

Table of Contents

WRF4G Tutorial part 2	2
How to manage WRF4G errors	2
How to add new computing resources to WRF4G	4
How to add a new WRF geographical domain	5
How to reconfigure the features of an experiment	6
How to resubmit an experiment	6
How to rerun a specific chunk of a realization	7
How to use wrf4g_kill command	7

WRF4G Tutorial part 2

How to manage WRF4G errors

In this section, we are going to see how to manage WRF4G errors. In order to do that, we are going to create a new experiment called `test_1`, based on `test`, in which the `start_date` will be `"2011-08-28_00:00:00"`. Follow the steps below.

```
[user@mycomputer~]$ cd $HOME/WRF4G/experiments

[user@mycomputer~]$ ls
single_test  wrfuc_physics  wrfuc_single_serial

[user@mycomputer~]$ cp -r single_test single_test_1

[user@mycomputer~]$ cd single_test_1

[user@mycomputer~]$ cat experiment.wrf4g | grep "experiment_name"
experiment_name = "test"

[user@mycomputer~]$ cat experiment.wrf4g | grep "start_date="
start_date="2011-08-28_12:00:00"

[user@mycomputer~]$ cat experiment.wrf4g | grep "experiment_name"
experiment_name = "test_1"

[user@mycomputer~]$ cat experiment.wrf4g | grep "start_date "
start_date="2011-08-28_00:00:00"

[user@mycomputer~]$ wrf4g_prepare
Warning: You are using resources.wrf4g located in the /home/carlos/WRF4G/etc/ directory.
Preparing namelist...
WRFV3/run/namelist.input
WRF Check Warning: CAM radiation selected but paerlev/levsiz/cam_abs_dim1/cam_abs_dim2 was not set. Fixing...
WRF Check Warning: radt is shorter than dx (0.500000)

---> Single params run
---> Continuous run
    ---> cycle_chunks: test_1 2011-08-28_00:00:00 2011-08-30_00:00:00
        ---> chunks 1: test_1 2011-08-28_00:00:00 2011-08-28_12:00:00
        ---> chunks 2: test_1 2011-08-28_12:00:00 2011-08-29_00:00:00
        ---> chunks 3: test_1 2011-08-29_00:00:00 2011-08-29_12:00:00
        ---> chunks 4: test_1 2011-08-29_12:00:00 2011-08-30_00:00:00

[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks  Comp.Res  WN      Run.Sta  ext  %
test             2   D   3/3     mycomputer  ciclón  Finished  0 100.00
test_1          -   P   0/4     -          -       Prepared  -  0.00

[user@mycomputer~]$ wrf4g_submit
Submitting realization: "test_1"
    Submitting Chunk 1: 2011-08-28_00:00:00 2011-08-28_12:00:00
    Submitting Chunk 2: 2011-08-28_12:00:00 2011-08-29_00:00:00
    Submitting Chunk 3: 2011-08-29_00:00:00 2011-08-29_12:00:00
    Submitting Chunk 4: 2011-08-29_12:00:00 2011-08-30_00:00:00

[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks  Comp.Res  WN      Run.Sta  ext  %
test             2   D   3/3     mycomputer  ciclón  Finished  0 100.00
test_1          3   F   1/4     mycomputer  ciclón  Failed    62 0.00
```

As you can see before, the realization `test_1` has finished with an exit code 62 (See [errors](#) table). The exit code 62 indicates that `ungrib` failed. In order to solve this error, we are going to check out the log of the chunk number 1.

```
[user@mycomputer~]$ cat $HOME/WRF4G/etc/resources.wrf4g | grep "WRF4G_BASEPATH="
WRF4G_BASEPATH="/home/user/WRF4G/repository/output"

[user@mycomputer~]$ cd $HOME/WRF4G/repository/output/test_1/test_1/log/

[user@mycomputer~]$ ls
log_1_4.tar.gz
```

The chunk log name is composed of using chunk number and job identifier.

```
log_{chunk_number}_{job_identifier}.tar.gz
```

```
[user@mycomputer~]$ tar xzvf log_1_4.tar.gz
WRF4G.log
configure.wps
ls.wps
ls.wrf
ungrib_GFS_2011082800.out
```

