

## IDIOT'S GUIDE TO USING ROSE/CYLC TO RUN JULES ON JASMIN

### **Introduction**

The purpose of this dummy or idiot's guide is to bring together all of my notes on using Rose/Cylc to run suites (e.g. relatively simple stand-alone JULES suites). This is written from the perspective of creating suites using PUMA (at the University of Reading), then submitting them to JASMIN/Lotus. This is written by and aimed at the ultimate programming dummy, Charlie Williams. Very many thanks to Rosalyn Hatcher and the rest of NCAS-CMS at Reading, and of course Toby Marthews, Alberto Martinez de la Torre and Edward Comyn-Platt at CEH, for providing invaluable advice and actually doing all the hard work.

This is a specifically targeted amalgamation of several sources, namely:

<https://jules.jchmr.org>

<http://jules.jchmr.org/content/getting-started>

<http://jules.jchmr.org/content/scratch>

<http://jules.jchmr.org/content/convert-namelists>

<http://help.ceda.ac.uk/category/107-batch-computing-on-lotus>

### **Initial setup and login**

The following assumes that all usernames have been created and registered (e.g. access to PUMA, access to MOSRS (as well as appropriate groups e.g. JULES ), access to JASMIN (as well as appropriate group workspaces). If not, see <http://help.ceda.ac.uk/collection/59-jasmin-documentation>.

Once registered, the following also assumes that initial setup (e.g. generating ssh keys) has already been done. If not, need to follow setup guides. For JASMIN, see:

<http://help.ceda.ac.uk/article/189-get-started-with-jasmin>

Once the above is done, several things to check on PUMA:

- 1) Make sure log into Jasmin works okay. Assuming above setup has worked and keys are registered, `ssh -Y <userID>@jasmin-login1.ceda.ac.uk` should work without being prompted for password

- 2) Make sure `.ssh/config` file exists and contains JASMIN keys. If it doesn't exist, copy file from home Met Department directory and modify (see below)
- 3) Make it simpler to directly login to JASMIN science nodes - add code below to `.ssh/config` file on PUMA (inserting userID and appropriate path to keys):

```
Host jasmin
Hostname jasmin-login1.ceda.ac.uk
User <userID>
IdentityFile ~/.ssh/id_rsa_jasmin
ForwardAgent yes
ControlMaster auto
ControlPath /tmp/ssh-socket-%r@%h-%p
ControlPersist 600
ForwardX11 yes
ServerAliveInterval 30
```

```
Host jasmin-sci2.ceda.ac.uk
User <userID>
IdentityFile ~/.ssh/id_rsa_jasmin
ForwardAgent yes
ProxyCommand ssh -Y jasmin -W %h:%p
ControlMaster auto
ControlPath /tmp/ssh-socket-%r@%h-%p
ControlPersist 600
ForwardX11 yes
ServerAliveInterval 30
```

*Note: last 2 lines in each paragraph are important, as this prevents connection being broken mid-process.*

Should now be able to log straight onto science node `jasmin-sci2` using `ssh -Y jasmin-sci2.ceda.ac.uk`

*Note: although this works, if logging in from home this might allow x-windows to open. See below.*

Add code below to `.bash_profile` file in home directory on JASMIN:

```
# Get the aliases and functions
```

```
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
```

```
# User specific environment and startup programs
```

```
export PATH=$PATH:$HOME/bin
```

```
HOST=$(hostname)

if [[ $HOST = "jasmin-sci2.ceda.ac.uk" || $HOST = "jasmin-cylc.ceda.ac.uk" ||
$HOST = "jasmin-sci1.ceda.ac.uk" ]]; then
    # Rose/cylc on jasmin-sci & Lotus nodes
    export PATH=/apps/contrib/metomi/bin:$PATH
fi
```

If logging in from home, can access PUMA directly from nx: `ssh -Y`

`<userID>@puma.nerc.ac.uk`. Then need to access JASMIN login node: `ssh -Y`

`<userID>@jasmin-login1.ceda.ac.uk`. Then, finally, login to sci node: `ssh -Y jasmin-sci2.ceda.ac.uk`. In other words, if x-windows are needed, can't just login directly to sci node from PUMA.

