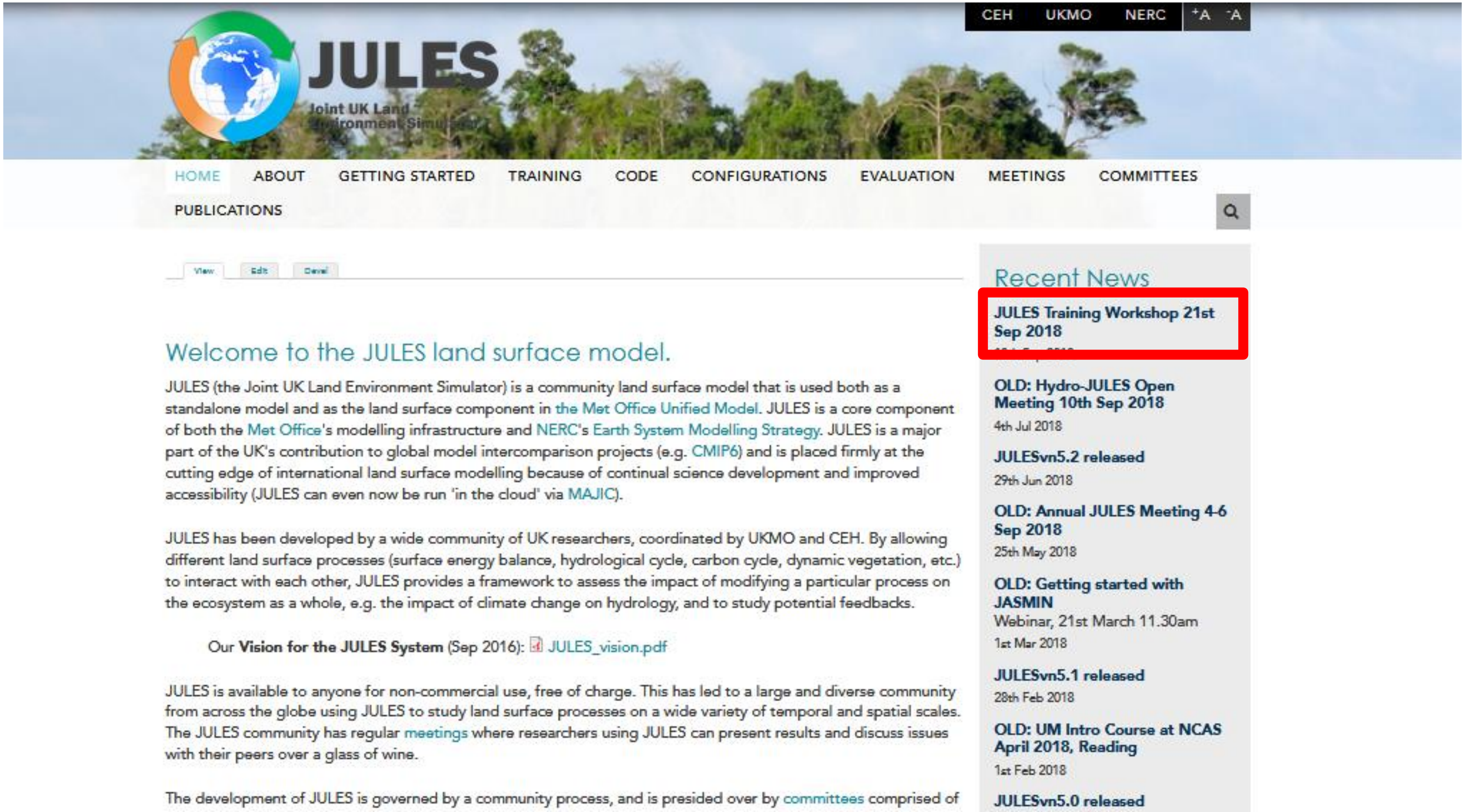
The JULES land surface model

http://jules.jchmr.org



Welcome to the JULES land surface model.

JULES (the Joint UK Land Environment Simulator) is a community land surface model that is used both as a standalone model and as the land surface component in the Met Office Unified Model. JULES is a core component of both the Met Office's modelling infrastructure and NERC's Earth System Modelling Strategy. JULES is a major part of the UK's contribution to global model intercomparison projects (e.g. CMIP6) and is placed firmly at the cutting edge of international land surface modelling because of continual science development and improved accessibility (JULES can even now be run 'in the cloud' via MAJIC).

JULES has been developed by a wide community of UK researchers, coordinated by UKMO and CEH. By allowing different land surface processes (surface energy balance, hydrological cycle, carbon cycle, dynamic vegetation, etc.) to interact with each other, JULES provides a framework to assess the impact of modifying a particular process on the ecosystem as a whole, e.g. the impact of climate change on hydrology, and to study potential feedbacks.

Our **Vision for the JULES System** (Sep 2016): JULES_vision.pdf

JULES is available to anyone for non-commercial use, free of charge. This has led to a large and diverse community from across the globe using JULES to study land surface processes on a wide variety of temporal and spatial scales. The JULES community has regular meetings where researchers using JULES can present results and discuss issues with their peers over a glass of wine.

The development of JULES is governed by a community process, and is presided over by committees comprised of

Recent News

JULES Training Workshop 21st Sep 2018

OLD: Hydro-JULES Open Meeting 10th Sep 2018
4th Jul 2018

JULESvn5.2 released
29th Jun 2018

OLD: Annual JULES Meeting 4-6 Sep 2018
25th May 2018

OLD: Getting started with JASMIN
Webinar, 21st March 11.30am
1st Mar 2018

JULESvn5.1 released
28th Feb 2018

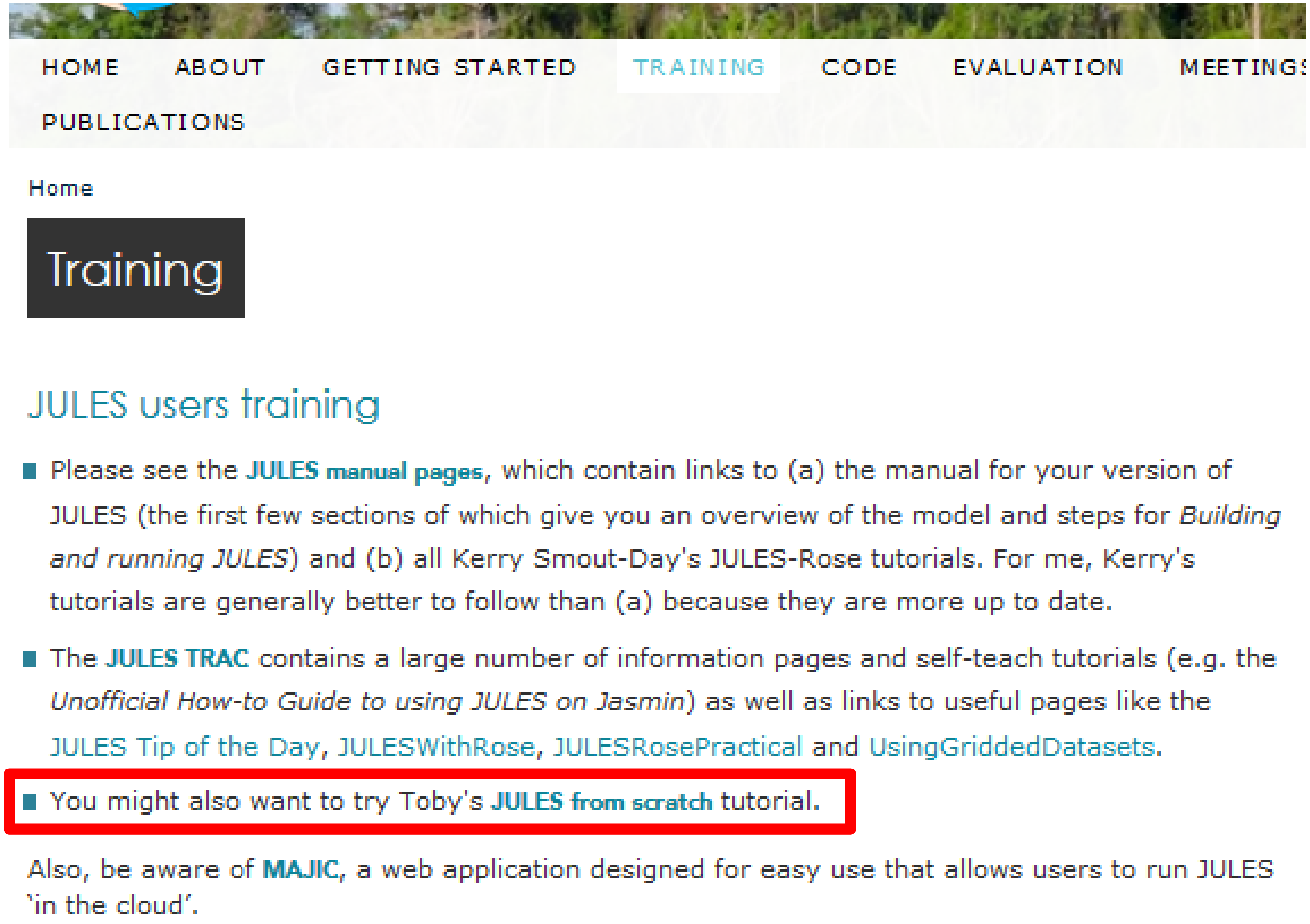
OLD: UM Intro Course at NCAS April 2018, Reading
1st Feb 2018

