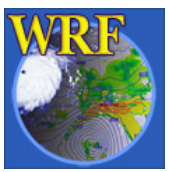
Sensitivity tests:

`isfflx:` 0, or 1

`icloud:` 0, or 1

Grell-Devenyi cumulus scheme control:

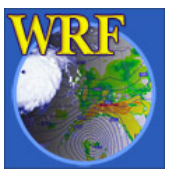
`maxiens, maxens, maxens2, maxens3:`
ensemble member dimensions for multiple closures and multiple parameter controls. Leave them as they are.



&dynamics

Diffusion/filter options:

- **`diff_opt, km_opt:`** typically not required when dx > 10 - 15 km
- **`w_damping:`** real-time only, used to control excessive vertical motion
- **`damp_top, zdamp, dampcoef:`** mostly used in idealized simulations. Do not work for real-data cases. In real-data cases, ptop is recommended to be placed at least at 50 mb (or ~ 20 km)



Recommended Options for dx < 15 km

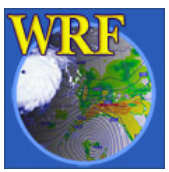
Some explicit diffusion is required, especially under extreme convective conditions:

&dynamics

`diff_opt = 1`

`km_opt = 4`

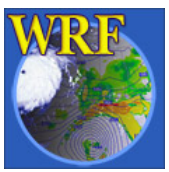
`w_damping = 1` (for real-time runs)



&dynamics

Other options to control various filters:

- **`smdiv:`** divergence damping control (~ 0.1)
- **`emdiv:`** external mode control (~ 0.01)
- **`epssm:`** coeff for vertically implicit off-centered acoustic step (~ 0.1)



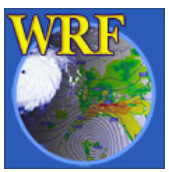
&dynamics

Base state parameter control: (used in *real*)

`base_temp:` default value is 290 K

`base_pres:` default value is 100000 Pa

`base_lapse:` default value is 50 K from 1000 to 400 mb

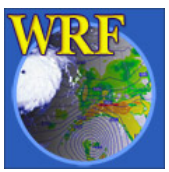


&dynamics

Other options:

`non_hydrostatic:` set to false to enable hydrostatic option

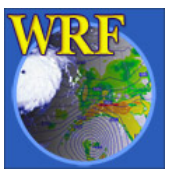
`time_step_sound:` may be altered when time step is very much larger than 6*DX



&bdy_control

Four choices:

- **`Open_xs, open_xe`**
- **`symmetric_xs, symmetric_xe`**
- **`periodic_xs, periodic_xe`**
- **`Specified`** (real-data only, and set before running **`real.exe`**)
  - **`Spec_bdy_width:`** = spec_zone + relax_zone
  - **`spec_zone:`** = 1 (should not change)
  - **`relax_zone:`** = 4 (can be varied)

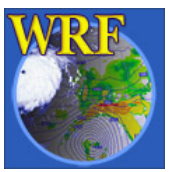


&namelist_quilt

Parallel I/O control:

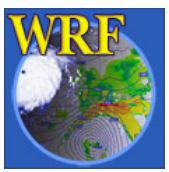
`nio_tasks_per_group (>0):` allow IO to be done on separate processors

`nio_groups (=0):` do not change



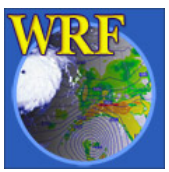
Example for a single domain run: max_dom=1

```
&time_control
run_days                            = 0,
run_hours                           = 12,
run_minutes                         = 0,
run_seconds                         = 0,
start_year                          = 2000, 2000, 2000,
start_month                         = 01,   01,   01,
start_day                           = 24,   24,   24,
start_hour                          = 12,   12,   12,
start_minute                        = 00,   00,   00,
start_second                        = 00,   00,   00,
end_year                            = 2000, 2000, 2000,
end_month                           = 01,   01,   01,
end_day                             = 25,   24,   25,
end_hour                            = 12,   12,   12,
end_minute                          = 00,   00,   00,
end_second                          = 00,   00,   00,
interval_seconds                    = 21600
input_from_file                     = .true.,.true.,.true.,
history_interval                    = 180,  60,   60,
frames_per_outfile                  = 1000, 1000, 1000,
restart                             = .false.,
restart_interval                    = 5000,
io_form_history                     = 2
```



