Wikiprint Book

Title: Experiment Configuration

Subject: TracMeteo - WRF4G2.0/Experiment

Version: 157

Date: 08/12/2022 04:29:04 PM

Table of Contents

Experiment Configuration **3**
Configuration format 3
DEFAULT section 3
Resource section 4
How to create one 4

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. Lines beginning with # are ignored. Allowing sections are `[DEFAULT]` and `[resource_name]`.

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 hours) of a temporal piece of a realization. That means that every chunk_size a `wrf-restart` will be generated.
  - **interval:** Interval (in hours) between realizations (optional).
  - **length:** Length (in hours) of each realization (optional).

    Examples:

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

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

- **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.
- **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.
- **parallel_environment:** Tag to configure the parallel execution environment, which can be `POE` for IBM's Parallel Environment, `SRUN` for SLURM
- **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.
- **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/`.
- **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_member: Set of labels (`|` separated) for multi-member ensemble (e.g. `extdata_member = member0 | member1`). Thus, input data should have the following format :

```
extdata_path/member0
extdata_path/member1
```

- **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 )
```

  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).
- **namelist_label_comb:** Set of labels (`|` separated) for multi-physics combinations defined on `namelist_values`.

  namelist_values: Users are able to over-write or add parameters to the namelist file. The name must 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 physic and one domain :

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

  Example for two physics and two domains :

```
namelist_label_comb =                        phy1 | phy2
namelist_values     = ra_lw_physics         | 2, 2 | 3, 3
                       ra_sw_physics        | 2, 2 | 3, 3
```

  If several physics are described, you have to use the `namelist_label_comb` variable to indicate their names.

Resource section

Each resource section has to begin with the line [resource_name] followed by `key = value` entries. Thus, experiment variables such as `clean_after_run`, `save_wps`, `real_parallel`, `wrf_parallel`, `domain_path`, `preprocessor`, `extdata_path`, `postprocessor`, `app` and `output_path` can be defined depending on each computing resource. Therefore you can simulate your realizations concurrently on different resources.

How to create one

You can write a `experiment.wrf4g` from scratch or use the templates that WRF4G provides 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_environment = MPIRUN
parallel_real       = yes
parallel_wrf        = yes
# Input data
domain_path         = /home/carlos/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
```