JULESvn5.0 released

The JULES land surface model

http://jules.jchmr.org

The JULES website has lots of training tutorials (many of them actually on the JULES TRAC or elsewhere).



Training

JULES users training

- Please see the **JULES manual pages**, which contain links to (a) the manual for your version of JULES (the first few sections of which give you an overview of the model and steps for *Building and running JULES*) and (b) all Kerry Smout-Day's JULES-Rose tutorials. For me, Kerry's tutorials are generally better to follow than (a) because they are more up to date.
- The **JULES TRAC** contains a large number of information pages and self-teach tutorials (e.g. the *Unofficial How-to Guide to using JULES on Jasmin*) as well as links to useful pages like the JULES Tip of the Day, JULESWithRose, JULESRosePractical and UsingGriddedDatasets.
- You might also want to try Toby's **JULES from scratch** tutorial.

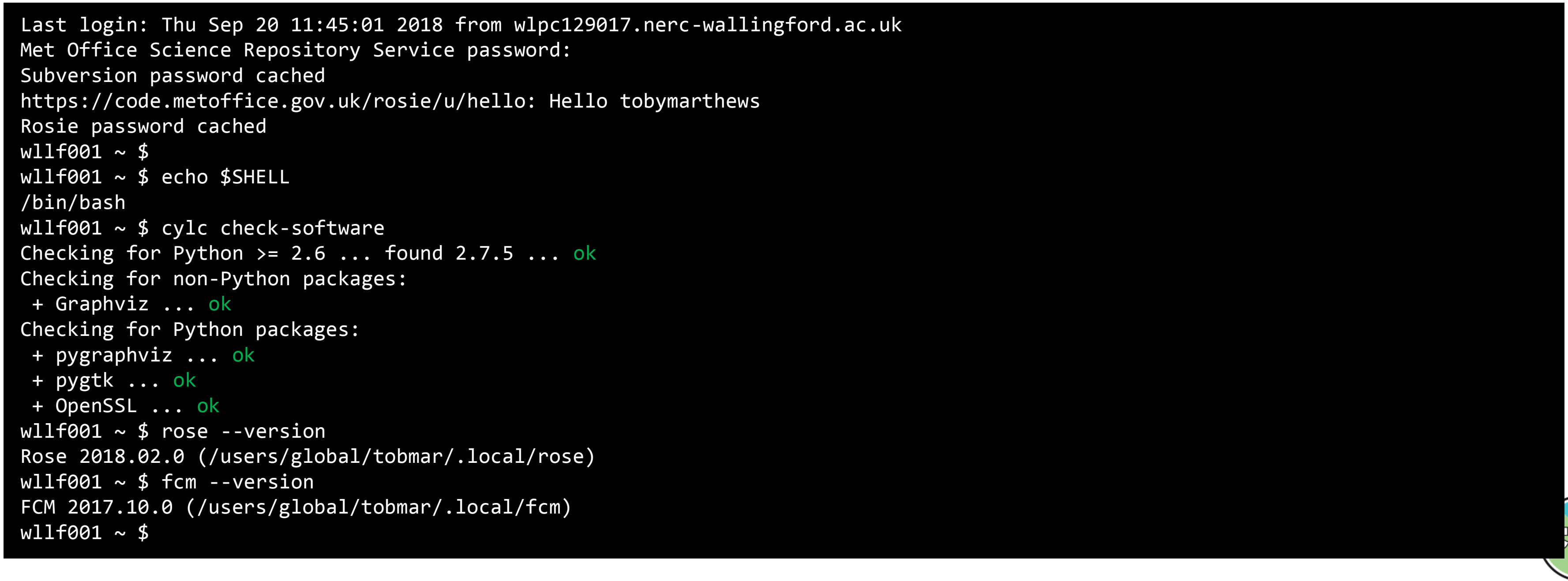
Also, be aware of **MAJIC**, a web application designed for easy use that allows users to run JULES 'in the cloud'.

JULES from Scratch

The *JULES from Scratch* tutorial is my attempt to lead you through all you need to get started on JULES. It takes you through setting up a JULES run right from the basics, including all required installation steps. Start on this page:

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

First thing I do there is to check you have *Cylc*, *Rose* and *FCM* installed on your system (if you don't have them, there are instructions on that link how to install them) and cache your *MOSRS* password (covered earlier today):

