Fundamentals on FCM, Rose/Cylc

University of Exeter

More info at:
http://jules-lsm.github.io/tutorial/bg_info/tutorial/fcm_fundamentals.html#

January 16th, 2019

Version Control System

- share work and encourage collaboration
- track progression and incremental changes to versions/revisions of work
- allow multiple flavours of some work to investigate the impact of change
- traceability of work → what version of the work was used and when
- to revert any change, branch from a version, merge in changes

http://jules-lsm.github.io/tutorial/bg_info/tutorial/fcm_fundamentals.html#

Subversion

- Incremental change tracking system
- Explore changes at each change and recall a version using its revision number.
- It takes snapshots of the directory tree vs time and records only incremental changes.
- A working copy (local copy) of a particular version of the repository is availiable for users to freely work on.
- Using copy-modify-merge allows each user of the repository to have their own personal working copy and they can work simultaneously and independently to modify their personal copies which can then be merged together later.
- Each change committed to the repository creates a new state of the filesystem tree (snapshot) → named revision number.
- The revision number is applied to the entire repository tree and not individual files and is therefore also known as the Global Revision Number. For example: at revision number N then a file may be the same as it was at revision number M but a different file was changed between the two revisions.
- simple interface for users

http://jules-lsm.github.io/tutorial/bg_info/tutorial/fcm_fundamentals.html#

FCM: based on and build on Subversion

- designed to be a helpful and simple interface for users to version control
- is a toolkit (wrapper) for Subversion and adds features to support the user experience of managing code and work easier
- simplifies and enhances working with Subversion to provide a user friendly interface to Subversion therefore allowing users to effectively version control there work
- can be used, if the site is setup, to link together Subversion and Trac. Trac is an environment by which projects can be hosted, by the site, and used to record information and tickets/issues
- is best used via the command line and is better not to interchange it with Subversion. FCM uses the site repositories, where the code can be managed in local repositories
- uses repositories for centrally storing data for a given project
- Configure: ∼/.subversion/config

http://jules-lsm.github.io/tutorial/bg_info/tutorial/fcm_fundamentals.html#

FCM: Some commands

- help: `fcm <command> help` → `fcm checkout --help`
- find out information of your working copy → `fcm info`
- finding out general differences from the repository → `fcm status|st`: !, ?, A, C, D, M, R
- retrieve code from a repository → `fcm checkout|co`
- commit code to the repository → `fcm commit|ci`
- delete files/directories → `fcm delete|del|remove|rm`
- create a branch → `fcm branch-create|bc|bcreate`
- for line-by-line or file-by-file differences from repository → `fcm diff|di --graphical, -g`
- difference of branches → `fcm branch-diff|bdiff|bdi`
- update your working copy to the one on the trunk → `fcm trunk update|up`
- move to a branch, after creating it → `fcm switch|sw`
- pull changes from one area in repository into another → `fcm merge`
- delete a branch → `fcm branch-delete|bdelete|bdel`

http://jules-lsm.github.io/tutorial/bg_info/tutorial/fcm_fundamentals.html#

JULES: Code

- JULES code from repository:

```
  fcm checkout fcm:jules.x_tr@vn5.3 directory-name
```

```
fcm co
       https://code.metoffice.gov.uk/svn/jules/main/trunk@vn5.3
```

```
  ls directory-name
       admin benchmark bin etc ModuleLeaders.txt
       rose-meta rose-stem src utils
```

http://jules-lsm.github.io/tutorial/bg_info/tutorial/fcm_fundamentals.html#

FCM: Some commands

- help: `fcm <command> help` → `fcm checkout --help`
- find out information of your working copy → `fcm info`
- finding out general differences from the repository → `fcm status|st`: !, ?, A, C, D, M, R
- retrieve code from a repository → `fcm checkout|co`
- commit code to the repository → `fcm commit|ci`
- delete files/directories → `fcm delete|del|remove|rm`
- create a branch → `fcm branch-create|bc|bcreate`
- for line-by-line or file-by-file differences from repository → `fcm diff|di --graphical, -g`
- difference of branches → `fcm branch-diff|bdiff|bdi`
- update your working copy to the one on the trunk → `fcm trunk update|up`
- move to a branch, after creating it → `fcm switch|sw`
- pull changes from one area in repository into another → `fcm merge`
- delete a branch → `fcm branch-delete|bdelete|bdel`

http://metomi.github.io/fcm/doc/user_guide/

FCM: Loading suite

```
fcm co
        https://code.metoffice.gov.uk/svn/roses-u/b/e/0/8/2/trunk/
        u-be082
```

```
ls u-be082
        app  bin  meta  rose-suite.conf rose-suite.info
        suite.rc
```

`ls -R u-be082` → more info of each sub-dir.

http://metomi.github.io/fcm/doc/user_guide/

Cylc commands

- Cylc: `cylc <COMMAND> --help`

  Graphical interface to monitor suites → `cylc gscan`

  The GUI → `cylc gui` or `gcylc`

  Monitor suites on the terminal → `cylc scan -c`

  `cylc view -g -j <SUITE_ID>`

  To follow progress of a suite without a GUI →

  ```
  tail -f  /cylc-run/SUITE_ID/log/suite/log
  ```

https://metomi.github.io/rose/doc/html/tutorial/cylc/index.html

Rose

help → `rose <COMMAND> --help`
To see the repositry → `rosie go`
Make a copy of a suite (NewSuite_ID) → `rosie copy`
Copy a suite → `rosie checkout`
Delete a suite → `rosie delete`
Edit a suite → `rose edit`
GUI of a suite → `rose sgc`
Run suite → `rose suite-run`
Run suite → `rose suite-run -new`
Run suite using new name → `rose suite-run -new -name=<NEW_NAME>`
Stop suite → `rose suite-shutdown`

```
rose suite-shutdown -name=<SUITE_NAME> -y -- --now
```

→ forces the suite to shutdown without waiting
Test when changing source code → `rose stem -new -S HOUSEKEEPING=false -source=.`

https://metomi.github.io/rose/doc/html/index.html

Durectory: cylc-run

- a copy of the suite
- place for storing namelist, output, plots

```
ls  /cylc-run/SUITE_ID/
app  bin  cylc-suite.db  log  log.20181214T132953Z
meta  rose-suite.info  share  suite.rc
suite.rc.processed  work
```

`ls -R  /cylc-run/u-be082` → more info of each sub-dir.