

## Table of Contents

<b>Experiment Configuration</b>	<b>2</b>
Configuration format	2
DEFAULT section	2
Ensemble section	3
Resource section	5
How to create one	5

## Experiment Configuration

In order to design a WRF4G experiment you need to edit a file called `experiment.wrf4g`. From this file, WRF4G will generate all WRF configuration files required to run the simulations planned for the experiment.

### Configuration format

The configuration experiment file consists of sections, each led by a [section] header, followed by `key = value` entries. Allowing sections are [DEFAULT], [ensemble/tag] and [resource/resource\_name]. Lines beginning with # are ignored.

### DEFAULT section

The DEFAULT section provides default values for all other sections. And its key values are:

- **name:** Name of the experiment it **MUST** be unique.
- **max\_dom:** Number of domains.
- **date\_time:** Describe dates for a list of realizations with the following format `start_date | end_date | chunk_size | interval | length:`
  - **start\_date:** Beginning of the list of realizations.
  - **end\_date:** End of the list of realizations.
  - **chunk\_size:** Length (in years, months, days or hours) of a temporal piece of a realization. It means that a `wrf-restart` will be generated every `chunk_size`.
  - **interval:** Interval (in years, months, days or hours) between realizations (optional).
  - **length:** Length (in years, months, days or hours) of each realization (optional).

Examples:

```
date_time = 1979-01-01_06:00:00 | 2010-12-31_18:00:00 | 36 hours | 24 hours | 36 hours
```

```
date_time = 1979-01-01_06:00:00 | 2010-12-31_18:00:00 | 36 hours | 1 days | 36 hours
```

```
date_time = 2011-08-20_12:00:00 | 2011-08-30_00:00:00 | 12 hours
           2011-09-23_12:00:00 | 2011-09-29_00:00:00 | 12 hours
```

- **chunk\_restart:** Either yes or no (default value is yes). If you won't create a restart files when a chunk finishes.
- **calendar:** Either standard or no\_leap (default value is standard). Type of calendar to create realizations.
- **timestep\_dxfactor:** If present, the time step is computed as `dx*timestep_dxfactor`, in kilometers. Defaults to 6, as suggested by the WRF team for most applications. Under some circumstances (cfl problems) a lower value may be needed. In any case, the time step is adjusted to the highest value lower than `timestep_dxfactor` times `dx` fitting evenly in one hour period.
- **np:** Number of processors requested in a parallel job.
- **requirements:** Requirements for computing resources where the experiment is going to run (for more information see [advanced configuration](#)).
- **environment:** Experiment environment variables are configuration options for jobs (for more information see [advanced configuration](#)).
- **clean\_after\_run:** Either yes or no (default value is no), indicating whether or not temporary simulation files should be removed. The maintenance of these files on running place could be desirable for debugging purposes.
- **log\_level:** Indicates the level of the information appended to the log files (default value is INFO). The list of levels available is ERROR, WARNING, INFO and DEBUG.

ulation files should be removed. The maintenance of these files on running place could be desirable for debugging purposes.

- **save\_wps:** Either yes or no (default value is no), indicating whether boundary and initial conditions (`real.exe` output) should be preserved or not. They will be used if the experiment is relaunched again.
- **domain\_path:** Path of the directory with the information about the domain of the simulations; this is, the files generated by `geogrid.exe` (`nameslit.wps` & `geo_em.d[nn].nc`) ([advanced configuration](#)).
- **preprocessor:** Name (just the ending [NAME] section in `preprocessor.[NAME]`) of the necessary pre-processor useful to make the specific input data available for WRF model. Users could not be interested in the permanent WRF-necessary modification of any source of data. With the specification and design of a pre-processor, necessary WRF modifications (e.g.: ASCII to GRIB, variables re-coding, complete input data). Pre-processors are included in `$WRF4G_DEPLOYMENT_DIR/repository/apps/preprocessor/`.
- **preprocessor\_optargs:** Set of values to add extra parameters to the `preprocessor` script. For instance :

```
preprocessor_optargs = member | 0 | 1
                    runtime | 12 | 12
```

- **extdata\_vtable**: Vtable of the `ungrib` module to be used to decode provided GRIB input data.
- **extdata\_path** : Path to the input data. It must be consistent with the pre-processor design ([advanced configuration](#)).
- **extdata\_interval**: Interval of the input data (in seconds). On multiple input data sources use the smallest one.
- **constants\_name**: List of filenames of intermediate-formatted files which contains time-invariant fields used by `metgrid`.
- **postprocessor**: Users might be interested in the transformation of the WRF output files. A first generic post-process of the output will be automatically carried out if a valid name `postprocessor.[NAME]` is provided. Post-processors are included under the `$WRF4G_DEPLOYMENT_DIR/repository/apps/postprocessor/` directory.

