

Table of Contents

Command Line Interface (CLI) for WRF4G	2
wrf4g_framework	2
Usage	2
Synopsis	2
wrf4g_prepare	2
Usage	2
Synopsis	2
Options	2
wrf4g_submit	2
Usage	2
Synopsis	2
Options	2
wrf4g_status	3
Usage	3
Synopsis	3
Options	3
Output field description	3
wrf4g_resources	4
Usage	4
Synopsis	4
Options	4
Output field description	4
wrf4g_priority	5
Usage	5
Synopsis	5
Options	5
wrf4g_kill	5
Usage	5
Synopsis	5
Options	5

Command Line Interface (CLI) for WRF4G

wrf4g_framework

Usage

wrf4g_framework {start|stop|restart|reload|status}

Synopsis

wrf4g_framework manages WRF4G framework components: GridWay and MySQL (in case it is needed). It loads the framework configuration from `$WRF4G_LOCATION/etc/framework4g.conf`.

wrf4g_prepare

Usage

wrf4g_prepare [--dry-run] [--reconfigure] [--verbose] [--help]

Synopsis

Given a file (experiment.wrf4g) describing the experiment, prepare it by creating the realization and chunks needed to perform the experiment.

Options

<code>--dry-run</code>	Perform a trial run with no changes made.
<code>--reconfigure</code>	Reconfigure experiment. With this option we can change the start and end date of the experiments and add new physics. Values are taken from a modified <code>experiment.wrf4g</code> .
<code>--verbose</code>	Verbose mode. Explain what is being done
<code>--help</code>	Shows this help

wrf4g_submit

Usage

wrf4g_submit [--dry-run] [--{exp Experiment|rea Realization|frea File|chunk Chunk|nchunk Number_of_chunks|nrea Number_of_realizations}] [--rerun] [--run-just-one][--priority P] [--verbose] [--force] [--tdep afterok | afternotok | afterany] [--help]

Synopsis

Submit an experiment or a realization. If the user doesn't specify an experiment or a realization with the options, the name of the experiment will be retrieved from `experiment.wrf4g` in case it exists in the current folder.

Options

<code>--version</code>	show program's version number and exit
<code>-h, --help</code>	show this help message and exit
<code>-n, --dry-run</code>	Perform a trial run with no changes made
<code>-e name, --exp=name</code>	Name of the experiment to submit
<code>-r name, --rea=name</code>	Name of the realization to submit
<code>-F FILE, --frea=FILE</code>	File containing the name of the realization to submit.
<code>-a, --rerun</code>	Force to run although the realization or experiment has finished
<code>-o, --run-just-one</code>	Run just the first chunk of the first realization. Only for testing purposes.
<code>-C N, --nchunk=N</code>	Run the next N chunks not finished of each realization
<code>-R N, --nrea=N</code>	Run the next N realizations not finished of the experiment

```

-p P, --priority=P    P is the priority of the experiment or realization that is
                    going to be launched (P is an integer between 0 and 20)
-v, --verbose        Verbose mode. It explains what is being done
-f, --force          Don't ask the user if he wants to submit an experiment
                    already submitted
-d TYPE_DEP, --tdep=TYPE_DEP
                    Specify dependencies between Chunks. afterok: The
                    chunk may be scheduled for execution only after jobs
                    jobid have finished without any error. afternotok: The
                    chunk may be scheduled for execution only after jobs
                    jobid have finished with errors. afterany: The chunk
                    may be scheduled for execution after jobs jobid have
                    finished, with or without errors.

```

wrf4g_status

Usage

```
wrf4g_status [--{exp Experiment|rea Realization}] [--long] [--ncharacters N] [--help]
```

Synopsis

Print the experiment or realization status.

Options

```

--version            show program's version number and exit
-h, --help          show this help message and exit
-e name, --exp=name Name of the experiment.
-r name, --rea=name Name of the realization
-l, --long          Show a detailed status.
-n NUMBER_OF_CHARACTERS, --ncharacters=NUMBER_OF_CHARACTERS
                    Print n characters of the Experiment's name
                    (default value is 20 characters)

```

Output field description

Summarized output: Shows realizations' status of each experiment (Default).

```

[user@mycomputer~]$ wrf4g_status
Experiment P    W    R    D    F
test      0    0    0    1    0
uc_phys   0    4    1    0    0
uc_single 1    0    0    0    0

