Creating a CLM Tag

Quick Guide to Creating a CLM tag for the CLM Code Management Team (CLM-CMT)

Erik Kluzek

Reference: Online SVN Handbook (http://svnbook.red-bean.com/nightly/en/index.html)
*Help on Subversion for CCSM: (http://bb.cgd.ucar.edu/showthread.php?t=254)

This is how to create a main CLM trunk tag. For working on a CLM branch see ?Using SVN to Work with CLM Development Branches

Step 1. Check out CLM trunk;

```
% setenv REPO https://svn-ccsm-models.cgd.ucar.edu/
% svn co $REPO/clm2/trunk ./clm_trunk
```

Step 2. Copy source modifications into mods your svn sandbox with the latest trunk tag.

Note: If you are changing a CLM parameter or a CLM surface dataset you must also change the XML files that create the CLM namelist (bld/namelist_files /namelist_defaults_clm.xml). See CLM User's Guide about this...

Adding files to the CLM XML database

Step 2.a. Move your changes into the sandbox

Move your source changes directly into your sandbox into the relevant source code directories. If you have them in a case SourceMods directory you need to copy them into "models/lnd/clm/src/*" first so that you can check them into SVN. If you are on a branch you first update your branch to the latest trunk version, and then apply the "merge" command in the trunk sandbox to move the changes on your branch onto the trunk (do merge between the trunk version and the last tag of your branch for that same trunk version).

Step 2.b. Check your changes:

Use the status and difference commands to make sure you have all

```
% svn status
% svn diff
```

If you have a new file use the add command to set it up so that it will be added to the repository in your next commit.

```
% svn add <filename>
```

Likewise if you need to delete an old file use the "rm" command to remove it on the next commit.

```
% svn rm <filename>
```

Step 3. Run test suite to look for problems with code changes.

The CLM User's Guide talks about the different tools for testing CLM...

The CLM User's Guide has a section about using the tools..

Tools for testing CLM

The CLM Testing page gives more information on using the test suite and the requirements for testing for tags.

Step 4. Record changes in the ChangeLog using the UpDateChangeLog script

Record information about your changes in the CLM ChangeLog. The ChangeLog appears in both the top level directory and the models/lnd/clm/doc directory, and the summary information is included in the ChangeSum file. To make this easier to edit and manage use the UpDateChangeLog.pl script that is in the top level of the clm trunk.

```
% UpDateChangeLog.pl <new_tag_name> "<one line description of changes>"
```

Modify ChangeLog including:

- Main purpose of tag (usually copy this over from the description in the CLM Upcoming tags page.
- Information (and bug-numbers) on bugs fixed.
- Information on changes including: changes to build-system, namelist, and files in the XML database.
- Test suite results and information
- Any externals updated (such as csm_share etc.).
- If the tag changes answers or if answers are bit-for-bit with the previous tag
- References to any diagnostic plots that record changes (especially if changes result in climate-changes).

If you need to update the date/time for your tag:

```
% UpDateChangeLog.pl -update
```

The UpDateChangeLog.pl script will include information on yourself and the
time of your commit to modify the ChangeLog/ChangeSum and include a template description of the change at the top of the ChangeLog file. It also copies these files down into models/lnd/clm/doc/ which helps to document CLM for CCSM4 tags.

Test level:

Type of testing done, and the standard process required for different types of tags (from least testing to greatest).

- doc: No testing, just documenting changes on major version
- critical: ONLY run CESM standard tests+whats needed to test feature added (only yellowstone)
- standard: Testing somewhere between critical and std-test
- std-test: Standard testing, run: test_system on: yellowstone (batch), frankfurt (interactive) and CESM clm.auxtest testlist, build-namelist unit tester, allIcompsets testlist
- reg-test: Regression testing, run all tests on all machines (annually), "test_driver.sh -i" on yellowstone (or if tools change)

Test review: annually review testing to make sure we are testing features that are needed. If a feature isn't needed anymore, we should at minimum remove the test, but preferably remove the code for the feature.

Externals updates:

It's recommend that externals that are updated in CLM tags should in general be the same as the latest CESM beta tag, because they will be the most well tested. But, sometimes later tags may be required for features that are needed in a new tag.

Step 5. Commit changes to svn repository.

Review your changes, make sure everything looks good and everything is added or removed as needed and then commit your changes to the trunk.

```
% svn status ! Check to see what files are effected
             ! make sure all files are added/removed
% svn diff   ! Review your changes
% svn commit -m "My commit message"
```

Step 6. Tag changes

Create a tag of your changes so that it's easy to get back to this exact version.

```
% svn copy $REPO/trunk $REPO/trunk_tags/clm<tag_name> -m "My Tag message"
```