

Introducing git fleximod

Extending the power of git

Jim Edwards,
NSF-NCAR CGD software engineer

CESM WORKSHOP June 11, 2024



Outline

1. What's wrong with `manage_externals` and why do we need to replace it?
2. Why can't we just use git?
3. Adding to the `.gitmodules` file
4. Sparse checkout feature
5. Backward compatibility

Why replace `manage_externals`?

- `Manage_externals` was created to support both git and subversion repositories
- It used the subversion API to support sparse checkouts (even from git repositories)
- GITHUB ended support for the subversion API in January 2024
- With all of our component repositories now in git we can use more

Git-fleximod command line

usage: git-fleximod [-h] [-C PATH] [-g [GITMODULES]] [-x [EXCLUDE ...]] [-f] [-o] [-v] [-V] [--backtrace] [-d]
 {update,status,test} [components ...]

git-fleximod manages checking out groups of gitsubmodules with additional support for Earth System Models

positional arguments:

{update,status,test} Subcommand of git-fleximod, choices are ['update', 'status']

components Specific component(s) to checkout. By default, all required submodules are checked out.

optional arguments:

-h, --help show this help message and exit

-C PATH, --path PATH Toplevel repository directory. Defaults to top git directory relative to current.

-g [GITMODULES], --gitmodules [GITMODULES]

The submodule description filename. Default: .gitmodules.

-x [EXCLUDE ...], --exclude [EXCLUDE ...]

Component(s) listed in the gitmodules file which should be ignored.

-f, --force Override cautions and update or checkout over locally modified repository.

-o, --optional By default only the required submodules are checked out. This flag will also checkout the optional submodules relative to the toplevel directory.

-v, --verbose Output additional information to the screen and log file. This flag can be used up to two times, increasing the verbosity level each time.

-V, --version Print version and exit.

--backtrace DEVELOPER: show exception backtraces as extra debugging output

-d, --debug DEVELOPER: output additional debugging information to the screen and log file.

Why not just use git?

We want CESM components as components

Components want independence

Submodules of CESM:

path = ccs_config

path = share

path = cime

path = libraries/mct

path = libraries/mpi-serial

path = components/cam

path = components/ww3

path = components/rtm

path = tools/statistical_ensemble_test/pyCECT

path = components/mosart

path = components/mizuroute

path = libraries/FMS

path = libraries/parallel-io

path = components/cdeps

path = components/cmeps

path = components/cice

path = components/cism

path = components/clm

path = components/mom

Submodules of CTSM

path = src/fates

path = components/cism

path = components/rtm

path = components/mosart

path = components/mizuRoute

path = ccs_config

path = cime

path = components/cmeps

path = components/cdeps

path = share

path = libraries/mct

path = libraries/parallelio

path = doc/doc-builder

The .gitmodules file

```
[submodule "clubb"]  
  path = src/physics/clubb  
  url = https://github.com/larson-group/clubb_release  
  fxrequired = AlwaysRequired  
  fxsparse = ../clubb_sparse_checkout  
  fxtag = clubb_4ncar_20231115_5406350  
  fxDONOTUSEurl = https://github.com/larson-group/clubb_release
```

```
[submodule "cism"]  
  path = components/cism  
  url = https://github.com/ESCOMP/CISM-wrapper  
  fxtag = cismwrap_2_2_001  
  fxrequired = ToplevelRequired  
  fxDONOTUSEurl = https://github.com/ESCOMP/CISM-wrapper
```

Backward compatibility

Suppose you want to update one or more CESM components but you aren't ready to update others and want to continue to use `manage_externals`.

Edit the Externals file from the previous version of your component and replace the `repo_url` key word with `from_submodule=True`

```
[ali_arms]
local_path = src/physics/ali_arms
protocol = git
repo_url =
https://github.com/ESCOMP/ALI-ARMS
tag = ALI_ARMS_v1.0.1
required = True
```

```
[ali_arms]
local_path = src/physics/ali_arms
protocol = git
required = True
from_submodule = True
```