In each log package you are able to see all WRF log binaries as well as WRF4G logs such as `WRF4G.log`, `wrfgel.out` and `monitor.log`. In our case, we are going to focus on `WRF4G.log` which is the main log.

```
[user@mycomputer~]$ cat WRF4G.log
* Mon Oct  1 17:27:45 CEST 2012: Creating WRF4G structure ...
~/home/user/WRF4G/repository/apps/WRFbin-3.1.1_r832INTEL_OMPI.tar.gz' -> ~/home/user/.gw_user_3/WRFbin-3.1.1_r832INTEL_OMPI
~/home/user/WRF4G/repository/output/test_1/test_1/namelist.input' -> ~/home/user/.gw_user_3/WRFV3/run/namelist.input'
* Mon Oct  1 17:27:46 CEST 2012: Preparing WRF4G binaries ...
* Mon Oct  1 17:27:46 CEST 2012: Creating parallel environment ...
* Mon Oct  1 17:27:46 CEST 2012: Using default configuration ...
* Mon Oct  1 17:27:46 CEST 2012: Checking restart information ...
* Mon Oct  1 17:27:46 CEST 2012: The boundaries and initial conditions are not available ...
* Mon Oct  1 17:27:46 CEST 2012: Downloading geo_em files and namelist.wps ...
/home/user/.gw_user_3/WRF4GEL/vcp -v /home/user/WRF4G/repository/domains/Santander_50km/* .
cp -v -R /home/user/WRF4G/repository/domains/Santander_50km/* /home/user/.gw_user_3/WPS
~/home/user/WRF4G/repository/domains/Santander_50km/geo_em.d01.nc' -> ~/home/user/.gw_user_3/WPS/geo_em.d01.nc'
~/home/user/WRF4G/repository/domains/Santander_50km/namelist.wps' -> ~/home/user/.gw_user_3/WPS/namelist.wps'
* Mon Oct  1 17:27:46 CEST 2012: Modifying namelist ...
Updating parameter start_date in file: namelist.wps
Updating parameter end_date in file: namelist.wps
Updating parameter max_dom in file: namelist.wps
Updating parameter prefix in file: namelist.wps
Updating parameter interval_seconds in file: namelist.wps
* Mon Oct  1 17:27:46 CEST 2012: About to run preprocessor and Ungrib ...
* Mon Oct  1 17:27:46 CEST 2012: Running preprocessor.default ...
Linking global data from: /home/user/WRF4G/repository/input/NCEP/GFS
~/home/user/.gw_user_3/WPS/grbData/gfs2011082812_00.grb' -> ~/home/user/WRF4G/repository/input/NCEP/GFS/2011/gfs2011082812
~/home/user/.gw_user_3/WPS/grbData/gfs2011082812_06.grb' -> ~/home/user/WRF4G/repository/input/NCEP/GFS/2011/gfs2011082812
~/home/user/.gw_user_3/WPS/grbData/gfs2011082812_12.grb' -> ~/home/user/WRF4G/repository/input/NCEP/GFS/2011/gfs2011082812
~/home/user/.gw_user_3/WPS/grbData/gfs2011082812_18.grb' -> ~/home/user/WRF4G/repository/input/NCEP/GFS/2011/gfs2011082812
~/home/user/.gw_user_3/WPS/grbData/gfs2011082812_24.grb' -> ~/home/user/WRF4G/repository/input/NCEP/GFS/2011/gfs2011082812
~/home/user/.gw_user_3/WPS/grbData/gfs2011082812_30.grb' -> ~/home/user/WRF4G/repository/input/NCEP/GFS/2011/gfs2011082812
~/home/user/.gw_user_3/WPS/grbData/gfs2011082812_36.grb' -> ~/home/user/WRF4G/repository/input/NCEP/GFS/2011/gfs2011082812
* Mon Oct  1 17:27:47 CEST 2012: Running ungrib ...
*****
WRF4G was deployed in ...
  /home/user/.gw_user_3
and it ran in ...
```

```
/home/user/.gw_user_3
*****
```

WRF4G.log shows that ungrib was the last WRF binary executed. Therefore, if you check out ungrib log, you will probably discover the error.

```
[user@mycomputer~]$ tail ungrib_GFS_2011082800.out
200.0 X      X      X      X      X
150.0 X      X      X      X      X
100.0 X      X      X      X      X
 70.0 X      X      X      X      X
 50.0 X      X      X      X      X

Subroutine DATINT: Interpolating 3-d files to fill in any missing data...
Looking for data at time 2011-08-28_00
ERROR: Data not found: 2011-08-28_00:00:00.0000
Begin rrpr
-----
```

