

Table of Contents

Run a simple experiment on the machine where WRF4G is installed	2
Start wrf4g_framework and list computing resources	2
Prepare and submit the test experiment	2
Check the experiment output and log	3

Run a simple experiment on the machine where WRF4G is installed

Before starting this tutorial make sure you have correctly installed WRF4G on your machine following the [Installation instructions](#)

Start wrf4g_framework and list computing resources

Simply start the [WRF4G Framework](#):

```
[user@mycomputer~]$ wrf4g_framework start
```

Doing this, we will start the services that manage the computing resources and keep track of the experiments.

Note that if you do not start wrf4g_framework you won't be able to work with WRF4G.

WRF4G framework is configured by default to use a single computing resource (the computer where WRF4G has been installed). To list the computing resources where you can submit experiment (in this case only localhost) run the following:

```
[user@mycomputer~]$ wrf4g_resources
HID PRIO OS          ARCH  MHZ  %CPU  MEM(F/T)  DISK(F/T)  N(U/F/T) LRMS          HOSTNAME
0   1   GNU/Linux2.6.32 x86_64 2000   0   439/3270  1143/8662  0/1/1 FORK          mycomputer
```

Prepare and submit the test experiment

Go to the directory where the experiment configuration files are located:

```
[user@mycomputer~]$ cd $WRF4G_LOCATION/experiments/single_test
```

Edit [experiment.wrf4g](#) and check its configuration.

Run [wrf4g_prepare](#) to prepare the experiment:

```
[user@mycomputer~]$ wrf4g_prepare
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 2011-08-30_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
```

Check the experiment status with [wrf4g_status](#). You will see the experiment is in P (Prepared status).

```
[user@mycomputer~]$ wrf4g_status
Experiment P   W   R   D   F
test       1   0   0   0   0

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

Submit the experiment with [wrf4g_submit](#) and check the status during one minute. You will see the status of the experiment: submitted, downloading input data, ungrib, metgrid, real and wrf.

```

[user@mycomputer~]$ wrf4g_submit
Submitting realization: "test"
    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
Experiment P   W   R   D   F
test       0   1   0   0   0

[user@mycomputer~]$ wrf4g_status --long
Realization    GW  Stat Chunks Comp.Res  WN      Run.Sta    ext  %
test          0   W  1/3    -        -      Submitted  -  0.00

[user@mycomputer~]$ wrf4g_status --long
Realization    GW  Stat Chunks Comp.Res  WN      Run.Sta    ext  %
test          0   R  1/3    mycomputer mycomputer real    -  0.00

[user@mycomputer~]$ wrf4g_status --long
Realization    GW  Stat Chunks Comp.Res  WN      Run.Sta    ext  %
test          0   R  1/3    mycomputer mycomputer WRF      -  0.00

[user@mycomputer~]$ wrf4g_status --long
Realization    GW  Stat Chunks Comp.Res  WN      Run.Sta    ext  %
test          0   R  2/3    mycomputer mycomputer Submitted - 33.33

[user@mycomputer~]$ wrf4g_status --long
Realization    GW  Stat Chunks Comp.Res  WN      Run.Sta    ext  %
test          0   R  3/3    mycomputer mycomputer Finished  0 100.00

```

The experiment had to finish with **ext** option equals **0** code which means it has finalized OK. If **ext** has a different code

Check the experiment output and log

The location where output and log files are stored is defined with the WRF4G_BASEPATH variable in [resources.wrf4g](#). Check \$WRF4G_LOCATION/etc/resources.wrf4g to discover where WRF4G_BASEPATH is pointing. You will see that it is pointing to WRF4G_LOCATION/repository/output. Go to WRF4G_LOCATION/repository/output/test/test and see the folders structure and the output files.

