Scientific user will only require to modify '`experiment.wrf4g`' in order to design, manage and develop the scientific experiments. This file configures the experiment. It provides all the parameters to the '`namelist.input`' file of the WRF. It has the following structure:

- **experiment_name:** Name of the experiment. **It has to be unique**. Duplicity of experiment names would give different WRF4G answers:

Domain of simulation

- **max_dom:** Number of domains
- **domain_name:** Name of the folder where there is all the information of the domain of simulation ('`nameslit.wps`' & '`geo_em.d[nn].nc`'). This folder will be find in `${WRF4G_DOMAINPATH}` of the resources.wrf4g? file

Input data

- **extdata_vtable:** Vtable of the `ungrib` module to be used to decode provided GRIB input data
- **extdata_preprocessor:** name (just the ending `[NAME]` section '`preprocessor.[NAME]`') of the necessary pre-processor useful to make the specific input data available for WRF model. User could not be interested on the permanent WRF-necessary modification of any source of data. With the specification and design of a pre-processor, necessary WRF modifications (g.e.: ASCII to GRIB, variables re-coding, complete input data). Pre-processors are included within the `WRF4G-1.0beta.tar.gz` tar file from '`${WRF4G_LOCATION}/repository/apps`'
- **extdata_path :** Path to the data. It must be consistent with the pre-processor design
- **extdata_interval:** Interval of the input data (in seconds). On multiple input data sources place the smallest one

Output data

- **postprocessor:** user migh tbe interested on the transformation of the WRF outputs. A first generic post-process of the output will be automatically done if a valid name `postprocessor.[NAME]` is provided. Post-processors are included within the `WRF4G-1.0beta.tar.gz` tar file from '`${WRF4G_LOCATION}/repository/apps`'

Experiment time-specification

- **start_date:** Beginning of the experiment
- **end_date:** End of the experiment.
- **chunk_size_h:** Length (in hour) of the smallest independent temporal piece of the experiment. It will be the same for all the realizations of the experiment and at the end of each one will be written a `wrf-restart` file in order to continue the experiment with the consecutive chunk

  multiple_dates: Binary flag (0:No, 1:Yes) indicating whether the experiment will be based on independent realizations started at different dates with the same length. When activated it has to be accompanied with appropriated values of:

  - simulation_interval_h: interval (in hour) between independent realizations. It can be specified as multiple values as follows: '`N*chunk_size`'
  - simulation_length_h: length (in hour) of the independent realization. It can be specified as multiple values as follows: '`M*chunk_size`'
- **rerun:** Binary flag (0;No , 1: Yes) indicating whether the experiment should be rerun although experiment has already run (over-writing output files and data-base values)

  multiple_parameters: Binary flag (0;No , 1: Yes) indicating whether the experiment will be based in independent realizations with different values on the WRF-namelist.input file. When activated it has to be accompanied with appropriated values of:

  - multiparams_variables: set of parameters that would be different (some of them or all) between realizations. It is given as a coma-separated WRF-namelist.input files (no record specification is necessary if the parameter already appears on the provided WRF-version namelist.input template?, If not it should be `[parameter]__[record]`)
  - multiparams_combinations: set of values that would take each parameter on the realization (coma-separated). Realizations are separated by '`/`'
  - multiparams_nitems: number of values that should have the namelist parameter. `${max_dom}`, as many values as domains; `1`, a single value
  - multiparams_labels: organization of the output of the experiment will be done with a folder for each realization. Pattern of the folder name of a given realization will be `[start_date]_[multiparams_combinations]('_' separated)`. However, if user provide a set of names ('`/`' separated) for each realization, folder names will be '`[start_date]_[multiparams_labels]`'

Debugging

- **clean_after_run:** binary flag (0: no, 1: yes) indicating whether heavy-staff of the simulation (g.e.: `wrf.exs, rsl.`) should be removed from '`${WRF4G_RUN_LOCAL}`'. It should be desirable for debugging purposes the maintenance of these files on running place. (Default value is 1)
- **save_wps:** binary flag (0: no, 1: yes) indicating whether boundary and initial conditions (`real.exe` output) should be preserved. They will be used if experiment launched again. (Default value is 0)

WRF-namelist parameters

User provides namelist values. They will be over-written on `multiple_parameters` experiments (only the included parameters). User can modify any of the paramters of the namelist. It has to be the same name as in the `namelist.input` with a prefix-flag and the record to its belongs

- **Single valued; NI_[namelist_parameter_name]_ _[record]:** `NI` flag indicating that [namelist_parameter_name] of record [record] has a single value
- **One value per all domains; NIN_[namelist_parameter_name]_ _[record]:** `NIN` flag indicating that [namelist_paramter_name] of record [record] has the same value for all the domains of the experiment
- **One value per domain; NIM_[namelist_name]_ _[record]:** `NIM` flag indicating that [namelist_paramter_name] of record [record] has a different value value for each the domains of the experiment

Others

- **timestep_dxfactor:** If present, the time step is computed as this fator times the dx 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 higest value lower than `timestep_dxfactor` times `dx` fitting evenly in a 1 hour period.

Examples

Multipledates runs

When the `multiple_dates` parameter is set to 0, two additional configuration options (**simulation_interval_h** and **simulation_lenght_h**) are available to set up hindcast runs. Their usage is best understood through a couple of examples. Imagine you are interested in running a 7-day forecast starting every day at 12Z for the period from 1-Jan-2001 to 31-Dic-2001. This can be accomplished by setting:

```
start_date = "2001-01-01_12:00:00"
end_date = "2002-01-01_00:00:00"
is_continuous = 0
 simulation_interval_h = 24
 simulation_lenght_h = 168 # this 7 days in hours
```

and the `wrf4g_submitter.sh` will setup 365 independent runs performing the 7-day forecasts which will be distributed in the grid or in your local cluster.