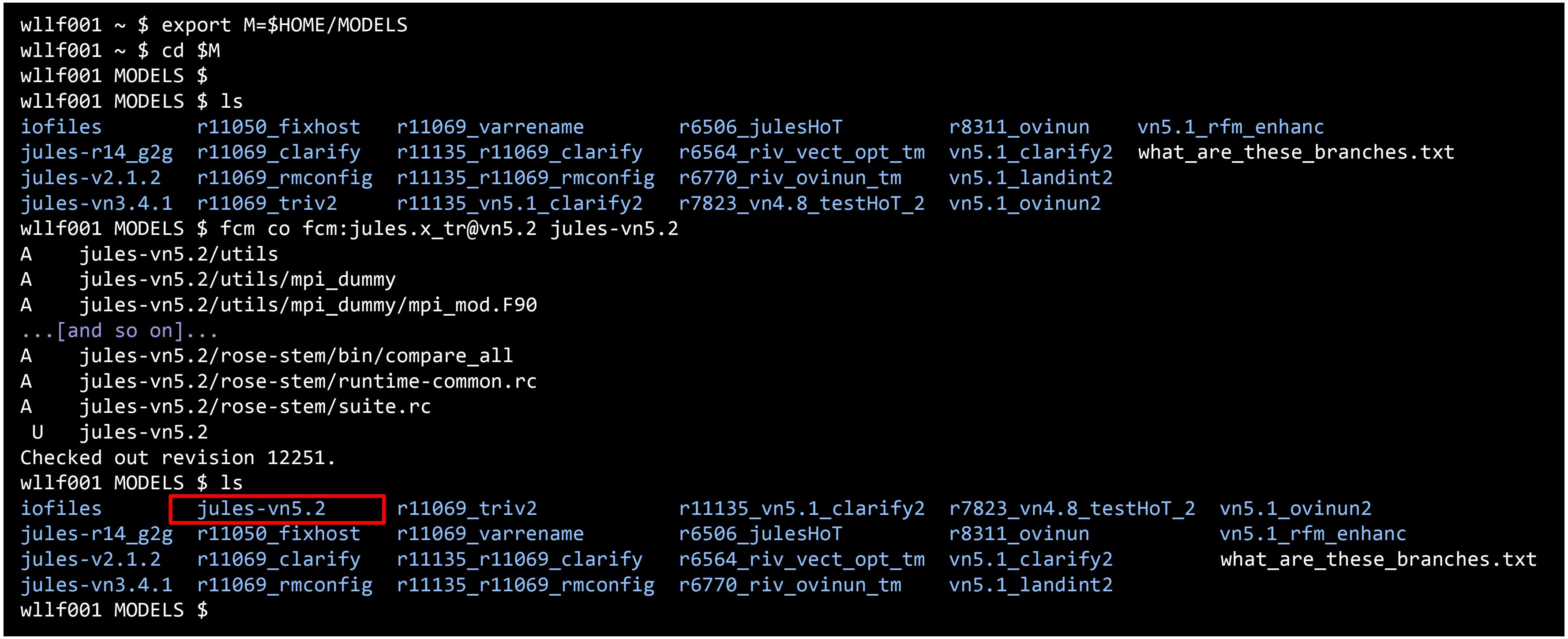


```
Last login: Thu Sep 20 11:45:01 2018 from wlpc129017.nerc-wallingford.ac.uk
Met Office Science Repository Service password:
Subversion password cached
https://code.metoffice.gov.uk/rosie/u/hello: Hello tobymarthews
Rosie password cached
wllf001 ~ $
wllf001 ~ $ echo $SHELL
/bin/bash
wllf001 ~ $ cylc check-software
Checking for Python >= 2.6 ... found 2.7.5 ... ok
Checking for non-Python packages:
 + Graphviz ... ok
Checking for Python packages:
 + pygraphviz ... ok
 + pygtk ... ok
 + OpenSSL ... ok
wllf001 ~ $ rose --version
Rose 2018.02.0 (/users/global/tobmar/.local/rose)
wllf001 ~ $ fcm --version
FCM 2017.10.0 (/users/global/tobmar/.local/fcm)
wllf001 ~ $
```

Install JULES vn5.2

I keep all my various versions of JULES in a directory $HOME/MODELS/, which I’m going to refer to as $M in all these instructions. Put a copy of JULESvn5.2 in $M using:

fcm co fcm:jules.x_tr@vn5.2 jules-vn5.2

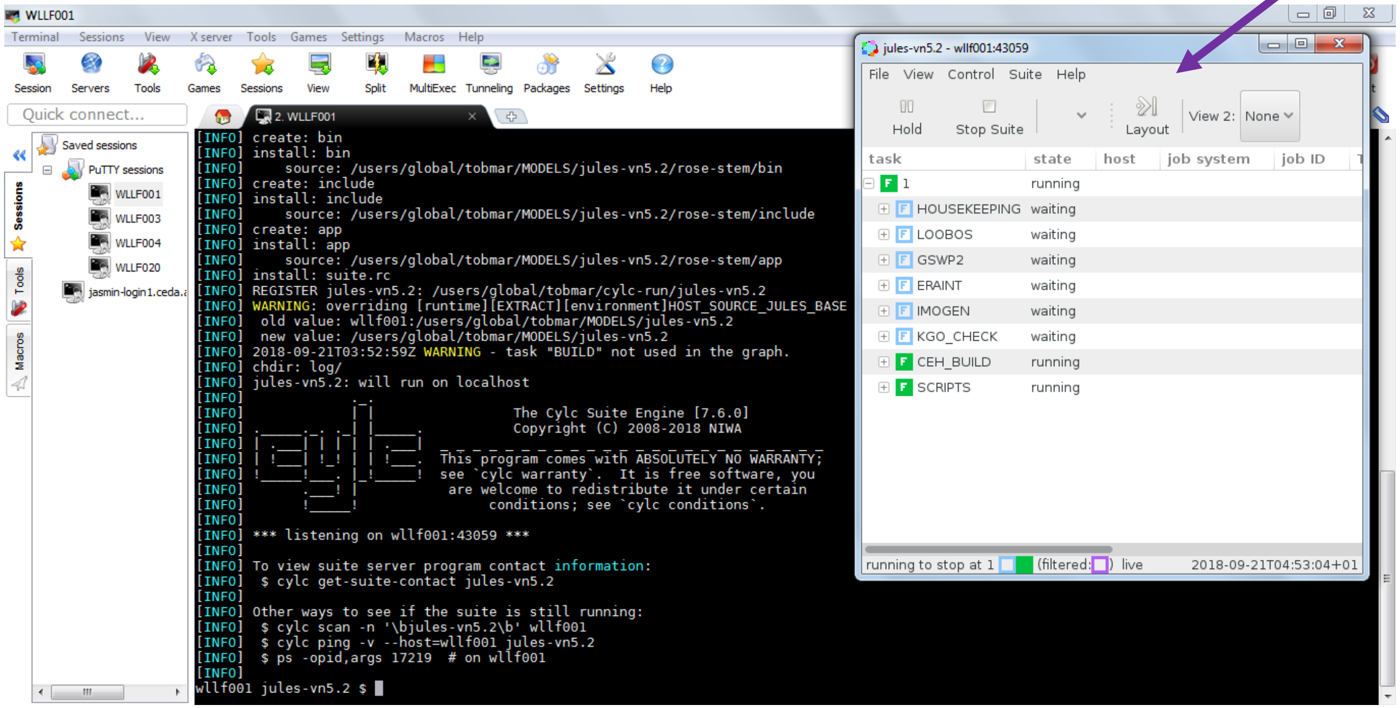


```
wllf001 ~ $ export M=$HOME/MODELS
wllf001 ~ $ cd $M
wllf001 MODELS $
wllf001 MODELS $ ls
iofiles        r11050_fixhost   r11069_varrename        r6506_julesHoT         r8311_ovinun    vn5.1_rfm_enhanc
jules-r14_g2g  r11069_clarify   r11135_r11069_clarify   r6564_riv_vect_opt_tm  vn5.1_clarify2  what_are_these_branches.txt
jules-v2.1.2   r11069_rmconfig  r11135_r11069_rmconfig  r6770_riv_ovinun_tm    vn5.1_landint2
jules-vn3.4.1  r11069_triv2     r11135_vn5.1_clarify2   r7823_vn4.8_testHoT_2  vn5.1_ovinun2
wllf001 MODELS $ fcm co fcm:jules.x_tr@vn5.2 jules-vn5.2
A    jules-vn5.2/utils
A    jules-vn5.2/utils/mpi_dummy
A    jules-vn5.2/utils/mpi_dummy/mpi_mod.F90
...[and so on]...
A    jules-vn5.2/rose-stem/bin/compare_all
A    jules-vn5.2/rose-stem/runtime-common.rc
A    jules-vn5.2/rose-stem/suite.rc
 U   jules-vn5.2
Checked out revision 12251.
wllf001 MODELS $ ls
iofiles        jules-vn5.2      r11069_triv2            r11135_vn5.1_clarify2  r7823_vn4.8_testHoT_2  vn5.1_ovinun2
jules-r14_g2g  r11050_fixhost   r11069_varrename        r6506_julesHoT         r8311_ovinun           vn5.1_rfm_enhanc
jules-v2.1.2   r11069_clarify   r11135_r11069_clarify   r6564_riv_vect_opt_tm  vn5.1_clarify2         what_are_these_branches.txt
jules-vn3.4.1  r11069_rmconfig  r11135_r11069_rmconfig  r6770_riv_ovinun_tm    vn5.1_landint2
wllf001 MODELS $
```

Check installation

Once jules-vn5.2 has been downloaded (‘checked out’), it’s best to check it with a basic run. If you are working at a site with KGO installed and set up (“Known Good Output”), then you can simply run the in-built Rose Stem tests with no further changes like this:

```
cd jules-vn5.2
rose stem --group=all --source=. --new
```