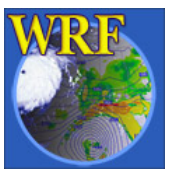
Example for a single domain run: max_dom=1

```
&domains
 time_step                           = 180,
 time_step_fract_num                 = 0,
 time_step_fract_den                 = 1,
 max_dom                             = 1,
 s_we                                = 1,     1,     1,
 e_we                                = 74,    112,   94,
 s_sn                                = 1,     1,     1,
 e_sn                                = 61,    97,    91,
 s_vert                              = 1,     1,     1,
 e_vert                              = 28,    28,    28,
 dx                                  = 30000, 10000,  3333,
 dy                                  = 30000, 10000,  3333,
 num_vert_levels                     = 40,
 eta_levels                          = 1.0, 0.992,  0.980,…
```



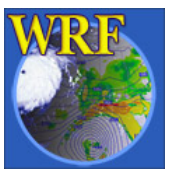
Example for a nested run: max_dom=2

```
&time_control
run_days                            = 0,
run_hours                           = 12,
run_minutes                         = 0,
run_seconds                         = 0,
start_year                          = 2000, 2000, 2000,
start_month                         = 01,   01,   01,
start_day                           = 24,   24,   24,
start_hour                          = 12,   12,   12,
start_minute                        = 00,   00,   00,
start_second                        = 00,   00,   00,
end_year                            = 2000, 2000, 2000,
end_month                           = 01,   01,   01,
end_day                             = 25,   24,   25,
end_hour                            = 12,   12,   12,
end_minute                          = 00,   00,   00,
end_second                          = 00,   00,   00,
interval_seconds                    = 21600
input_from_file                     = .true.,.true.,.true.,
history_interval                    = 180,  60,   60,
frames_per_outfile                  = 1000, 1000, 1000,
restart                             = .false.,
restart_interval                    = 5000,
io_form_history                     = 2
```



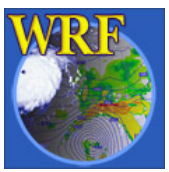
Example for a nested run: max_dom=2

```
&domains
 time_step                           = 180,
 time_step_fract_num                 = 0,
 time_step_fract_den                 = 1,
 max_dom                             = 2,
 s_we                                = 1,     1,     1,
 e_we                                = 74,    112,   94,
 s_sn                                = 1,     1,     1,
 e_sn                                = 61,    97,    91,
 s_vert                              = 1,     1,     1,
 e_vert                              = 28,    28,    28,
 dx                                  = 30000, 10000,  3333,
 dy                                  = 30000, 10000,  3333,
 grid_id                             = 1,     2,     3,
 parent_id                           = 0,     1,     2,
 i_parent_start                      = 0,     31,    30,
 j_parent_start                      = 0,     17,    30,
 parent_grid_ratio                   = 1,     3,     3,
 parent_time_step_ratio              = 1,     3,     3,
 feedback                            = 1,
 smooth_option                       = 0
```



Example for a nested restart run: first run

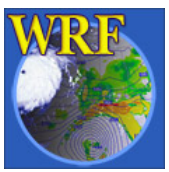
```
&time_control
run_days                            = 0,
run_hours                           = 12,
run_minutes                         = 0,
run_seconds                         = 0,
start_year                          = 2000, 2000, 2000,
start_month                         = 01,   01,   01,
start_day                           = 24,   24,   24,
start_hour                          = 12,   12,   12,
start_minute                        = 00,   00,   00,
start_second                        = 00,   00,   00,
end_year                            = 2000, 2000, 2000,
end_month                           = 01,   01,   01,
end_day                             = 25,   24,   25,
end_hour                            = 12,   12,   12,
end_minute                          = 00,   00,   00,
end_second                          = 00,   00,   00,
interval_seconds                    = 21600
input_from_file                     = .true.,.true.,.true.,
history_interval                    = 180,  60,   60,
frames_per_outfile                  = 1000, 1000, 1000,
restart                             = .false.,
restart_interval                    = 720,
io_form_history                     = 2
io_form_restart                     = 2
```



Example for a nested restart run: first run

- The first model run will write a restart file at hour 12:

`wrfrst_d01_2000-01-25_00:00:00`

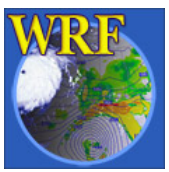


Example for a nested restart run: first run

- The first model run will write a restart file at hour 12:

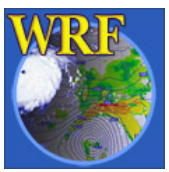
  wrfrst_d01_2000-01-25_00:00:00

- The restart run will read this file to start.