**output\_path**: Path to the output experiment files ([advanced configuration](#)). Under this path you will find the following tree directory :

```
[experiment_name]
|-- [realization_name]
    |-- log      ( logs files )
    |-- output   ( output WRF files )
    |-- restart  ( restart WRF files )
    |-- realout  ( real.exe output files )
```

**parallel\_env**: Tag to configure the parallel execution environment (default value is `MPIRUN`), which can be `POE` for IBM's parallel environment, `SRUN` for SLURM batch system, or `MPIRUN` for OpenMPI or MPICH. In case you want to write your own parallel configuration, use `DUMMY` value.

With this option you have to indicate the following variables :

**parallel\_run**: Command to launch parallel applications. Example:

```
parallel_run = mpirun -np 32 -pernode 16 --mca mpi_paffinity_alone 1
```

**parallel\_run\_pernode**: Command to launch one process node for configuration purposes. Example:

```
parallel_run_pernode = mpirun -pernode
```

- **parallel\_real**: Either yes or no (default value is yes), indicating whether the `real.exe` binary is compiled in serial or parallel mode.
  - **parallel\_wrf**: Either yes or no (default value is yes), indicating whether the `wrf.exe` binary is compiled in serial or parallel mode.
- app**: This is to define applications with the following format `tag | type | value` ([advanced configuration](#)):
- **tag**: Name identifier. `wrf_all_in_one` tag is a special name that indicates which is a bundle that includes WRFV3 and WPS files, netcdf, ncos cdos and openmpi libraries.
  - **type**: Either bundle or command.
  - **value**: If type is **bundle** indicates the path of the bundle that can have the following formats `zip`, `tar`, `tar.gz` (or `tgz`), and `tar.bz2` (or `tbz2`). If type is **command**, you can type shell commands to configure netcdf, openmpi or WRF.
- **namelist\_version**: Namelist version to use ( from 3.2 to 3.7.1 ).

**namelist\_values**: Users are able to over-write or add parameters to the namelist file. The parameter's name has to be the same as the `namelist.input` entry. No record specification is necessary if the parameter already appears on the provided WRF-version `namelist.input` template. If not, it should be `record.parameter`.

- **single**: Indicate that namelist parameter of record has a single value. One value per all domains. (Optional)
- **max\_dom**: Indicate that namelist parameter of record has the same value for all the domains of the experiment. (Optional)

Example for one domain :

```
namelist_values = single:ra_sw_physics | 3
                single:ra_lw_physics | 3
```

Example for two domains :

```
namelist_values = ra_sw_physics | 2, 2
                ra_lw_physics | 2, 2
```

## Ensemble section

Using ensemble sections involve creating independent simulations.

Each ensemble section has to begin with the line `[ensemble/tag]` followed by `key = value` entries. This type of section allows to create independent realizations by overloading the key described in `DEFAULT` section, all keys expect name can be overloaded. Thus, ensemble sections are well-suited for sensitivity studies by modifying physical schemes, initial conditions and boundary conditions, and so on. The following experiment configuration example tries to show how to take advantage of this WRF4G feature by setting two ensemble sections with WRF compiled with two different compilers:

```
[DEFAULT]
# Experiment configuration
name = test
# Simulation domain
max_dom = 1
# Experiment time-specification
#           start_date      | end_date      | chunk_size_h
date_time  = 2011-08-28_12:00:00 | 2011-08-30_00:00:00 | 12
calendar   = standard
timestep_dxfactor = 6
# Running options
np = 1
requirements = ARCH = "x86_64"
clean_after_run = yes
save_wps = no
parallel_env = MPIRUN
parallel_real = yes
parallel_wrf = yes
# Vtables must exist as Vtable.[input_extdata]
extdata_vtable = GFS
# Seconds between global analysis input times
extdata_interval = 21600
preprocessor = default
postprocessor = SFC
domain_path = /home/user/repository/domains/Santander_50km
extdata_path = /home/user/repository/input/NCEP/GFS
output_path = /home/user/test/output
# WRF-namelist parameters. Override namelist.input variables here
namelist_version = 3.4.1
namelist_values = spec_bdy_width | 10
                  spec_zone      | 1
                  relax_zone     | 9
                  feedback       | 0
                  history_interval | 180
                  frames_per_outfile | 3
                  e_vert         | 28
                  radt           | 15
                  mp_physics     | 5
                  cu_physics     | 1
                  ra_lw_physics  | 1
                  ra_sw_physics  | 1
                  sf_sfclay_physics | 2
                  bl_pbl_physics | 2
                  sf_surface_physics | 2
                  physics.topo_wind | 3

[ensemble/gfortran]
app = wrf_all_in_one | bundle | /home/user/repository/apps/WRF/WRFbin-3.4.1_gfortran.tar.gz
[ensemble/intel]
app = wrf_all_in_one | bundle | /home/user/repository/apps/WRF/WRFbin-3.4.1_intel.tar.gz
```

## Resource section

Using resource sections do NOT involve creating independent simulations.