WLLF001
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. WLLF001
Saved sessions
PuTTY sessions
WLLF001
WLLF003
WLLF004
WLLF020
jasmin-login1.ceda.a
Sessions
Tools
Macros
jules-vn5.2 - wllf001:43059
File View Control Suite Help
Hold
Stop Suite
Layout
View 2: None

```
[INFO] create: bin
[INFO] install: bin
[INFO]     source: /users/global/tobmar/MODELS/jules-vn5.2/rose-stem/bin
[INFO] create: include
[INFO] install: include
[INFO]     source: /users/global/tobmar/MODELS/jules-vn5.2/rose-stem/include
[INFO] create: app
[INFO] install: app
[INFO]     source: /users/global/tobmar/MODELS/jules-vn5.2/rose-stem/app
[INFO] install: suite.rc
[INFO] REGISTER jules-vn5.2: /users/global/tobmar/cylc-run/jules-vn5.2
[INFO] WARNING: overriding [runtime][EXTRACT][environment]HOST_SOURCE_JULES_BASE
[INFO]  old value: wllf001:/users/global/tobmar/MODELS/jules-vn5.2
[INFO]  new value: /users/global/tobmar/MODELS/jules-vn5.2
[INFO] 2018-09-21T03:52:59Z WARNING - task "BUILD" not used in the graph.
[INFO] chdir: log/
[INFO] jules-vn5.2: will run on localhost
[INFO]              ._.
[INFO]              | |                 The Cylc Suite Engine [7.6.0]
[INFO] ._____._. ._| |_____.           Copyright (C) 2008-2018 NIWA
[INFO] | .___| | | | | .___|
[INFO] | !___| !_! | | !___.  This program comes with ABSOLUTELY NO WARRANTY;
[INFO] !_____!___. |_!_____!  see `cylc warranty`.  It is free software, you
[INFO]       .___! |           are welcome to redistribute it under certain
[INFO]       !_____!               conditions; see `cylc conditions`.
[INFO]
[INFO] *** listening on wllf001:43059 ***
[INFO]
[INFO] To view suite server program contact information:
[INFO]  $ cylc get-suite-contact jules-vn5.2
[INFO]
[INFO] Other ways to see if the suite is still running:
[INFO]  $ cylc scan -n '\bjules-vn5.2\b' wllf001
[INFO]  $ cylc ping -v --host=wllf001 jules-vn5.2
[INFO]  $ ps -opid,args 17219  # on wllf001
[INFO]
wllf001 jules-vn5.2 $ █
```

task	state	host	job system	job ID
⊟ F 1	running			
⊞ F HOUSEKEEPING	waiting			
⊞ F LOOBOS	waiting			
⊞ F GSWP2	waiting			
⊞ F ERAINT	waiting			
⊞ F IMOGEN	waiting			
⊞ F KGO_CHECK	waiting			
⊞ F CEH_BUILD	running			
⊞ F SCRIPTS	running			

running to stop at 1 □🟩 (filtered:□) live 2018-09-21T04:53:04+01

New ‘Cylc GUI’ window showing the progress of the Rose Stem tests.

If you close this window by accident before they complete, type:

cylc gscan &

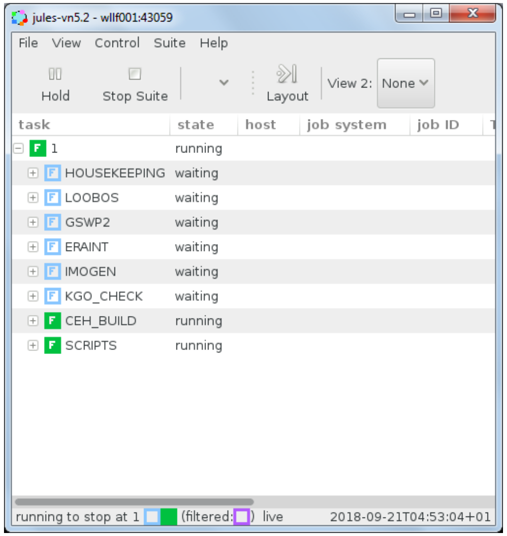
and double-click on the right job to find it again.

To save typing, add this line to your ~/.bashrc file and you’ll be able to abbreviate it to just “rstem” instead of that command:
alias rstem='rose stem --group=all --source=. --new'

Check installation

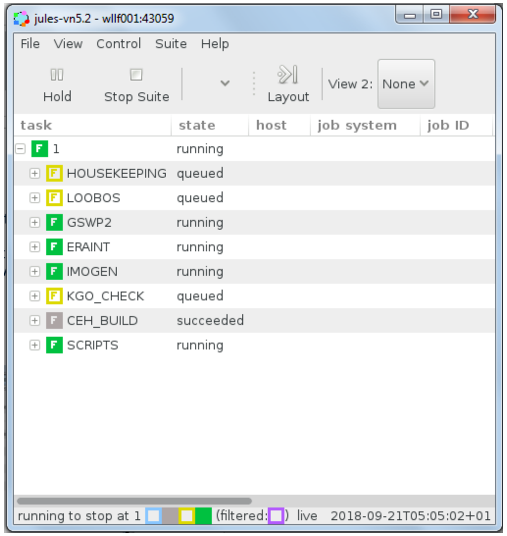
If these tests all pass, then you know JULESvn5.2 has been installed properly.

These rose stem tests are important for developing the code, but we're not interested in that here: all the tests *should* pass, but we're just using them as a quick way of verifying the installation so really as soon as the Cylc GUI window comes up we've achieved that.



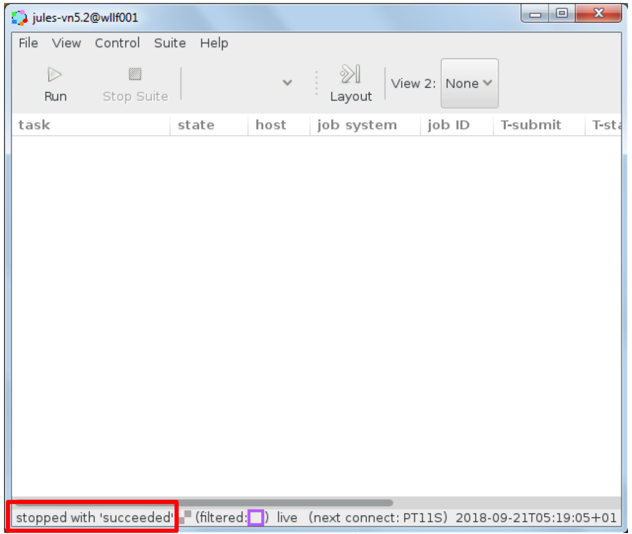
task	state	host	job system	job ID	T
F 1	running				
F HOUSEKEEPING	waiting				
F LOOBOS	waiting				
F GSWP2	waiting				
F ERAINT	waiting				
F IMOGEN	waiting				
F KGO_CHECK	waiting				
F CEH_BUILD	running				
F SCRIPTS	running				

running to stop at 1 (filtered:) live 2018-09-21T04:53:04+01



task	state	host	job system	job ID
F 1	running			
F HOUSEKEEPING	queued			
F LOOBOS	queued			
F GSWP2	running			
F ERAINT	running			
F IMOGEN	running			
F KGO_CHECK	queued			
F CEH_BUILD	succeeded			
F SCRIPTS	running			

running to stop at 1 (filtered:) live 2018-09-21T05:05:02+01



task	state	host	job system	job ID	T-submit	T-sta

stopped with 'succeeded' (filtered:) live (next connect: PT11S) 2018-09-21T05:19:05+01

Approx. 15 min

Note 1: The "Stop Suite" button doesn't actually stop the whole suite (!!) - see 'Troubleshooting' slide below
Note 2: The jobs will all disappear when the last one completes (there's no option for changing that).

Check installation

If you *don't* work at a site with KGO installed and set up, then you can't run rose stem tests.

