

CCSM Component Logging and Unit Handling

CCSM I/O logging behavior and unit handling

- all component models will be required to replace any writes to unit 6 with calls to writes to a generic integer
 - component models will be responsible for individually handling their non-masterproc info. Components will be encouraged to write non-masterproc info either to separate files or have processor id's prepended to the log output (this latter will probably be more optimal in the case where the component is run on potentially thousands of processors)
- unit 6 will still be reserved as a default stdout unit number
 - this will currently be used for libraries such as mct, pio, ...)
 - this will currently be used for share code output, until a cleaner mechanism is implemented (see below)
- components will be encouraged to use `shr_file_getUnit` and `shr_file_freeUnit` for files other than log files, however each component would be free to implement their own compatible version of the unit manager
 - this is particularly important in the implementation of a sequential system
 - log files have their own special issues related to naming convention and `shr_file_chStdOut` and as a result the implementation of unit numbers for log files has been deferred
 - `shr_file_getUnit` and `shr_file_freeUnit` currently works and will continue to work on a pe-basis
- **Issues with current log file implementations**
- concurrent system
 - in each component directory model/ where model = [atm, ice, ocn, lnd, cpl], there is a file, `model_stdio.nml`, that contains the namelist, [namelist / stdio / dir, stdin, stdout, NLFIL], where stdout is the name of the file that is used to redirect unit 6 to and is assumed to be in the directory dir (where model input and output is located)
 - all components currently call `shr_file_chstdout` to redirect unit 6 to this named file.
- sequential system
 - how do we want to determine the redirection of unit 6 for every component?
 - Would it be reasonable to implement a mechanism similar to the concurrent system, where every component reads in a stdio namelist file and then invokes the a call to `shr_file_chstdout`?
 - Alternatively, the module `shr_input_info_mod.F9` determines a unique log file postfix. This could be utilized by every component model in the sequential system to construct its own log file to redirect from.
 - Do we want to unify the redirection of log files in the concurrent and sequential system?
 - How do we want to treat share code log output? Tony has suggested the following.
 - no output should be default
 - add a `shr_file_logunit` variable to `shr_file_mod` and set this to 6. slowly migrate all share code to use this variable instead of "6" (just like the components).
 - add a `shr_file_debug` variable to `shr_file_mod` and set this to 0. slowly migrate all share code to trigger write statements off this variable.