The problem is that there is not input data to simulate the first chunk.

```
[user@mycomputer~]$ cat experiment.wrf4g | grep "extdata_path"
extdata_path = "${WRF4G_INPUT}/NCEP/GFS"

[user@mycomputer~]$ cat $HOME/WRF4G/etc/resources.wrf4g | grep "WRF4G_INPUT="
WRF4G_INPUT="/home/user/WRF4G/repository/input"

[user@mycomputer~]$ ls -l $HOME/WRF4G/repository/input/NCEP/GFS/2011/
total 36388
-rw-r--r-- 1 user user 4909850 2012-09-13 11:38 gfs2011082812_00.grb
-rw-r--r-- 1 user user 5411705 2012-09-13 11:38 gfs2011082812_06.grb
-rw-r--r-- 1 user user 5411214 2012-09-13 11:38 gfs2011082812_12.grb
-rw-r--r-- 1 user user 5415031 2012-09-13 11:38 gfs2011082812_18.grb
-rw-r--r-- 1 user user 5397677 2012-09-13 11:38 gfs2011082812_24.grb
-rw-r--r-- 1 user user 5386190 2012-09-13 11:38 gfs2011082812_30.grb
-rw-r--r-- 1 user user 5316014 2012-09-13 11:38 gfs2011082812_36.grb
```

Create a new experiment called `test_2`, based on `test`, and add this line `timestep_dxfactor="manual:10000"` to `experiment.wrf4g` file. After that, submit the new experiment

How to add new computing resources to WRF4G

WRF4G uses [DRM4G](#) to access to different Distributed Resource Managements (DRM) such as:

- PBS/Torque
- SGE
- FORK
- LoadLeveler
- SLURM

In order to add new resources, you need to edit `ComputingResources` section in the [framework4g.conf](#) file which is located under `$WRF4G_LOCATION/etc` directory. The file has to contain one resource per line with the format:

```
resource_name = attributes
...
resource_name = attributes
```

where:

- **resource_name**: It is the name of the computing resource.

attributes: They are the static attributes of the computing resource. The syntax is:

```
<scheme>://<username>@<host:port>?<query>
```

scheme: URL schemes available are "ssh" and "local".

- **ssh**: In order to connect to remote resource via SSH
- **local**: In order to use a local resource
- **username**: user name on the resource
- **host**: host name
- **port**: host port to connect to. By DEFAULT is 22

query: contains additional information of computing resources. The query string syntax is:

- key1=value1;key2=value2;key3=value3

The keys available are:

- **LRMS_TYPE**(mandatory): Type of LRMS system [pbs | sge | fork | slurm]
- **NODECOUNT**(mandatory): Maximum number of job slots for a resource.
- **QUEUE_NAME**(mandatory): Queue name to configure. It is mandatory except for "LRMS_TYPE=fork"
- **SSH_KEY_FILE**(optional): It defines the key file for "ssh" connection. By DEFAULT is ~/.ssh/id_rsa.
- **PROJECT**(optional): It specifies the project which the jobs are assigned. It is only optional for SGE and PBS.
- **PARALLEL_TAG**(optional): It defines the parallel environments available for SGE.
- **TEMP_DIR**(optional): Temporary directory on the resource to store data. By DEFAULT is \$HOME. TEMP_DIR path must be absolute.
- **RUN_DIR**(optional): Temporary directory used to run WRF Model on the resource. By DEFAULT is \$HOME. RUN_DIR path must be absolute.

Examples of configuration:

```
mycomputer = local://localhost?LRMS_TYPE=fork;NODECOUNT=2
```

```
PBS_cluster = local://localhost?LRMS_TYPE=pbs;QUEUE_NAME=estadistica;NODECOUNT=10
```

```
SGE_cluster = local://localhost?LRMS_TYPE=sge;PROJECT=l.project;NODECOUNT=40
```

If you want to configure a remote computing resource through ssh protocol, you need to put your private keys into your [?ssh-agent](#), and it will handle your authentication thereafter (see [Appendix A](#)) or set up SSH login without *password* (see [Appendix B](#)).