*Note: when accessing PUMA from nx, need to use full PUMA name not just local host definition. Also, -Y flag is important and must be included in order to allow x-forwarding. Lastly, when accessing JASMIN login node, username must be included.*

### **If a password is ever requested**

This means the ssh-agent has died, so needs to be restarted. There are no passwords on JASMIN - only way to access is to use keys. Once these have been generated, there is almost never a need to delete them!

To restart the ssh-agent:

- 1) `ssh-add ~/.ssh/id_rsa_jasmin`
- 2) If 1) fails (with message like "*Could not connect to authentication agent*"), need to delete `~/.ssh/environment.puma` file, logout of PUMA and back in again to start up a new ssh-agent, then repeat 1)

If keys really need to be deleted (e.g. because they have been forgotten), need to regenerate them following instructions at above websites. For JASMIN, once the key pairs have been regenerated, need to paste the public key into profile, accessed by JASMIN accounts portal at <https://accounts.jasmin.ac.uk/>. It may take up to 20 minutes for this to register.

## **Setting up a Rose suite to run JULES on JASMIN**

Firstly, need to obtain a suite.

Example here: u-al609

This is Ros H's copy of Toby's JULES v4.8 suite, but using a branch which includes new inundation code. This code will be fully incorporated into v4.9, but is not yet available. So far, Ros H has modified this so that it runs on JASMIN.

Original versions:

- Suite: u-ai227 at <https://code.metoffice.gov.uk/svn/roses-u/a/i/2/2/7/>
- Branch:  
[https://code.metoffice.gov.uk/trac/jules/log/main/branches/dev/tobymarthews/vn4.8\\_testHoT?rev=7219](https://code.metoffice.gov.uk/trac/jules/log/main/branches/dev/tobymarthews/vn4.8_testHoT?rev=7219)

On PUMA, run `rosie go`. Choose relevant suite (e.g. u-al609), right click and select Copy. This both copies and checks out the suite, making it editable under home directory. Example here: u-am232. Open suite, either by right clicking on the suite and then 'Edit suite', or the Edit button.

### **Things to change within Rose**

- 1) UserID: even though username is changed when suite is copied, userID still needs to be changed manually in `roses/<suiteID>/suite.rc` in home directory on PUMA
- 2) Metadata: both the metadata for both `fcmake` and `jules` need to be copied from Ros' directory (at `/home/ros/meta/vn4.8_testHoT` where she made a small change to Toby's branch in order to add in the `jasmin-lotus-intel` option) to home directory on PUMA. Then, in Rose (in `fcmake > meta` and `jules > meta` respectively) metadata should point to same place e.g.:  
`/home/charlie/jules/v48_toby/meta/vn4.8_testHoT/rose-meta/jules-standalone/vn4.8`
- 3) Build: another copy of same branch needs to be copied from Ros' directory (at `/home/users/rshatcher/branches/vn4.8_testHoT`) to home directory on JASMIN. Then, in Rose (in `fcmake > Build configuration`) path to point to this e.g.:  
`/home/users/cwilliams2011/jules/v48_toby/branches/vn4.8_testHoT`

*Note: Because suite is being submitted from PUMA, this is where metadata are used. But extract and compile is being done on JASMIN, so build configuration needs to reside on JASMIN.*

- 4) Warning triangles: for unknown reasons, despite metadata pointing to correct location, there will be a lot of red warning triangles throughout suite. Possible to get rid of some of these, by going into relevant section and clicking on red cross - this will either resolve error or remove it completely. Some errors will be obvious (e.g. dates incorrectly formatted). However, even after removing these errors, some warning triangles will remain despite no obvious problem in relevant section. These can be ignored!
- 5) Driving data: need to decide whether suite is to run with 1D or 2D versions of driving/ancillary data, and set namelists (especially ancillaries.nml and model\_grid.nml) accordingly. From here on, 1D version is used. For driving data, WFDEI (which = E2OBS from Earth2observe project) in 2D is on jules\_bd group workspace, at /group\_workspaces/jasmin2/jules\_bd/data/E2OBS/met\_data. However, access to this is restricted and is not generally allowed for most JULES users (it holds sensitive MAJIC application). With very polite asking, group workspace manager *might* allow read-only access. If not, or if 1D version is wanted, WFDEI in 1D is at /group\_workspaces/jasmin/jules/data/WFD-EI-Forcing.
- 6) Ancillary data: if using 1D version of driving/ancillary data (which is faster to run), ancillary files are in /group\_workspaces/jasmin/jules/data/WFD-EI-Forcing/ancils. Start dumps are also here (in WFD-EI-Forcing/dumps).