Doing a check of your installation is still a good idea, however, which you can do by downloading and running a small test suite. The traditional test suite for the JULES community is based on data from an eddy flux tower in the Netherlands called Loobos. I've prepared a version of this compatible with JULESvn5.2 called 'Loobos_orig' and you can download it from http://jules.jchmr.org/content/modify-suite.

Unlike with the rose stem tests, which use site settings to automatically be able to run on your platform, you will need to make some modifications to that Loobos_orig suite to have it run on your system. The details are on that page too, but don't follow those steps yet (save for later) because I want to explain about Rose suites first.

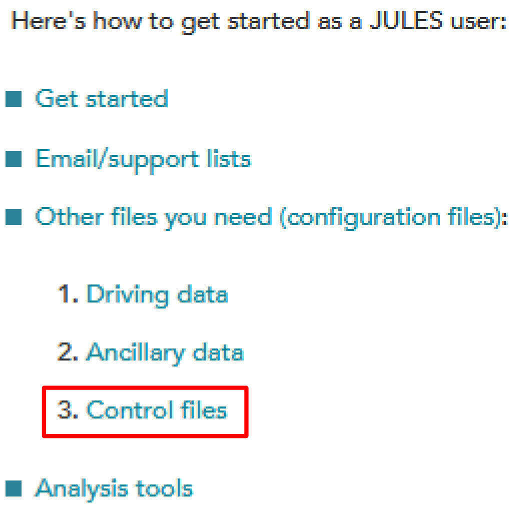
Doing your own simulation

Now we're all set up and can actually use JULES for something.

In common with any model of this type, JULES needs THREE elements to do a simulation:

- *Driving data*,
- *Ancillary/prescribed data* and
- *Control files*

(together these are called the **model configuration**). I've put advice on where to get these from on http://jules.jchmr.org/content/getting-started.



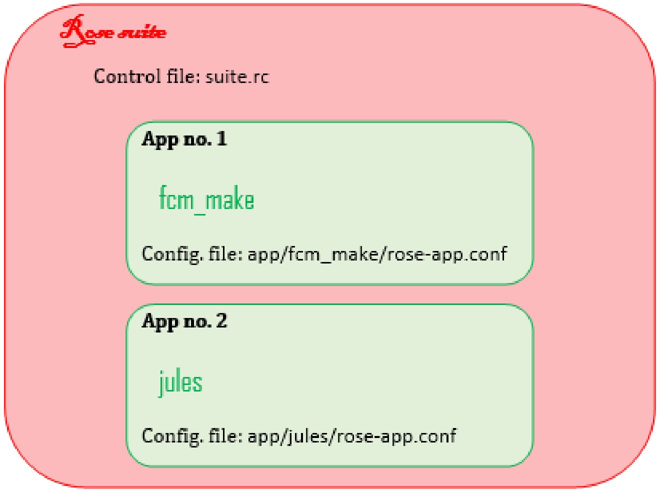
Here's how to get started as a JULES user:

- Get started
- Email/support lists
- Other files you need (configuration files):
  1. Driving data
  2. Ancillary data
  3. Control files
- Analysis tools

For the control files, these usually come in the form of a ***Rose suite***.

Rose suites: what are they?

A *Rose suite* is a ‘container’ for *apps*. JULES Rose suites always have two apps that run sequentially: *fcm_make* (the compilation step) and *jules* (actually running the model) (see http://jules.jchmr.org/content/rose-suites).



Rose suite

Control file: suite.rc

App no. 1

fcm_make

Config. file: app/fcm_make/rose-app.conf

App no. 2

jules

Config. file: app/jules/rose-app.conf

For example, my suite u-am539 is stored on my system at ~/roses/u-am539/ (which from now on I’m going to refer to as $RSUITE):

```
wllf001 ~ $ export RSUITE=$HOME/roses/u-am539
wllf001 ~ $ echo $RSUITE
/users/global/tobmar/roses/u-am539
wllf001 ~ $
```

I can open this suite in two different ways:

1. On UNIX using the Rose Edit GUI: `rose edit -C $RSUITE &`

2. On UNIX using a text editor: `nedit $RSUITE/rose-suite.info $RSUITE/suite.rc $RSUITE/app/fcm_make/rose-app.conf $RSUITE/app/jules/rose-app.conf &`

(i.e. opening four text files at once: the suite info file, suite control file, config file for the compile and config file for JULES. It’s the last of these that contains all the parameters and options for the JULES run).

When editing JULES suites, all parameters are explained on the JULES manual pages so I keep that open at the same time too (there is a search box lower left for you to find any parameter you are unfamiliar with):
http://jules-lsm.github.io/vn5.2/namelists/contents.html

Rose suites: the Rose Editor

You can edit a Rose suite in TWO different ways:

1. On UNIX using the Rose Edit GUI:

```
rose edit -C $RSUITE &
```

Tip: Always change your settings in the Editor so that View -> View All Ignored Variables is selected. This is because the search box on the main screen only searches visible parameters (e.g. try searching for *l_veg_compete*).

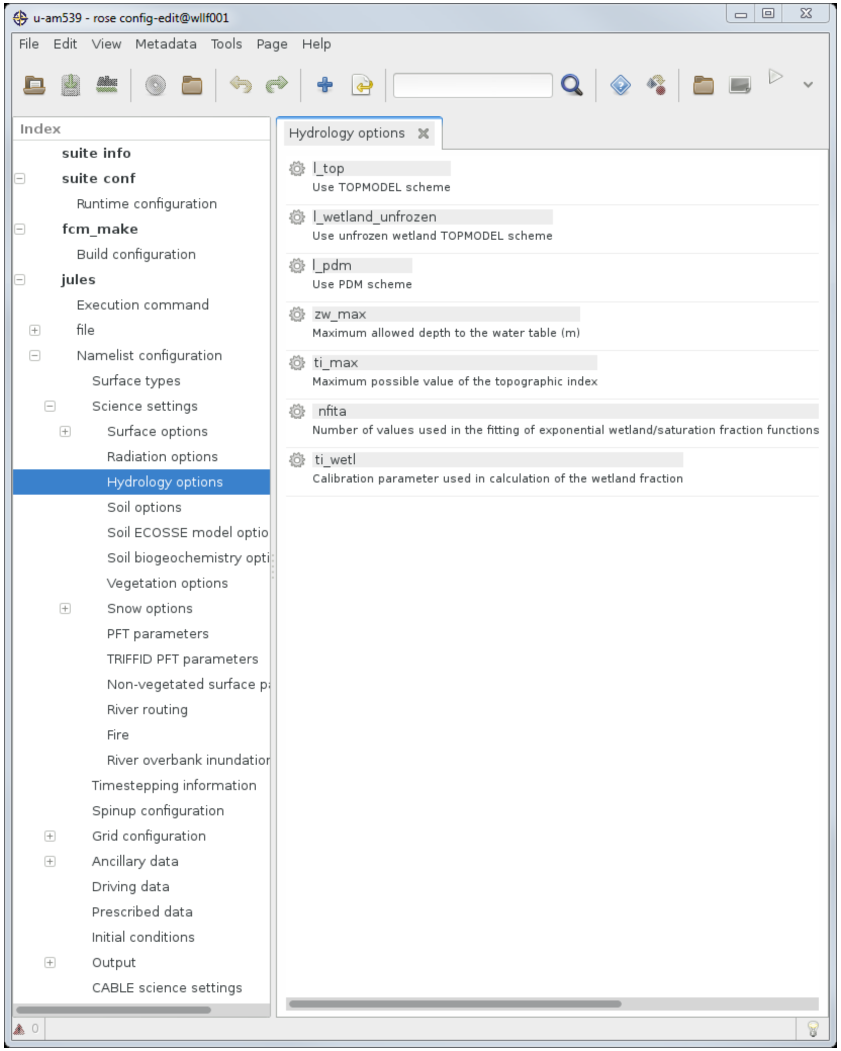
2. On UNIX using a text editor:

```
nedit $RSUITE/rose-suite.info
$RSUITE/suite.rc
$RSUITE/app/fcm_make/rose-app.conf $RSUITE/app/jules/rose-app.conf &
```

(i.e. opening four text files at once: the suite info file, suite control file, config file for the compile and config file for JULES. It's the last of these that contains all the parameters and options for the JULES run).