In addition, you will probably need to update WRF4G_BASEPATH, WRF4G_DOMAINPATH, WRF4G_INPUT and WRF4G_APPS variables, which are defined in [resources.wrf4g](#). Due to the fact that these variables may point to other machines. See [running environment](#) for more information.

```
remote_PBS = ssh://user@meteol.macc.unican.es?LRMS_TYPE=pbs;QUEUE_NAME=short;NODECOUNT=10
```

After modifying `ComputingResources` section, WRF4G takes few seconds in order to update the changes

Now, delete `mycomputer` resource and try to configure WRF4G on another resource

How to add a new WRF geographical domain

As stated before, a step of the WRF Preprocessor System (WPS), called `geogrid`, is not included in the WRF4G workflow. Thus, the user must deal with it by hand, or using another tool, such as [?WRF portal](#). `Geogrid` does the task of extracting the fixed fields (orography, land use data, etc.) that WRF needs to run at a given resolution and region. In the WRF4G framework, the output of `geogrid` is known as the "domain" of an experiment. Inside the original WRF4G tarball there are two example domains, `Santander_50km` and `wrfuc`, located in `$WRF4G_LOCATION/repository/domains`. Here we are going to see how a new one can be added. It's important to note that there exists an [?excellent on-line tutorial](#) for running WPS and WRF by hand. Users of WRF4G are encouraged to work through this tutorial before start running with WRF4G itself. This framework is intended to make life much easier for WRF users, but knowledge about WRF itself is needed to deal with common errors and issues, and to carry on a correct interpretation of the results.

First of all, WPS binaries are needed. If you don't have them in your system, you will need to build them from source. Instructions for doing this are available too in the [?WRF-ARW online tutorial](#). Once you have the binaries, you will need to prepare a `namelist.wps` file defining your requirements for the domain (size, location, resolution, etc.) Also, the `GEOGRID.TBL` file is needed to be present into a folder called `geogrid`, in the same place where `geogrid` is going to be executed. The final structure is the following:

```
[user@mycomputer~]$ tree
.
|-- geogrid
|   |-- GEOGRID.TBL
|-- namelist.wps
```

Provided the namelist and the GEOGRID.TBL, the next step is to execute geogrid.exe. This will generate a netCDF file called geo_em_d01.nc, which contains all the fixed fields that WRF needs to run with that domain.

```
[user@mycomputer~]$ ${PATH_TO_GEOGRID}/geogrid.exe >& geogrid.log
[user@mycomputer~]$ ~/ls
geo_em.d01.nc  geogrid  namelist.wps  geogrid.log
```

The next and final step is to copy the geo_em and the namelist.wps files to a directory with the name chosen for the domain (e.g. Europe_15k). This folder must be located in the path specified by the variable WRF4G_DOMAINPATH in [resources.wrf4g](#). After following these steps, the new domain is available for using it in any WRF4G experiment, just setting the variable domain equal to the name of the directory in [experiments.wrf4g](#) (the folder where we located geo_em_d01.nc and namelist.wps).

How to reconfigure the features of an experiment

If you want to reconfigure the features of an experiment and you execute wrf4g_prepare, you will probably see:

```
[user@mycomputer~]$ wrf4g_prepare
Warning: You are using resources.wrf4g located in the $HOME/WRF4G/etc/ directory.
Experiment already exists
```

In order to do it, you need to execute wrf4g_prepare --reconfigure. In the example below, we are going to extend the end_date of one experiment.

```
[user@mycomputer~]$ cat experiment.wrf4g | grep "end_date=\""
end_date="2011-08-30_00:00:00"

[user@mycomputer~]$ cat experiment.wrf4g | grep "end_date=\""
end_date="2011-09-01_00:00:00"

[user@mycomputer~]$ wrf4g_prepare --reconfigure
Warning: You are using resources.wrf4g located in the /home/carlos/WRF4G/etc/ directory.
Preparing namelist...
WRFV3/run/namelist.input
WRF Check Warning: CAM radiation selected but paerlev/levsiz/cam_abs_dim1/cam_abs_dim2 was not set. Fixing...
WRF Check Warning: radt is shorter than dx (0.500000)

---> Single params run
---> Continuous run
    ---> cycle_chunks: test 2011-08-28_12:00:00 2011-09-01_00:00:00
        ---> chunks 1: test 2011-08-28_12:00:00 2011-08-29_00:00:00
        ---> chunks 2: test 2011-08-29_00:00:00 2011-08-29_12:00:00
        ---> chunks 3: test 2011-08-29_12:00:00 2011-08-30_00:00:00
        ---> chunks 4: test 2011-08-30_00:00:00 2011-08-30_12:00:00
        ---> chunks 5: test 2011-08-30_12:00:00 2011-08-31_00:00:00
        ---> chunks 6: test 2011-08-31_00:00:00 2011-08-31_12:00:00
        ---> chunks 7: test 2011-08-31_12:00:00 2011-09-01_00:00:00
```