*Note: v4.8 now stores ancillary data in dump files, so need to decide whether to start from a dump file and take ancillary data from here (in which case, under each ancillary namelist, read\_from\_dump should be set to .true.) OR only use dump files to restart and specify ancillary data from individual files. Advantage of former way = only have one file to move around so easier to change platform. Disadvantage = harder to remember where original data came from. Also, using a dump file requires the correct start date. From here on, doing latter way.*

Once decided, apply pathnames to relevant ancillary files (in jules > Namelist configuration > Ancillary data > ...). Only ancillary file not in above locations is river ancillary file, which is always 2D (regardless of which is chosen), which is in /home/users/cwilliams2011/tilaka.d/jules.d/ancils.d/river\_ancil\_global05deg\_MASTE

R\_WFD\_WFDEI\_and\_extra\_layers.nc. If old version of river ancillary file is used, it doesn't contain the logn\_mean and logn\_stdev layers, which are now needed.

*Note: these layers don't actually have any data north of 60°N*

- 7) If copying namelists from older ones (e.g. before Rose), much easier to do it manually rather than using Rose GUI. All namelists from Rose suite are amalgamated in roses/<suiteID>/app/jules/rose-app.conf where namelists resemble old-style, so easier to simply paste in from old versions.

*Note: in this file, namelists are in alphabetical order, not grouped together as before.*

*Also, note: in old version of namelists, there was always a , after every line. In new version, there isn't. So don't simply paste in old version, as otherwise will have to remove all , which is a major faff!*

- 8) Modify executable at the top of roses/<suiteID>/app/jules/rose-app.conf, so that it uses MPI and directs output to separate files (see below)
- 9) Once everything has been correctly setup, including driving data, ancillary files, run length & timestepping, spin-up, initial conditions, output diagnostics (all in jules > Namelist configuration > ...), can begin the run, either by pressing the 'Play' button, or rose suite-run -C <suiteID>

## Queueing

5 queues are available on JASMIN, with different priorities and allowable maximum runtimes. If running short chunks (< 6 hours), best to use short-serial as it has highest priority. Don't specify a runtime near the maximum, as otherwise suite will be put to the back of the queue. Even using highest priority queue, waiting time might be several hours (e.g. up to 24 hours). JASMIN people sometimes freeze/reset queues over the weekend.

Name	Maximum runtime	Default time	Priority
short-serial	24hrs	1hr	30
par-single	48hrs	1hr	25
par-multi	48hrs	1hr	20
long-serial	168hrs	1hr	10
high-mem	48hrs	1hr	30

To specify which queue, modify bottom of roses/<suiteID>/suite.rc so that -q = desired queue and -W = desired runtime (in hours and minutes).

To check status of suite (and which queue) is being used, use `bjobs`.

To change queues (while still pending) use `bmod -q <queue-name> <jobID>`, where `<queue-name>` = desired new queue and `<jobID>` = job number (first column when using `bjobs`)

## Restarting

Things to change when restarting from a dump:

- 1) Runtime (specify new start and end dates): in `[namelist:jules_time]`
- 2) Output (change start and end dates to match runtime, and change `run_id` to a new name: in `[namelist:jules_output]`, `[namelist:jules_output_profile(x)]` where `x` = profile number
- 3) Initial conditions: in `[namelist:jules_initial]`
  - If starting without a dump:

```
const_val=0.0,275.0,10.0,0.0,50.0,0.75,0.75,278.0,0.759,3.0
dump_file=.false.
file=""
nvars=10
total_snow=.true.
use_file=10*.false.
var='canopy','tstar_tile','cs','gs','rgrain','snow_tile',
'sthuf','t_soil','sthzw','zw'
```
  - If restarting from a dump:

```
dump_file=.true.,
total_snow=.false.,
file="<path/dump>"
```
- 4) Once an initial run has finished, possible to avoid the `fcmake` step on restart using `rose app-run`: `rose app-run -C <suiteID>/app/jules >job.out`. However, time saved doing this is minimal (`fcmake` won't recompile every time, but only after a change to the source code), so not necessary.

## Output monitoring

Each home directory on JASMIN has default space of 10GB. To check space (whilst in home directory), use `pan_quota`. To check space on group workspace (whilst in workspace), use `pan_df`.

