

Adding Variables to the Restart File

- [Introduction – why do variables need to be added to the restart file?](#)
- [How do you add a variable to the restart file?](#)
- [How do you figure out what variables need to go on the restart file?](#)
- [How do you test that you have exact restart?](#)

Introduction – why do variables need to be added to the restart file?

Any new variables that are added to CLM need to be evaluated if they need to go on the restart file. The idea with the restart file is to exactly preserve the model state between run submissions so that you can't tell the difference between a continuous simulation and one that starts up from a restart file.

So when you add new variables -- what type of variables have to go on the restart file?

Prognostic state variables where the state is updated from timestep to timestep nearly always need to go on the restart file. For these variables, an additional issue that developers need to think about is what an appropriate value should be for a cold-start (i.e. a simulation that is started without a restart or initialization file).

Fluxes that are calculated each time-step rarely have to go on the restart file.

Variables that are only local to one subroutine and parameters or data read in from input files typically do NOT need to go on the restart file.

These rules are not steadfast. Sometimes additional variables are required in order to preserve answers on the history files. Sometimes, this has to do with sequencing issues where maybe something is calculated after being saved to the restart file, so it doesn't have the correct value for the next timestep on the restart file.

For some general information on the restart mechanism for CESM see http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/usersguide/x1582.html#running_ccsm_restarts

How do you add a variable to the restart file?

To add a variable to the restart file, you add an additional call to the "Restart" method of the type that contains the new variable. It looks something like this...

```
call restartvar(ncid=ncid, flag=flag, varname='T_SOISNO', xtype=ncd_double, &
               dim1name='column', dim2name='levtot', switchdim=.true., &
               long_name='soil-snow temperature', units='K', &
               interpinic_flag='interp', readvar=readvar, data=this%t_soisno_col)
```

If it's a module level variable that is saved within a module the call is added the same as above, to something that is called at initialization. The data pointer that is pointed to (with the data keyword) needs to be persistent and available at all times. As we already said an array that is only local to a given subroutine would NOT be able to be added to the restart file in this way.

How do you figure out what variables need to go on the restart file?

Now, the question is – how do you figure out exactly which variables need to go on the restart file and which ones don't? There are two approaches: an analytical approach and a trial and error approach. Often solving restart problems requires doing some of both.=

Analytic approach: consider each new variable and decide if there is a reason it should be on the restart file. Add these variables to the restart file. Then complete restart tests and see if they pass. If the restart tests do not pass, then reconsider the variables that have NOT been added and decide if there is a reason one or more might actually need to be added.

Trial and error method: Add the new variables to the restart files and then assess if it was needed or not. For example, add all the new variables, make sure restarts work correctly, and then gradually eliminate variables which are not required out through successive tests. TYou can do this in groups, or some other method to go through it methodically one variable at a time.

In either case you also want to make sure you are outputting any new fields added to the history file, to make sure those new variables are preserved correctly on the history file.

How do you test that you have exact restart?

To test by hand you run a simulation for some period, say 10 days, and then run a simulation with a restart part way through (say at day 5) and then you make sure the global averages of the fields passed in the coupler as well as the cpl and clm history files for the simulation with restart is identical to the one that ran continuously. So you compare the results of the days 6-10 from the restart simulation to the one that ran continuously.

Doing this by hand is a bit tricky but we have tests setup to make testing restarts easy.

Restart test procedure:

```
cd cime/scripts
```

```
./create_test -testname ERS.fl19_g16.ICLM45BGC.yellowstone_intel.clm-default
```

This creates a test case for you with the testname and some numbers to give the date and time on the end. You "cd" to that directory, and then run the \$CASE.test_build script to build the model and run the \$CASE.submit script to submit the tests.

The above test is an Exact-Restart test from Startup. There are other types of ER tests as well such as ERI which tests from startup, hybrid and branch as well.

NOTE: the compset modifications in 'clm-default' are required to test clm restart files. Without these modifications, only the coupler files are tested.

For more information on automated testing see [CLM Testing](#). See the list of tests at <http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/usersguide/x2731.html>