

Stage-1-11

Cleanup time-managers and move to clocks

Ideal engineering time: 14 days

Estimate of code lines: 5k, 500 scripts

Actual: 11.7k CAM, 9k csm_share, 1.5k clm, 1k esmf_wrf_timemgr, .1 scripts

Actual total: 23.5k

16 weeks

150lpd

2 weeks vacation, Breckenridge and ESMF meeting

- Use chghilevtimemgr_clm3_expa_61 branch for CLM (changes lnd_comp_mct.F90, drivers program_csm.F90, program_off.F90 stays same, time_manager.F90 stays same).
- Pass in sync_clock from top level to atm, lnd, ice, ocn components
- Get rid of nlend, and rstwr except in ???_comp.F90 files. ???_comp_mct.F90 set these based on the SyncClock passed into them.
- Leave comctl and consol as it is.
- Pass in SyncClock and CCSMInit to readnamelist and use them to set the relevant values.
- Remove alarm in restart use SyncClock and restart alarm methods.
- Leave clock on restart file – but on restart compare with SyncClock and make sure times are in sync.
- Use Sync clock to get start and ref times.
- Remove ic_ymd and ic_tod – not needed anymore.
- Remove order of dependence of CAM and CLM namelists
- Change cam namelist to cam_inparm and CLM to clm_inparm
- Change CCSM scripts for namelist change.
- Remove get_clock and get_curr_ESMF_Time methods from time_manager – not needed.
- Remove public attribute and ranting and raving from tm_clock declaration in time_manager.
- Remove tm_aqua_planet from time_manager, ccsminparm sets this by setting perpetual_run and perpetual_ymd appropriately at the top level.
- Remove input namelist (other than dtim) from time_manager.
- timemgr_init redone to be based off of input sync_clock set at top level.
- Brian E. will move calculation of dtim based on resolution from build-namelist to inside of dynamics code.
- Set top level driver information from seq_ccsm namelist pass to sub-components.
- ccsminparm:
 - case_desc
 - case_name
 - start_type
 - archive_dir
 - mss_irt
 - mss_wpass
 - brnch_retain_casename
- timemgr_inparm:
 - calendar
 - stop_noption
 - stop_n
 - stop_tod
 - stop_ymd
 - stop_tod
 - stop_final_ymd
 - restart_noption
 - restart_n
 - start_ymd ! MUST BE SET!
 - start_tod ! Defaults to zero
 - ref_ymd
 - ref_tod
 - perpetual_run
 - perpetual_ymd
 - atm_cpl_dt ! MUST BE SET! coupling frequency of atm to surface AND for now must equal camexp dtim
 - orb_obliq
 - orb_eccen
 - orb_mvlp
 - orb_iyear_AD
 - atm_ideal_phys
 - atm_adiabatic
 - aqua_planet ! If aqua_planet is set, perpetual_ymd is also set to 3/21
- Change build_namelist to also create ccsminparm namelist
- Add initial method to shr_timemgr_mod to create sync_clock and to create a new clock from the sync_clock with a different dtim.
- Initialization method in time_manager creates it's internal tm_clock based on the input sync_clock with it's own dtim.

Definition part of seq_ccsm_mct.F90.

Top clock/InitInfo level initialization...

```

!
! Initialize Initialization Information and clock setup objects
!
call shr_inputInfo_setDefaults( CCSMInit )
call shr_inputInfo_readNL( NLFileName, LogPrint=masterproc, MPICom=mpicom, &
                           MasterTask=masterproc, CCSMInitOut=CCSMInit )
! Get Perpetual run mode (in case aqua_planet mode is set)
call shr_inputInfo_get( CCSMInit, perpetual_run=perpetual_run, perpetual_ynd=perpetual_ynd )
call shr_timeMgr_setDefaults( ClockSetup, perpetual_run, perpetual_ynd )
call shr_timeMgr_readNL( NLFileName, LogPrint=masterproc, MPICom=mpicom, &
                        MasterTask=masterproc, SetupOut=ClockSetup )

! If a startup type
if ( .not. shr_inputInfo_IsRestart( CCSMInit ) )then
  ! Setup the clock and check that the ClockSetup object is valid
  call shr_timeMgr_setupClock( ClockSetup, LogPrint=masterproc, &
                              desc="Sequential CCSM with MCT Master Clock", &
                              ClockOut=SyncClock )
! Else if a continue or branch type
else
  ! Read in restart file
  call shr_inputInfo_readRPointer( MPICom, masterproc, CCSMInit, rest_file )
  call shr_inputInfo_readRestart( rest_file, MPICom=mpicom, LogPrint=masterproc, &
                                  MasterTask=masterproc, CCSMInitOut=CCSMInit )
end if

```

Component Initialization calls..

```

!
! Figure out which components will run
!
atm_present = shr_inputInfo_runModel( CCSMInit, 'atm' )
lnd_present = shr_inputInfo_runModel( CCSMInit, 'lnd' )
ocn_present = shr_inputInfo_runModel( CCSMInit, 'ocn' )
ice_present = shr_inputInfo_runModel( CCSMInit, 'ice' )
!
! Initialization of phase 1 of cam and surface components
!
if ( atm_present ) call atm_init1_mct( gsMap_atm, a2x_a, x2a_a, CCSMInit, SyncClock )
if ( lnd_present ) call lnd_init_mct ( gsMap_lnd, x2l_l, l2x_l, CCSMInit, SyncClock )
if ( ocn_present ) call ocn_init_mct ( gsMap_ocn, x2o_o, o2x_o, CCSMInit, SyncClock )
if ( ice_present ) call ice_init_mct ( gsMap_ice, x2i_i, i2x_i, CCSMInit, SyncClock )