When editing JULES suites, all parameters are explained on the JULES manual pages so I usually keep that open at the same time too (there is a search box lower left for you to find any parameter you are unfamiliar with):

http://jules-lsm.github.io/vn5.2/namelists/contents.html



u-am539 - rose config-edit@wllf001

Index

- suite info
- suite conf
  - Runtime configuration
- fcm_make
  - Build configuration
- jules
  - Execution command
  - file
  - Namelist configuration
    - Surface types
    - Science settings
      - Surface options
      - Radiation options
      - Hydrology options
      - Soil options
      - Soil ECOSSE model optio
      - Soil biogeochemistry opti
      - Vegetation options
      - Snow options
      - PFT parameters
      - TRIFFID PFT parameters
      - Non-vegetated surface p
      - River routing
      - Fire
      - River overbank inundatior
    - Timestepping information
    - Spinup configuration
    - Grid configuration
    - Ancillary data
    - Driving data
    - Prescribed data
    - Initial conditions
    - Output
    - CABLE science settings

Hydrology options

l_top
Use TOPMODEL scheme

l_wetland_unfrozen
Use unfrozen wetland TOPMODEL scheme

l_pdm
Use PDM scheme

zw_max
Maximum allowed depth to the water table (m)

ti_max
Maximum possible value of the topographic index

nfita
Number of values used in the fitting of exponential wetland/saturation fraction functions

ti_wetl
Calibration parameter used in calculation of the wetland fraction

Rose suites: where to get them?

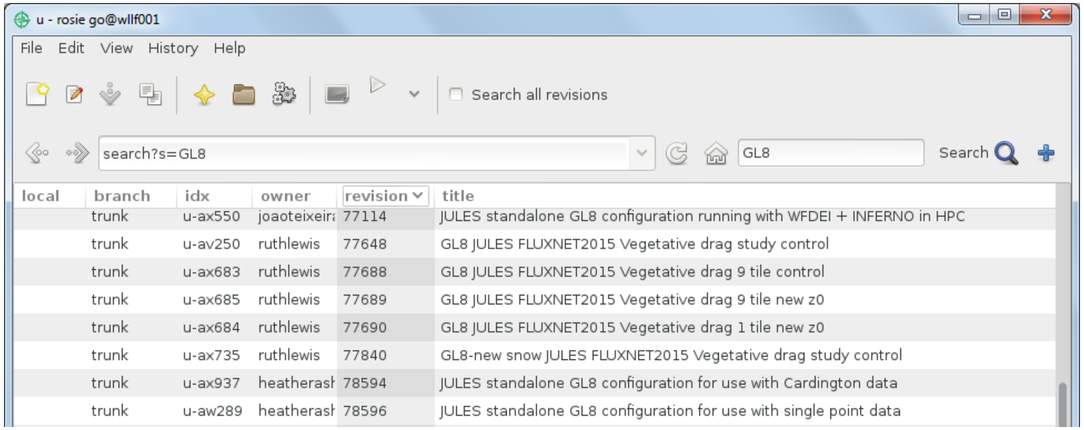
See information on http://jules.jchmr.org/content/getting-started#control-files for how to get hold of Rose suites. There are basically two ways:

1. Download a suite from one of the (many) standard configurations on http://jules.jchmr.org/content/configurations

2. On UNIX using the *Rose Suite Discovery Engine*: `rosie go &`

(select Edit → the ‘u’ data source, then use the search box). To download, choose:

- **Checkout Suite** to take a duplicate of it
- or **Copy Suite** to use it for your own work (you’ll get a new suite ID and this will belong to you).



☐ Search all revisions

search?s=GL8

GL8

local	branch	idx	owner	revision	title
	trunk	u-ax550	joaoteixeira	77114	JULES standalone GL8 configuration running with WFDEI + INFERNO in HPC
	trunk	u-av250	ruthlewis	77648	GL8 JULES FLUXNET2015 Vegetative drag study control
	trunk	u-ax683	ruthlewis	77688	GL8 JULES FLUXNET2015 Vegetative drag 9 tile control
	trunk	u-ax685	ruthlewis	77689	GL8 JULES FLUXNET2015 Vegetative drag 9 tile new z0
	trunk	u-ax684	ruthlewis	77690	GL8 JULES FLUXNET2015 Vegetative drag 1 tile new z0
	trunk	u-ax735	ruthlewis	77840	GL8-new snow JULES FLUXNET2015 Vegetative drag study control
	trunk	u-ax937	heatherash	78594	JULES standalone GL8 configuration for use with Cardington data
	trunk	u-aw289	heatherash	78596	JULES standalone GL8 configuration for use with single point data

The *Rosie Go* window. *Rosie Go* expects a directory ~/roses/ to be on your system, which it uses as a store for downloaded Rose suites.

Every suite has its own URL and you can use this to see the changes to the suite over time, e.g.

https://code.metoffice.gov.uk/trac/roses-u/browser/a/m/5/3/9

(to see a changeset, click on the red number (not the cog) in the **Rev** column → View changes). Equivalently, go to https://code.metoffice.gov.uk/rosie/u/, put “u-am539” in the search box, find the right suite and then click “../” once to go up to ‘trunk’.

Modifying a Rose Suite

It's important to remember that Rose suites are NOT version-independent and are also NOT platform-independent: if you have downloaded a suite suitable for JULESvn5.1 that ran on a Univ. Durham's computer, for example, you will need to modify it to have it run for JULESvn5.2 on a Univ. Exeter's computer (or JASMIN, or ARCHER, etc.) even if all the science settings are identical.

Apart from very tiny suites (e.g. my examples in this tutorial), suites also do NOT contain the driving and ancillary files they need to run (they just point to them), so if you want to use that suite from Durham you will also generally have to download these files separately (by which I mean email the suite authors and ask for copies).

Details of the modification step are on this page:

http://jules.jchmr.org/content/modify-suite

JULES from Scratch

We’re mostly through *JULES from Scratch* now. Let’s recap:

- We started with **installation**, starting at the page http://jules.jchmr.org/content/scratch, which got you set up with *Cylc*, *Rose* and *FCM* installed and also JULES itself (each has their own page describing the installation steps). Then we checked the installation with a quick run.

- **Rose suites** came next http://jules.jchmr.org/content/rose-suites, what they are and where to get them. These are the control files for a JULES run and details of the essential modification steps you need to go through to get a downloaded suite working on your system are here: http://jules.jchmr.org/content/modify-suite

See my ‘getting started’ page http://jules.jchmr.org/content/getting-started for the other files you need like driving data, ancillary files (and links to analysis tools are there too because you will need to plot out the results of your runs too).

- All that’s left is to go through **a worked example** ...

A grid run example

ERA-Interim

WFD → WFDEI

Here's a grid run using WATCH data that you should be able to follow if you can download the files from https://www.tobymarthews.com/jules-short-course-2016.html :

□ Download the ancil files, unzip and save them all in a directory called **ancils/** on your system.

□ Download the driving data files (all the other .zip files), unzip and put them all in a directory called **WFD-EI-Forcing/** (i.e. you should end up with several files each in their own directory like /.../WFD-EI-Forcing/Tair_WFDEI_land/Tair_WFDEI_land_200306.nc).

□ Check out suite u-am539 from Rosie Go and save in **~/roses/** on your system. That suite is set up for the CEH linux boxes, so you need to modify it: go carefully through all the 'modify suite' steps on http://jules.jchmr.org/content/modify-suite to get it working on your platform, including:
- Search-replace path /users/global/tobmar/MODELS/iofiles/io_wetlands/ancils/ to the absolute path of **ancils/** on your system.
- Search-replace path /prj/nceo/WFD-EI-Forcing/ to the absolute path of **WFD-EI-Forcing/** on your system.