How to resubmit an experiment

If one experiment has finished with an error and you want to rerun the experiment again. You have to execute:

```
[user@mycomputer~]$ wrf4g_submit --rerun -f -e test1
Submitting realization: "test1"
  Submitting Chunk 1:      2011-08-28_12:00:00      2011-08-29_00:00:00
  Submitting Chunk 2:      2011-08-29_00:00:00      2011-08-29_12:00:00
  Submitting Chunk 3:      2011-08-29_12:00:00      2011-08-30_00:00:00
```

How to rerun a specific chunk of a realization

Imagine, you want to resubmit the chunk number **1** of the realization **test2**. In this case, it is highly recommended that you use the option `--dry-run` of `wrf4g_submit` command before you submit your chunk in order to make sure you are submitting that chunk an nothing else.

```
[user@mycomputer~]$ wrf4g_submit --dry-run --rerun -c 1 -f -r test2
Submitting realization: "test2"
  Submitting Chunk 1:      2011-08-28_12:00:00      2011-08-29_00:00:00

[user@mycomputer~]$ wrf4g_submit --rerun -c 1 -f -r test2
Submitting realization: "test2"
  Submitting Chunk 1:      2011-08-28_12:00:00      2011-08-29_00:00:00
```

How to use wrf4g_kill command

In this example, we are going to simulate an experiment with independent realizations which has [multiple_parameters flag](#) activated. The experiment is composed of five realizations with three chunk per realization. In order to use [wrf4g_kill command](#), we are going to first submit the `wrfuc_physics` experiment.

```
[user@mycomputer~]$ cd $HOME/WRF4G/experiments/wrfuc_physics

[user@mycomputer~]$ ls
experiment.wrf4g

[user@mycomputer~]$ cat experiment.wrf4g | grep "param"
multiple_parameters=1
multiparams_variables="mp_physics,cu_physics,ra_lw_physics,ra_sw_physics,sf_sfclay_physics,bl_pbl_physics,sf_surface_physics"
multiparams_nitems="${max_dom},${max_dom},${max_dom},${max_dom},${max_dom},${max_dom},${max_dom}"
multiparams_combinations="5,1:1:0,1,1,2,2,2/4,1:1:0,1,1,1,1,2/4,1:1:0,1,1,2,2,2/4,1:1:0,1,1,7,7,2/4,3:3:0,1,1,7,7,2"
multiparams_labels="phys1/phys2/phys3/phys4/phys5"

[user@mycomputer~]$ wrf4g_prepare
Warning: You are using resources.wrf4g located in the /home/carlos/WRF4G/etc/ directory.
Preparing namelist...
WRFV3/run/namelist.input
WRF Check Warning: CAM radiation selected but paerlev/levsiz/cam_abs_dim1/cam_abs_dim2 was not set. Fixing...
WRF Check Warning: radt is shorter than dx (0.500000)

--->Realization: multiparams=phys1 2011-08-28_12:00:00 2011-08-30_00:00:00
Updating parameter mp_physics in file: namelist.input
Updating parameter cu_physics in file: namelist.input
Updating parameter ra_lw_physics in file: namelist.input
Updating parameter ra_sw_physics in file: namelist.input
Updating parameter sf_sfclay_physics in file: namelist.input
Updating parameter bl_pbl_physics in file: namelist.input
Updating parameter sf_surface_physics in file: namelist.input
---> Continuous run
  ---> cycle_chunks: uc_phys__phys1 2011-08-28_12:00:00 2011-08-30_00:00:00
    ---> chunks 1: uc_phys__phys1 2011-08-28_12:00:00 2011-08-29_00:00:00
    ---> chunks 2: uc_phys__phys1 2011-08-29_00:00:00 2011-08-29_12:00:00
    ---> chunks 3: uc_phys__phys1 2011-08-29_12:00:00 2011-08-30_00:00:00

--->Realization: multiparams=phys2 2011-08-28_12:00:00 2011-08-30_00:00:00
Updating parameter mp_physics in file: namelist.input
```

```

Updating parameter cu_physics in file: namelist.input
Updating parameter ra_lw_physics in file: namelist.input