Each resource section has to begin with the line `[resource/resource_name]` followed by `key = value` entries. Experiment variables such as `clean_after_run`, `save_wps`, `parallel_environment`, `parallel_real`, `parallel_wrf`, `domain_path`, `preprocessor`, `extdata_path`, `postprocessor`, `app` and `output_path` can be defined independently on each computing resource. Therefore, realizations are able to be simulated concurrently on different resources. For instance, different resources usually have different paths :

```
[DEFAULT]
# Experiment configuration
name                = test
# Simulation domain
max_dom             = 1
# Experiment time-specification
#
#       start_date          | end_date          | chunk_size_h
date_time           = 2011-08-28_12:00:00 | 2011-08-30_00:00:00 | 12
calendar            = standard
timestep_dxfactor   = 6
# Running options
np                  = 1
requirements        = ARCH = "x86_64"
clean_after_run     = yes
save_wps            = no
parallel_env        = MPIRUN
parallel_real       = yes
parallel_wrf        = yes
# Vtables must exist as Vtable.[input_extdata]
extdata_vtable      = GFS
# Seconds between global analysis input times
extdata_interval    = 21600
preprocessor        = default
postprocessor       = SFC
# WRF-namelist parameters. Override namelist.input variables here
namelist_version    = 3.4.1
namelist_values     = spec_bdy_width   | 10
                    spec_zone        | 1
                    relax_zone       | 9
                    feedback         | 0
                    history_interval | 180
                    frames_per_outfile | 3
                    e_vert           | 28
                    mp_physics       | 4
                    radt              | 15
                    ra_lw_physics    | 3
                    ra_sw_physics    | 3

[resource/name_of_resource_1]
app                 = wrf_all_in_one | bundle | /home/user/repository/apps/WRF/WRFbin-3.4.1_r2265_gfortran.tar.gz
domain_path         = /home/user/repository/domains/Santander_50km
extdata_path        = /home/user/repository/input/NCEP/GFS
output_path         = /home/user/test/output

[resource/name_of_resource_2]
app                 = wrf_all_in_one | bundle | /home/user2/repository/apps/WRF/WRFbin-3.4.1_r2265_gfortran.tar.gz
domain_path         = /home/user2/repository/domains/Santander_50km
extdata_path        = /home/user2/repository/input/NCEP/GFS
output_path         = /home/user2/test/output
```

## How to create one

You can write a `experiment.wrf4g` from scratch or use the templates that WRF4G provides (e.g. `single`, `physics` or `default`) by typing :

```
$ wrf4g exp test define --from-template single
$ cat ./test/experiment.wrf4g
[DEFAULT]
# Experiment configuration
name = test
# Simulation domain
max_dom = 1
# Experiment time-specification
#
# start_date | end_date | chunk_size_h
date_time = 2011-08-28_12:00:00 | 2011-08-30_00:00:00 | 12
calendar = standard
timestep_dxfactor = 6
# Running options
np = 1
requirements = ARCH = "x86_64"
clean_after_run = yes
save_wps = no
parallel_env = MPIRUN
parallel_real = yes
parallel_wrf = yes
# Input data
domain_path = /home/user/WRF4G_2_0/repository/domains/Santander_50km
# Vtables must exist as Vtable.[input_extdata]
extdata_vtable = GFS
extdata_path = /home/user/WRF4G_2_0/repository/input/NCEP/GFS
# Seconds between global analysis input times
extdata_interval = 21600
preprocessor = default
# Output
output_path = /home/user/test/output
postprocessor = SFC
# app
app = wrf_all_in_one | bundle | /home/user/wrf4g/repository/apps/WRF/WRFbin-3.4.1_r2265_gfortran.tar.gz
# WRF-namelist parameters. Override namelist.input variables here
namelist_version = 3.4.1
namelist_values = spec_bdy_width | 10
                  spec_zone | 1
                  relax_zone | 9
                  feedback | 0
                  history_interval | 180
                  frames_per_outfile | 3
                  e_vert | 28
                  mp_physics | 4
                  radt | 15
                  ra_lw_physics | 3
                  ra_sw_physics | 3
```