□ Open u-am539 in Rose Edit and change the duration of the run (Timestepping namelist) to just the first 15 days of June 2003 (midnight is always at the start of the day, so this means main_run_end should be 2003-06-16 00:00:00).

□ Change the output directory (Output namelist) /users/global/tobmar/MODELS/iofiles/io_riv/output1 to a directory (of your choice) on your system (check it exists). Leave run_id as '$ROSE_TASK_NAME'

□ Click the Play button in Rose Edit and it should work (if it doesn't, please email me tobmar@ceh.ac.uk with a screenshot and I'll try to help out!).

Troubleshooting

For the following, I assume you've put your Rose Suite path in environment variable $RSUITE, e.g.:

```
wllf001 ~ $ export RSUITE=$HOME/roses/u-am539
wllf001 ~ $ echo $RSUITE
/users/global/tobmar/roses/u-am539
wllf001 ~ $
```

- While the suite is running you can right-click on jules → View → job stdout to check on progress (this file holds the `[INFO]` lines JULES produces during execution).
- If you close the Cylc GUI accidentally, type `cylc gscan &` and double-click on your Rose suite job to reopen it. In the same way, you can reopen a suite you've left running e.g. overnight.
- A new directory ~/cylc-run/ will appear on your system when the run starts (if not there already). When you run a rose suite, a copy of the whole suite is put in this directory at location $CSUITE defined like this ($CSUITE is what I call the suite's 'run directory' described on https://metomi.github.io/rose/doc/html/tutorial/rose/suites.html#suite-directory-vs-run-directory):

```
wllf001 ~ $ export CSUITE=$HOME/cylc-run/${RSUITE##*/}
wllf001 ~ $ echo $CSUITE
/users/global/tobmar/cylc-run/u-am539
wllf001 ~ $
```

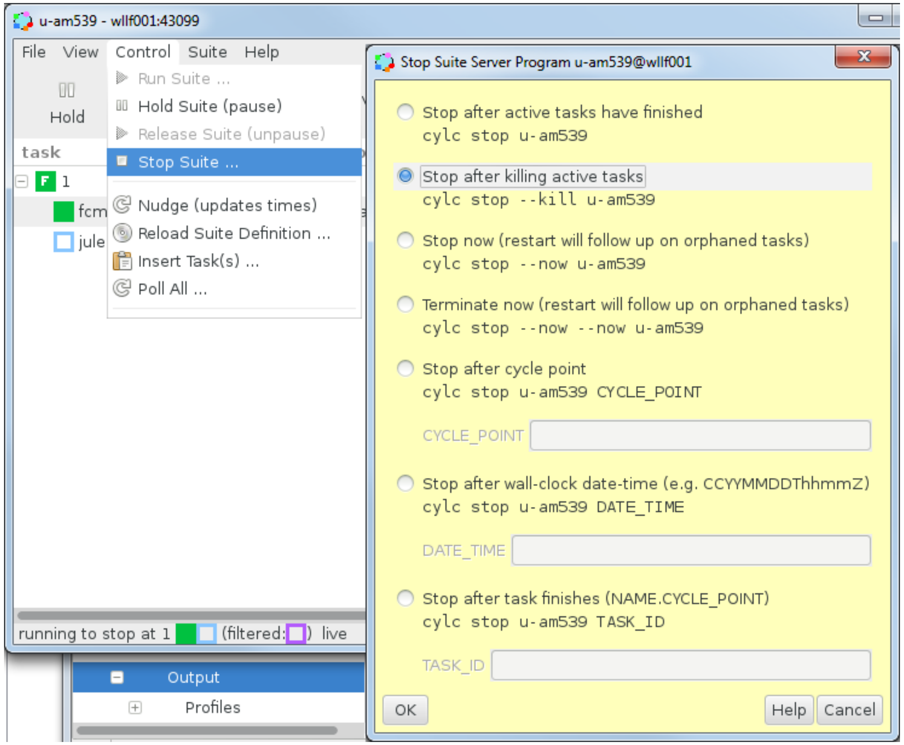
- Note that Rose suites run independently of the session you're in (Cylc suites run as daemons) so you gain nothing by opening three separate shells and initiating three runs in each rather than running three in the same session.
- If you find that the *fcm_make* step works, but it hangs when it moves on to the *jules* app, check that you have correctly added the JULES command set to $PATH (see http://jules.jchmr.org/content/get-jules).

Troubleshooting

- If you want to abort/stop a suite:

DON'T JUST CLOSE OFF THE CYLC GUI WINDOW (the one with the): the suite will still continue in the background and will prevent you starting another run of the same suite.

DON'T CLICK THE BIG STOP SUITE BUTTON: Surprisingly, this *doesn't* stop the suite **: Cylc will actually wait for all submitted / running jobs to complete (during which it says "stopping"), and only when they've completed will it stop the suite. I feel it really should be called a 'Complete submitted jobs & exit' button (but that's just my opinion)



Stop Suite Server Program u-am539@wllf001

○ Stop after active tasks have finished
`cylc stop u-am539`

● Stop after killing active tasks
`cylc stop --kill u-am539`

○ Stop now (restart will follow up on orphaned tasks)
`cylc stop --now u-am539`

○ Terminate now (restart will follow up on orphaned tasks)
`cylc stop --now --now u-am539`

○ Stop after cycle point
`cylc stop u-am539 CYCLE_POINT`

CYCLE_POINT

○ Stop after wall-clock date-time (e.g. CCYYMMDDThhmmZ)
`cylc stop u-am539 DATE_TIME`

DATE_TIME

○ Stop after task finishes (NAME.CYCLE_POINT)
`cylc stop u-am539 TASK_ID`

TASK_ID

OK Help Cancel

If you want to actually kill/stop your suite, you need to go into the menus like I'm showing in the screenshot right and choose 'Stop after killing active tasks', which will do the equivalent of the command:

```
cylc stop ${RSUITE##*/} --kill
```

If that still doesn't work, try NCAS's advice at http://cms.ncas.ac.uk/wiki/RoseCylc/Hints#Problemsshuttingdownsuites .

** In *Cylc* terminology, the suite is just the calling structure for a set of jobs, so this button does 'stop the suite' in the sense that no further jobs will be initiated, but it doesn't stop any jobs that have already been called (see http://metomi.github.io/rose/doc/html/cheat-sheet.html).