Updating parameter ra_sw_physics in file: namelist.input
Updating parameter sf_sfclay_physics in file: namelist.input
Updating parameter bl_pbl_physics in file: namelist.input
Updating parameter sf_surface_physics in file: namelist.input
---> Continuous run
    ---> cycle_chunks: uc_phys__phys2 2011-08-28_12:00:00 2011-08-30_00:00:00
        ---> chunks 1: uc_phys__phys2 2011-08-28_12:00:00 2011-08-29_00:00:00
        ---> chunks 2: uc_phys__phys2 2011-08-29_00:00:00 2011-08-29_12:00:00
        ---> chunks 3: uc_phys__phys2 2011-08-29_12:00:00 2011-08-30_00:00:00

--->Realization: multiparams=phys3 2011-08-28_12:00:00 2011-08-30_00:00:00
Updating parameter mp_physics in file: namelist.input
Updating parameter cu_physics in file: namelist.input
Updating parameter ra_lw_physics in file: namelist.input
Updating parameter ra_sw_physics in file: namelist.input
Updating parameter sf_sfclay_physics in file: namelist.input
Updating parameter bl_pbl_physics in file: namelist.input
Updating parameter sf_surface_physics in file: namelist.input
---> Continuous run
    ---> cycle_chunks: uc_phys__phys3 2011-08-28_12:00:00 2011-08-30_00:00:00
        ---> chunks 1: uc_phys__phys3 2011-08-28_12:00:00 2011-08-29_00:00:00
        ---> chunks 2: uc_phys__phys3 2011-08-29_00:00:00 2011-08-29_12:00:00
        ---> chunks 3: uc_phys__phys3 2011-08-29_12:00:00 2011-08-30_00:00:00

--->Realization: multiparams=phys4 2011-08-28_12:00:00 2011-08-30_00:00:00
Updating parameter mp_physics in file: namelist.input
Updating parameter cu_physics in file: namelist.input
Updating parameter ra_lw_physics in file: namelist.input
Updating parameter ra_sw_physics in file: namelist.input
Updating parameter sf_sfclay_physics in file: namelist.input
Updating parameter bl_pbl_physics in file: namelist.input
Updating parameter sf_surface_physics in file: namelist.input
---> Continuous run
    ---> cycle_chunks: uc_phys__phys4 2011-08-28_12:00:00 2011-08-30_00:00:00
        ---> chunks 1: uc_phys__phys4 2011-08-28_12:00:00 2011-08-29_00:00:00
        ---> chunks 2: uc_phys__phys4 2011-08-29_00:00:00 2011-08-29_12:00:00
        ---> chunks 3: uc_phys__phys4 2011-08-29_12:00:00 2011-08-30_00:00:00

--->Realization: multiparams=phys5 2011-08-28_12:00:00 2011-08-30_00:00:00
Updating parameter mp_physics in file: namelist.input
Updating parameter cu_physics in file: namelist.input
Updating parameter ra_lw_physics in file: namelist.input
Updating parameter ra_sw_physics in file: namelist.input
Updating parameter sf_sfclay_physics in file: namelist.input
Updating parameter bl_pbl_physics in file: namelist.input
Updating parameter sf_surface_physics in file: namelist.input
---> Continuous run
    ---> cycle_chunks: uc_phys__phys5 2011-08-28_12:00:00 2011-08-30_00:00:00
        ---> chunks 1: uc_phys__phys5 2011-08-28_12:00:00 2011-08-29_00:00:00
        ---> chunks 2: uc_phys__phys5 2011-08-29_00:00:00 2011-08-29_12:00:00
        ---> chunks 3: uc_phys__phys5 2011-08-29_12:00:00 2011-08-30_00:00:00

[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks Comp.Res  WN      Run.Sta      ext  %
uc_phys__phys1  -   P  0/3   -         -       Prepared     -  0.00
uc_phys__phys2  -   P  0/3   -         -       Prepared     -  0.00
uc_phys__phys3  -   P  0/3   -         -       Prepared     -  0.00
uc_phys__phys4  -   P  0/3   -         -       Prepared     -  0.00
uc_phys__phys5  -   P  0/3   -         -       Prepared     -  0.00