```

- P: Prepared
- W: Waiting
- R: Running
- F: Failed

Long output: Shows a detailed realization status of every experiment

```

[user@mycomputer~]$ wrf4g_status --long
Realization      GW  Stat Chunks Comp.Res  WN      Run.Sta      ext  %
test             2   D   3/3   mycomputer sipcl8   Finished    0 100.00
uc_phys__phys1   3   R   1/3   mycomputer sipcl8   WRF         - 0.00
uc_phys__phys2   6   W   1/3   -           -         Submitted   - 0.00

```

uc_phys__phys3	9	W	1/3	-	-	Submitted	- 0.00
uc_phys__phys4	12	W	1/3	-	-	Submitted	- 0.00
uc_phys__phys5	15	W	1/3	-	-	Submitted	- 0.00
uc_single	-	P	0/3	-	-	Prepared	- 0.00

- Realization: Realization name.
- Status: It can take the following values: P(Prepared), S(Submitted), R(Running), F(Failed) and D(Done).
- Chunks [Chunk currently running/Total Chunks]: A realization is split into chunks. Each chunk is sent as a job.
- Computer resource: Computing Resource where the job is running. (It has to be one of the resources listed by wrf4g_resources)
- WN: Computing node where the job is running.
- Run.Sta: Job status in the WN (Downloading data, running ungrib, real, wrf, ...)
- ext: Exit Code. If exit code is different from 0, there has been an error. Error codes are explained in `$WRF4G_LOCATION/lib/bash/wrf4g_exit_codes.sh`
- % : percentage of simulation finished.

wrf4g_resources

Usage

```
wrf4g_resources [-h] [-c delay] [-nfx] [-m job_id] [host_id]
```

Synopsis

Print information about all the resources configured in framework4g.conf (default)

Options

```
-h          print this help
-c delay    refresh host information every delay seconds
-n          do not print the header
-f          full format
-x          xml format
-m job_id   print hosts matching the requirements of a given job
host_id     only monitor this host_id, printing also queue information
```

Output field description

```
[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_6 2000   0   484/3270  1152/8662   0/1/1 FORK  mycomputer
1   1   GNU/Linux2.6.18 x86_6 2127   0   12/2048  671273/921472 88/0/88 PBS   asna_t1
2   1   GNU/Linux2.6.18 x86_6 2127   0   11/2048  671273/921472 168/0/168 PBS  asna_b2
```

FIELD INFORMATION

```
HID          host unique identification assigned by the GridWay system
PRIO         priority assigned to the host
OS           operating system
ARCH         architecture
MHZ          CPU speed in MHZ
%CPU         free CPU ratio
MEM(F/T)     system memory: F = Free, T = Total
DISK(F/T)    secondary storage: F = Free, T = Total
N(U/F/T)     number of slots: U = used by GridWay, F = free, T = total
LRMS         local resource management system, the jobmanager name
HOSTNAME     FQDN of this hostthe
```

QUEUE FIELD INFORMATION

```
QUEUENAME    name of this queue
```

SL(F/T)	slots: F = Free, T = Total
WALLT	queue wall time
CPUT	queue cpu time
COUNT	queue count number
MAXR	max. running jobs
MAXQ	max. queued jobs
STATUS	queue status
DISPATCH	queue dispatch type
PRIORITY	queue priority

wrf4g_priority

Usage

```
wrf4g_priority [--dry-run] [--(exp Experiment |rea Realization |frea File)] [--priority P] [--verbose] [--help]
```

Synopsis

Change the priority of an experiment/realization. The priority must be in range [0,20] and default value is 0. When a chunk gets a priority of 20, it becomes urgent. This chunk is dispatched as soon as possible, passing all the scheduling policies.

Options

--version	show program's version number and exit
-h, --help	show this help message and exit
-n, --dry-run	Perform a trial run with no changes made
-e name, --exp=name	Name of the experiment to submit
-r name, --rea=name	Name of the realization to submit
-F FILE, --frea=FILE	File containing the name of the realization to change the priority
-v, --verbose	Verbose mode. Explain what is being done
-p PRIORITY, --priority=PRIORITY	The priority must be in range [0,20].

wrf4g_kill

Usage

```
wrf4g_kill [--dry-run] [--(exp Experiment |rea Realization|frea File)] [--verbose] [--help]
```

Synopsis

Kill an experiment or realization. Only the chunks which haven't been done, will go back to "Prepared" status.

Options

--version	show program's version number and exit
-h, --help	show this help message and exit
-n, --dry-run	Perform a trial run with no changes made
-e name, --exp=name	Name of the experiment to submit
-r name, --rea=name	Name of the realization to submit
-F FILE, --frea=FILE	File containing the name of the realization to kill.
-v, --verbose	Verbose mode. Explain what is being done.