```

Atm initialization

```

subroutine atm_init1( shr_ccsm_init, atm_in, &
                    atm_out, CCSMInit, syncClock )
  type(shr_inputInfo_InitType), intent(IN) :: CCSMInit
  type(shr_timeMgr_clockType), intent(IN) :: sync_clock
  call timemgr_init( syncClock )
.
.
.
  module time_manager
    type(shr_timeMgr_clockType) :: tm_clock ! Local instance of clock for CAM
    subroutine timemgr_init( sync_clock )
      tm_clock = shr_timeMgr_SetupFromClock( dtime, sync_clock )
    .
    .
  end module

```

con_cam.F90

```

call ccsm_seq_timer_init()
call shr_msg_stdio('atm')
call cpl_interface_init(cpl_fields_atmname,mpicom)
call spmdinit ()
call ccsm_seq_printlogheader()      ! Print Model heading and copyright message
call ccsmini1( sync_clock ) ! Get orbital information from coupler to put in sync_clock
call shr_ccsm_input_nl( unit=5, ccsm_initdata, sync_clock, &
                        log_print=masterproc )
call atm_init(          ccsm_initdata, atm_in, atm_out, sync_clock)
call ccsmini2(          ccsm_initdata, atm_in,          sync_clock )
do while ( .not. shr_timemgr_is_last_step( sync_clock ) )
  call atm_run1(        atm_in, atm_out )
  call ccsm_run(        atm_in, atm_out, sync_clock )
  call atm_run2(        atm_in, atm_out )
  call atm_run3(        atm_out )
  call atm_run4(        atm_in, atm_out , sync_clock)
  call shr_timemgr_advance_timestep( sync_clock )
end do
call atm_final(         atm_in, atm_out )
call ccsmfin(          )
call cpl_interface_finalize(cpl_fields_atmname)

```

Example input namelist:

```

&ccsm_inparm
  case_name          = 'camrun'
  start_type         = "initial"
/
&timemgr_inparm
  atm_cpl_dt         = 1800
  iyear_ad           = 1950
  restart_nmonths    = 1
  start_type         = "initial"
  start_ymd          = 901
  stop_ndays         = 1
/
&cam_inparm
  absems_data        = '/fs/cgd/csm/inputdata/atm/cam/rad/abs_ems_factors_fastvx.c030508.nc'
  aeroptics          = '/fs/cgd/csm/inputdata/atm/cam/rad/AerosolOptics_c050419.nc'
  bnd_topo           = '/fs/cgd/csm/inputdata/atm/cam/topo/USGS-gtopo30_32x64_c050520.nc'
  bndtvaer           = '/fs/cgd/csm/inputdata/atm/cam/rad/AerosolMass_V_32x64_clim_c031022.nc'
  bndtvo             = '/fs/cgd/csm/inputdata/atm/cam/ozone/pcmdio3.r8.64x1_L60_clim_c970515.nc'
  bndtvs             = '/fs/cgd/csm/inputdata/atm/cam/sst/sst_HadOIBl_bc_32x64_clim_c030228.nc'
  ncdata             = '/fs/cgd/csm/inputdata/atm/cam/inic/haus/cami_0000-09-01_32x64_L26_c030918.nc'
/
&clm_inparm
  fpftcon            = '/fs/cgd/csm/inputdata/lnd/clm2/pftdata/pft-physiology-cn16.c040719'
  fsurdat            = '/fs/cgd/csm/inputdata/lnd/clm2/srfddata/cam/clms_3.1_32x64_c050523.nc'
/

```

CCSM input namelist:

```

&ccsm_inparm
  case_name           = 'csmrun'
  start_type          = "continue"
  restart_pfile       = "./ccsm.csmrun.rpointer"
/
&timemgr_inparm
  atm_cpl_dt          = 3600
  start_ymd           = 901
  stop_nyears         = 9999
/
&cam_inparm
  absems_data         = '/fs/cgd/csm/inputdata/atm/cam/rad/abs_ems_factors_fastvx.c030508.nc'
  aeroptics           = '/fs/cgd/csm/inputdata/atm/cam/rad/AerosolOptics_c050419.nc'
  bnd_topo            = '/fs/cgd/csm/inputdata/atm/cam/topo/USGS-gtopo30_32x64_c050520.nc'
  bndtvaer            = '/fs/cgd/csm/inputdata/atm/cam/rad/AerosolMass_V_32x64_clim_c031022.nc'
  bndtvo              = '/fs/cgd/csm/inputdata/atm/cam/ozone/pcmdio3.r8.64x1_L60_clim_c970515.nc'
  bndtvs              = '/fs/cgd/csm/inputdata/atm/cam/sst/sst_HadOIBl_bc_32x64_clim_c030228.nc'
  dtime               = 1800
  ncdata              = '/fs/cgd/csm/inputdata/atm/cam/inic/gaus/cami_0000-09-01_32x64_L26_c030918.nc'
/

```

Note:

In the creation of the cam namelist ncdata takes into account what the start date is set to. If dates don't agree – then it may not be able to create the needed input datasets.