```



```
[user@mycomputer~]$ wrf4g_submit
Submitting realization: "uc_phys__phys1"
  Submitting Chunk 1:    2011-08-28_12:00:00    2011-08-29_00:00:00
  Submitting Chunk 2:    2011-08-29_00:00:00    2011-08-29_12:00:00
  Submitting Chunk 3:    2011-08-29_12:00:00    2011-08-30_00:00:00
Submitting realization: "uc_phys__phys2"
  Submitting Chunk 1:    2011-08-28_12:00:00    2011-08-29_00:00:00
  Submitting Chunk 2:    2011-08-29_00:00:00    2011-08-29_12:00:00
  Submitting Chunk 3:    2011-08-29_12:00:00    2011-08-30_00:00:00
Submitting realization: "uc_phys__phys3"
  Submitting Chunk 1:    2011-08-28_12:00:00    2011-08-29_00:00:00
  Submitting Chunk 2:    2011-08-29_00:00:00    2011-08-29_12:00:00
  Submitting Chunk 3:    2011-08-29_12:00:00    2011-08-30_00:00:00
Submitting realization: "uc_phys__phys4"
  Submitting Chunk 1:    2011-08-28_12:00:00    2011-08-29_00:00:00
  Submitting Chunk 2:    2011-08-29_00:00:00    2011-08-29_12:00:00
  Submitting Chunk 3:    2011-08-29_12:00:00    2011-08-30_00:00:00
Submitting realization: "uc_phys__phys5"
  Submitting Chunk 1:    2011-08-28_12:00:00    2011-08-29_00:00:00
  Submitting Chunk 2:    2011-08-29_00:00:00    2011-08-29_12:00:00
  Submitting Chunk 3:    2011-08-29_12:00:00    2011-08-30_00:00:00
```

```
[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks Comp.Res  WN      Run.Sta      ext  %
uc_phys__phys1  0   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys2  3   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys3  6   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys4  9   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys5 12   W  1/3   -          -       Submitted    -  0.00
```

```
[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks Comp.Res  WN      Run.Sta      ext  %
uc_phys__phys1  0   R  1/3   mycomputer ciclogenes WRF         -  0.00
uc_phys__phys2  3   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys3  6   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys4  9   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys5 12   W  1/3   -          -       Submitted    -  0.00
```

As the experiment is working now, we are going to stop the chunks of the `uc_phys__phys1` realization using `wrf4g_kill` command.

```
[user@mycomputer~]$ wrf4g_kill -r uc_phys__phys1

[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks Comp.Res  WN      Run.Sta      ext  %
uc_phys__phys1  -   P  0/3   -          -       Prepared     -  0.00
uc_phys__phys2  3   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys3  6   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys4  9   W  1/3   -          -       Submitted    -  0.00
uc_phys__phys5 12   W  1/3   -          -       Submitted    -  0.00
```

Note that, `Run.Sta` has changed into `Prepared` value. If you want you submit again the realization, you only need to execute `wrf4g_submit -r uc_phys__phys1`.

```
[user@mycomputer~]$ wrf4g_submit -r uc_phys__phys1
Submitting realization: "uc_phys__phys1"
  Submitting Chunk 1:    2011-08-28_12:00:00    2011-08-29_00:00:00
  Submitting Chunk 2:    2011-08-29_00:00:00    2011-08-29_12:00:00
  Submitting Chunk 3:    2011-08-29_12:00:00    2011-08-30_00:00:00
```

```

[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks Comp.Res  WN      Run.Sta      ext  %
uc_phys__phys1  15  W  1/3  -        -        Submitted    - 0.00
uc_phys__phys2   3  R  1/3  mycomputer ciclogenes real      - 0.00
uc_phys__phys3   6  W  1/3  -        -        Submitted    - 0.00
uc_phys__phys4   9  W  1/3  -        -        Submitted    - 0.00
uc_phys__phys5  12  W  1/3  -        -        Submitted    - 0.00

[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks Comp.Res  WN      Run.Sta      ext  %
uc_phys__phys1  15  W  1/3  -        -        Submitted    - 0.00
uc_phys__phys2   3  R  1/3  mycomputer ciclogenes WRF      - 0.00
uc_phys__phys3   6  W  1/3  -        -        Submitted    - 0.00
uc_phys__phys4   9  W  1/3  -        -        Submitted    - 0.00
uc_phys__phys5  12  W  1/3  -        -        Submitted    - 0.00

```

The `uc_phys__phys2` is running because all realizations are independents.

Now, try to stop the `uc_phys` experiment and then resubmit it again