```

[user@mycomputer~]$ ls -lh /tmp/WRF4G/repository/output/test/test/output
total 420K
-rw-rw-r-- 1 user user 6.2K 2011-10-28 12:40 wrf24hc_d01_20110828T120000Z_20110828T120000Z.nc
-rw-rw-r-- 1 user user 6.2K 2011-10-28 12:41 wrf24hc_d01_20110829T000000Z_20110829T000000Z.nc
-rw-rw-r-- 1 user user 6.2K 2011-10-28 12:43 wrf24hc_d01_20110829T120000Z_20110829T120000Z.nc
-rw-rw-r-- 1 user user 14K 2011-10-28 12:40 wrfout_d01_20110828T120000Z_20110828T180000Z.nc
-rw-rw-r-- 1 user user 11K 2011-10-28 12:40 wrfout_d01_20110828T210000Z_20110829T000000Z.nc
-rw-rw-r-- 1 user user 14K 2011-10-28 12:41 wrfout_d01_20110829T000000Z_20110829T060000Z.nc
-rw-rw-r-- 1 user user 11K 2011-10-28 12:41 wrfout_d01_20110829T090000Z_20110829T120000Z.nc
-rw-rw-r-- 1 user user 14K 2011-10-28 12:43 wrfout_d01_20110829T120000Z_20110829T180000Z.nc
-rw-rw-r-- 1 user user 11K 2011-10-28 12:43 wrfout_d01_20110829T210000Z_20110830T000000Z.nc
-rw-rw-r-- 1 user user 3.4K 2011-10-28 12:40 wrfrain_d01_20110828T190000Z_20110828T190000Z.nc
-rw-rw-r-- 1 user user 3.4K 2011-10-28 12:41 wrfrain_d01_20110829T070000Z_20110829T070000Z.nc
-rw-rw-r-- 1 user user 3.4K 2011-10-28 12:43 wrfrain_d01_20110829T190000Z_20110829T190000Z.nc
-rw-rw-r-- 1 user user 99K 2011-10-28 12:40 wrfxtrm_d01_20110828T120000Z_20110829T000000Z.nc
-rw-rw-r-- 1 user user 99K 2011-10-28 12:41 wrfxtrm_d01_20110829T000000Z_20110829T120000Z.nc
-rw-rw-r-- 1 user user 99K 2011-10-28 12:43 wrfxtrm_d01_20110829T120000Z_20110830T000000Z.nc

[user@mycomputer~]$ tree /tmp/WRF4G/repository/output/test/test
.
??? log

```

```
?   ??? log_1_1.tar.gz
?   ??? log_2_2.tar.gz
?   ??? log_3_3.tar.gz
??? namelist.input
??? output
?   ??? wrfout_d01_19830825T120000Z_19830825T233000Z.nc
?   ??? wrfout_d01_19830826T000000Z_19830826T000000Z.nc
?   ??? wrfout_d01_19830826T000000Z_19830826T113000Z.nc
?   ??? wrfout_d01_19830826T120000Z_19830826T120000Z.nc
?   ??? wrfout_d01_19830826T120000Z_19830826T233000Z.nc
?   ??? wrfout_d01_19830827T000000Z_19830827T000000Z.nc
?   ??? wrfrain_d01_19830826T000000Z_19830826T000000Z.nc
?   ??? wrfrain_d01_19830826T120000Z_19830826T120000Z.nc
?   ??? wrfrain_d01_19830827T000000Z_19830827T000000Z.nc
?   ??? wrfxtrm_d01_19830825T120000Z_19830825T233000Z.nc
?   ??? wrfxtrm_d01_19830826T000000Z_19830826T000000Z.nc
?   ??? wrfxtrm_d01_19830826T000000Z_19830826T113000Z.nc
?   ??? wrfxtrm_d01_19830826T120000Z_19830826T120000Z.nc
?   ??? wrfxtrm_d01_19830826T120000Z_19830826T233000Z.nc
?   ??? wrfxtrm_d01_19830827T000000Z_19830827T000000Z.nc
??? realout
??? restart
    ??? wrfrst_d01_19830826T000000Z.nc
    ??? wrfrst_d01_19830826T120000Z.nc
    ??? wrfrst_d01_19830827T000000Z.nc
```