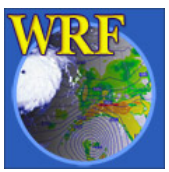
Example for a nested restart run: restart run

```
&time_control
run_days                            = 0,
run_hours                           = 12,
run_minutes                         = 0,
run_seconds                         = 0,
start_year                          = 2000, 2000, 2000,
start_month                         = 01,   01,   01,
start_day                           = 25,   25,   25,
start_hour                          = 00,   00,   00,
start_minute                        = 00,   00,   00,
start_second                        = 00,   00,   00,
end_year                            = 2000, 2000, 2000,
end_month                           = 01,   01,   01,
end_day                             = 25,   24,   25,
end_hour                            = 12,   12,   12,
end_minute                          = 00,   00,   00,
end_second                          = 00,   00,   00,
interval_seconds                    = 21600
input_from_file                     = .true.,.true.,.true.,
history_interval                    = 180,  60,   60,
frames_per_outfile                  = 1000, 1000, 1000,
restart                             = .true.,
restart_interval                    = 720,
io_form_history                     = 2
io_form_restart                     = 2
```



Recommended

Start with the namelist or namelists in a particular test directory, and make modifications.



Check Output

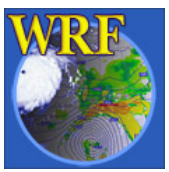
- If one runs the model on a single processor or shared memory machine, it is a good practice to save stardard out/error to a file:

  `wrf.exe >& wrf.out`

- If one runs the model using mpi, the stardard out and error are going to

  `rsl.out.xxxx`

  `rsl.error.xxxx`



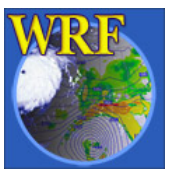
Check Output

- These files provide interesting information about the model run:
  - Whether the model run is successful:

    type **`tail wrf.out`**

    …

    `wrf: SUCCESS COMPLETE WRF`



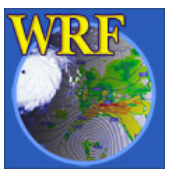
Check Output

– How long does it take the model to integrate one time step?

```
Timing for main: time 2006-01-21_23:55:00 on domain   2:    4.91110 elapsed seconds.
Timing for main: time 2006-01-21_23:56:00 on domain   2:    4.73350 elapsed seconds.
Timing for main: time 2006-01-21_23:57:00 on domain   2:    4.72360 elapsed seconds.
Timing for main: time 2006-01-21_23:57:00 on domain   1:   19.55880 elapsed seconds.
```

– How long does it take the model to write one-time history file ?

```
Timing for Writing wrfout_d02_2006-01-22_00:00:00 for domain        2:    1.17970 elapsed seconds.
Timing for main: time 2006-01-22_00:00:00 on domain   1:   27.66230 elapsed seconds.
Timing for Writing wrfout_d01_2006-01-22_00:00:00 for domain        1:    0.60250 elapsed seconds.
```



Check Output

– If the model has become unstable, it will print out where CFL has been voilated:

```
    5 points exceeded cfl=2 in domain          1 at time    4.200000
 MAX AT i,j,k:          123          48         3 cfl,w,d(eta)=   4.165821
   21 points exceeded cfl=2 in domain          1 at time    4.200000
MAX AT i,j,k:           123          49        4 cfl,w,d(eta)=   10.66290
   27 points exceeded cfl=2 in domain          1 at time    4.250000
MAX AT i,j,k:           123          51        3 cfl,w,d(eta)=   25.43345
   70 points exceeded cfl=2 in domain          1 at time    4.250000
 MAX AT i,j,k:          123          49        3 cfl,w,d(eta)=   35.73432
```