Viewing the logs

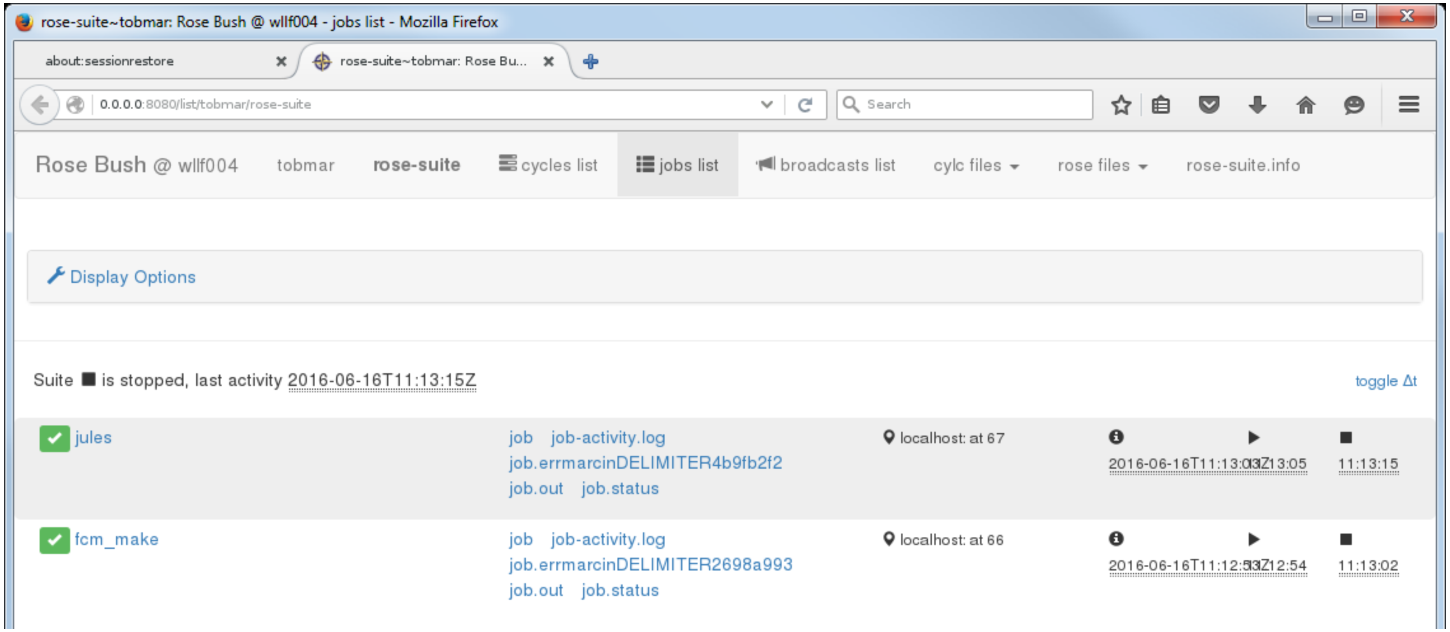
- Rather than printing progress information to the screen (`[INFO]` lines) and any execution errors, when JULES is run through Rose/Cylc these two are diverted into two textfiles called ‘log files’ stored in the directory ~/cylc-run/ . Progress information goes into a file called **job.out** and any errors go to a file called **job.err**, which you can open from the command line like this:

```
more $CSUITE/log/job/1/jules/NN/job.out
more $CSUITE/log/job/1/jules/NN/job.err
```

Or, as mentioned above, you can also open these two log files through the Cylc GUI by right-clicking on the 'jules   running' line → View → [job stdout or job stderr]

If your job succeeds, note that the Cylc GUI will always clear the job away and you won’t be able to open these files with the mouse any more, however, which I find a bit annoying (the files are still accessible from the command line, though).

Viewing the logs



rose-suite~tobmar: Rose Bush @ wllf004 - jobs list - Mozilla Firefox

about:sessionrestore | rose-suite~tobmar: Rose Bu...

0.0.0.0:8080/list/tobmar/rose-suite

Suite ■ is stopped, last activity 2016-06-16T11:13:15Z

✓ jules	job job-activity.log job.errmarcinDELIMITER4b9fb2f2 job.out job.status	localhost: at 67	2016-06-16T11:13:03Z13:05	11:13:15
✓ fcm_make	job job-activity.log job.errmarcinDELIMITER2698a993 job.out job.status	localhost: at 66	2016-06-16T11:12:53Z12:54	11:13:02

A more user-friendly way of viewing these logs that works on most systems is to use **Rose Bush**:

```
rose slv --name=${RSUITE##*/}
```

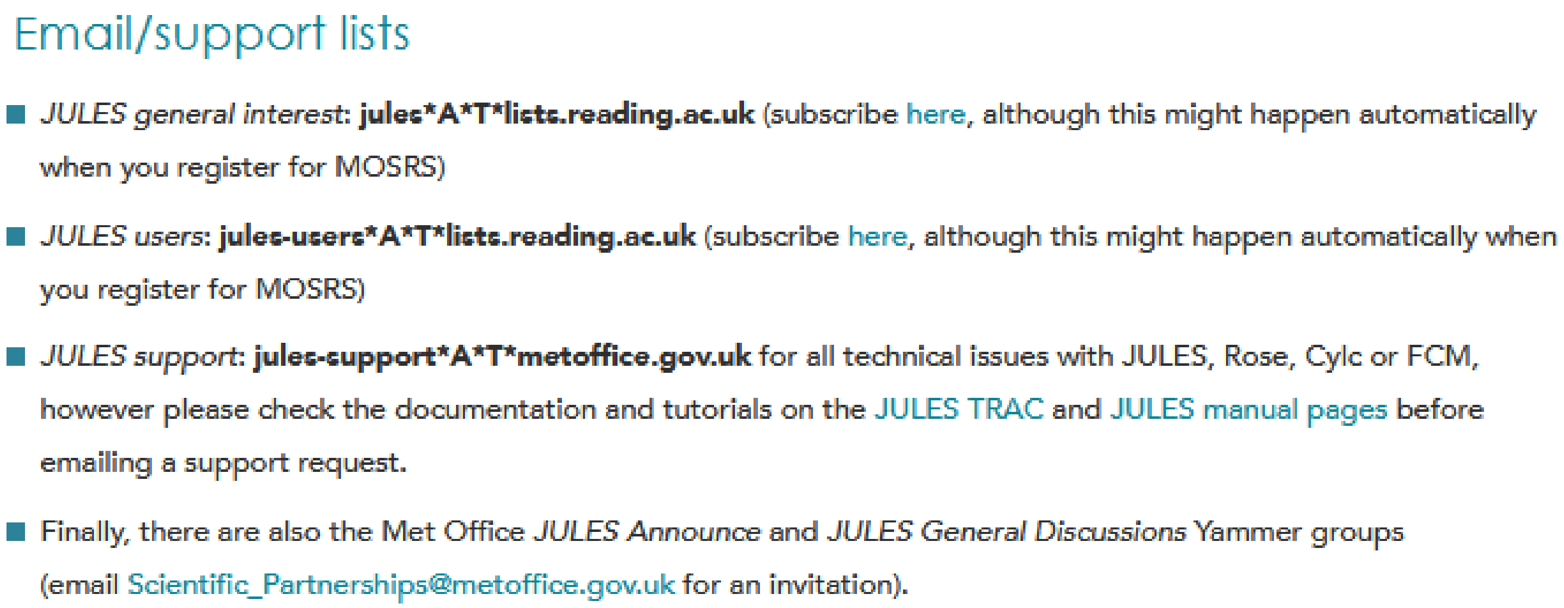
(n.b. on some systems this command can take 10-15 sec to open; on Monsoon, instead launch firefox on exvmsrose and go to http://localhost/rose-bush/ - see http://cms.ncas.ac.uk/wiki/RoseCylc/Hints#CantviewoutputinRosebush).

Rose Bush also has a ‘history’ feature and will show you the logs of previous runs. This is useful, but note that if you ever delete the ~/cylc-run/ directory from your system then you will wipe this history too (I sometimes clear ~/cylc-run/ because it’s really just a scratch directory and it sometimes gets very large).

Help resources

Finally, YOU ARE NOT ALONE in your labours and trials with JULES, Rose, Cylc and FCM ...

Please do use the support email lists/groups or, if you don't feel these are fit for your purposes (too high-brow, too technical, unfriendly, simply scary ...) then create your own and I'll add it to the list on http://jules.jchmr.org/content/getting-started#email-and-support-lists .



Email/support lists

- *JULES general interest*: **jules*A*T*lists.reading.ac.uk** (subscribe here, although this might happen automatically when you register for MOSRS)
- *JULES users*: **jules-users*A*T*lists.reading.ac.uk** (subscribe here, although this might happen automatically when you register for MOSRS)
- *JULES support*: **jules-support*A*T*metoffice.gov.uk** for all technical issues with JULES, Rose, Cylc or FCM, however please check the documentation and tutorials on the JULES TRAC and JULES manual pages before emailing a support request.
- Finally, there are also the Met Office *JULES Announce* and *JULES General Discussions* Yammer groups (email Scientific_Partnerships@metoffice.gov.uk for an invitation).

Finally, it would help the support team(s) a lot if you could always state what server system you are on. You can get this information from typing `cat /etc/os-release` on UNIX (e.g. In Sep 2018 I'm on Fedora 20).

www.TobyMarthews.com

Thank you very much, everyone!