Once suite is running, a new directory, `cylc-run`, will be created in home directory on both PUMA and JASMIN. This contains, among other things, job status and error log files, as named below. Files are in `cylc-run/<suiteID>/log/job/1/jules/01`. These won't appear on PUMA until the end of the run, however they will appear immediately on JASMIN. To view the [INFO] lines as suite is running, use `tail -f run.log`

Within Cycl GUI, once suite is finished (either succeeded or failed), can right click on View > job stdout\* to look at how far it got (this is where the [INFO] lines, which used to appear at the terminal, are now diverted to). If it has failed, right click on View > job stderr\* to look at errors. If Cycl GUI freezes, or nothing happens, check status of suite using `bjobs`. If Cycl GUI is accidentally shut down, restart it with `rose sgc`. This won't affect the running suite.

*\* Note: For running JULES in parallel (on JASMIN) only, these files don't seem to be generated whilst suite is running, only afterwards, which is of little use. To get this output to be generated whilst running, modify <suiteID>/app/jules/rose-app.conf at the top, e.g.*

[command]

```
default=/usr/local/bin/mpirun.lotus /home/users/cwilliams2011/cylc-run/u-am232/  
share/fcm_make/build/bin/jules.exe > [/home/users/cwilliams2011/cylc-run/u-  
am232/log/job/1/jules/01/run.log 2> /home/users/cwilliams2011/cylc-run/u-  
am232/log/job/1/jules/01/error.log
```

*The first part points the JULES executable to the MPI on Lotus - if this is not included, JULES will run on serial rather than parallel and will be much slower. The second parts point the output and errors to 2 separate files, respectively, which can be defined according to user.*

## Fixing errors & debugging

- 1) Assuming metadata and build configuration is correctly setup, first task (fcm\_make) will successfully run (relatively quickly), before moving on to the second task (jules). Once it begins running, it will probably fail. Check errors as above. Likely that error will be either trying to find a variable and failing. This is because there have been several changes since v4.7, e.g. there is now a new [namelist:jules\_soil\_biogeochem] which contains several variables from several other namelists e.g. kaps (which used to be in [namelist:jules\_surface]). Very difficult to know which ones have been moved, so only way is to run suite, look at error file, then type variable into search box at <http://jules-lsm.github.io/vn4.8/>. This should say where the variable should be.
- 2) Interaction between Rose GUI and rose-app.conf (which, namelists, are essentially are the same thing and can therefore modify each other) is sometimes flakey. If changes are made in rose-app.conf whilst GUI is open, then GUI should be shut down and restarted before submitting (should also do this if changes are made to suite.rc whilst GUI is open). Likewise, if changes are made in GUI whilst rose-app.conf is open, then errors (e.g. random punctuation) often appear in rose-app.conf e.g. output diagnostics are sometimes repeated, causing suite failure. So always check both sources when changes are made to one or other.
- 3) If Cycl GUI freezes, or need to stop/kill the suite: right click on task (e.g. jules) then 'Kill'. Then `rose suite-shutdown --name=<suiteID>` to completely shutdown the suite. Have to wait a few minutes before resubmitting - sometimes to submit twice, ignoring error which appears. Unusually, this may not completely kill the suite. If so:
  - a) If Cycl GUI has shut down or frozen, run `cycl gscan` or `rose sgc` to reopen it
  - b) Double-click on all tasks to open windows for each and leave open
  - c) For each task, reset the status of jules to 'failed' and then click 'Stop'
  - d) At terminal, run `top -u <userID>` and then `kill -9 <PID>` to kill any jules.exe still running
  - e) This may not clearly show all processes, so if suite is still running (e.g. if another suite cannot be started), use `ps -flu <userID> | grep <suiteID>`. This should list any processes still running (e.g. "python..."). Kill these processes (using the number in the 4<sup>th</sup> column)
  - f) Finally, delete the entire directory `cycl-run` on both JASMIN and PUMA
- 4) On rare occasions, restarting won't be allowed because it thinks there is still a file within `cycl-run` on JASMIN on PUMA (e.g. in `cycl-run/<suiteID>/log/job/1/jules/01`),

despite this entire directory being deleted. This is because, very occasionally, the system is very slow to release a file lock - it *usually* resolves itself after 24 hours, so wait and then try again.