J. Input Parameters for the Flux Coupler

The Flux Coupler uses the FORTRAN namelist input parameter processing capability available in most FORTRAN compilers as an extension to FORTRAN 77. This is also a standard feature of FORTRAN 90. The coupler reads this input namelist, called drv.parm, from stdin at runtime. The following list shows all available namelist input parameters. The list shows the variable names of the input parameter variable name as it appears in both the input namelist and the model code. Recall that FORTRAN is not case sensitive with respect to variable names, but it is case sensitive with regard to character strings.

Flux Coupler input variables follow a naming convention in which the first few letters of the variable identify a basic functionality group:

- **rest_xxxx** selects options with respect to model initial conditions and the acquisition and creation of restart files,
- **stop_xxxx** selects options with respect to when a simulation will stop,
- **hist_xxxx** selects options with respect to when and what type of history data is created,
- **mss_xxxx** selects options with respect to data file storage on NCAR's Mass Storage System,
- **diag_xxxx** selects options with respect to run–time diagnostics of the physical simulation,
- **init_xxxx** selects options with respect to special model initialization procedures,
- **msg_xxxx** selects options with respect to the message passing environment,
- **info_xxxx** selects options with respect to monitoring the computational performance of the model.

A list of the Flux Coupler Input Namelist variables follows which includes:

- the default value of the variable. While reasonable default values are selected for all variables, users might want to alter these values according to their application.
- If and when the variable is required input. Note that very few input parameters are required. In a few cases, changing one default value will cause another input parameter to become required.
- a brief description of the purpose and effect of the parameter.
- see an example input namelist which uses the variable (examples follow the list).

rest_type

Default: "initial"
Required: no (but default is of limited usefulness)
Description: This selects the run type. Valid choices are: "initial", "continue", "branch", "regenerate." Selecting "branch" or "regenerate" makes rest_bfile a required input.
See Example: 1, 2, 3, 4

rest_dir

Default: "/invalid/rest_dir/"
Required: yes, always
Description: This is a mass store directory name. Restart and history files go into this directory. For continuation or regeneration runs, the initial condition restart file must also be in this directory.
See Example: 1, 2, 3, 4

rest_bfile

Default: "/invalid/rest_bfile"
Required: no (required if rest_type = "branch" or "regeneration", ignored otherwise)
Description: This is the mass store file name of initial condition restart data.
See Example: 3, 4

rest_date

Default: 00000830 (30 Aug, year 0)
Required: no (optional for initial and branch runs, ignored otherwise)
Description: This is the restart date (encoded: yyyymmdd).

- On initial runs, rest_date is the initial date of the simulation.
- On branch runs, if rest_date > 0, it will over-ride the date contained in the rest_bfile IC restart file. Otherwise the date from the rest_bfile data file is used.
- On continuation or regeneration runs rest_date is not used.

See Example: 1

rest_str

Default: "< no rest_str >"
Required: no, but highly recommended
Description: This is a short text string, up to 80 characters, which is included in output files to help identify the model run.
See Example: 1, 2, 3, 4

rest_freq

Default: "monthly"
Required: no
Description: This is the restart frequency, one of "monthly" or "ndays" which selects how often restart data files are created.
See Example: 2, 4

rest_nday

Default: 14
Required: no (only used if rest_freq = "ndays")
Description: If rest_freq = "ndays", restart files will be created every n days. More specifically, whenever $mod(d-p,n)=0$, where
d=the simulation day (1 Jan, year 1 <=> $d=365$)
p=an offset such that $mod(d-p,n) = 0$ on the date specified by rest_odate
n=rest_nday

See Example: 4

rest_odate

Default: 0
Required: no
Description: Selects restart file offset calendar date, used in conjunction with rest_nday when rest_freq = "ndays." If rest_odate > 0, it is a valid calendar date (encoded: yyyymmdd). Restart data is created every n days relative to (and including) either

- the initial date of the run (if rest_odate $\leq$ 0)
- the date hist_odate (if rest_odate > 0)

See Example: 4

stop_option

Default: "newyear"
Required: no
Description: Selects when to stop the simulation.

- "newyear" => stop on the next January 1st
- "newdecade" => stop on the next decade
- "ndays" => stop after advancing n days where n=stop_nday
- "date" => stop when stop_date is reached This makes stop_date a required input.

See Example: 1, 2, 3, 4

stop_nday

Default: 1
Required: no
Description: If stop_option = "ndays," stop after stop_nday days.
See Example: 3

stop_date

Default: 00000101 (1 Jan, year 0)
Required: no (required if stop_option = "date," ignored otherwise)
Description: If stop_option = "date," stop on this date.
See Example: 4

hist_freq

Default: "monthly"
Required: no
Description: One of "monthly," "biweekly," "weekly," or "ndays," which selects how often history data files are created.

- "monthly" => data is created on the 1st of each month
- "biweekly" => data is created on the 1st and 15th of each month
- "weekly" => data is created on the 1st, 8th, 15th, and 22nd of each month
- "ndays" => history data is created every n days relative to a history offset date specified by hist_odate.

Note: a history file is never created on a simulation stopping date, but a history file can be created on a simulation starting date.
See Example: 2

hist_nday

Default: 14
Required: no (only used if hist_freq = "ndays")
Description: If hist_freq = "ndays" > 0, restart files will be created every n days. More specifically, whenever $mod(d-p,n)=0$, where
d=the simulation day (1 Jan, year 1 <=> $d=365$)
n=hist_nday
p=an offset such that $mod(d-p,n)=0$ on the date specified by rest_odate
See Example: 2

hist_odate

Default: 0
Required: no (only used if hist_freq = "ndays")
Description: Selects history file offset calendar date, used in conjunction with hist_nday when hist_freq = "ndays". If hist_odate > 1, it is a valid calendar date (encoded: yyyymmdd). See hist_freq for usage. History data is created every n days relative to (and including) either

- the initial date of the run (if hist_odate ≤ 0)
- the 1st of the month (if hist_odate = 1)
- the date hist_odate (if hist_odate > 1)

See Example: 2

hist_tavg

Default: 1 (true)
Required: no
Description: If hist_tavg is true (non–zero), component models will be requested to create monthly averaged history data files.
See Example: 4

mss_opts

Default: "nowait, nomail"
Required: no
Description: Options used when data files are sent to NCAR's Mass Storage System (MSS) via a mswrite library call. This default request uses asynchronous data file transfer. See the mswrite man pages at NCAR for more details.
See Example: 4

mss_pass

Default: " "
Required: no
Description: Mswrite password string for files sent to the MSS. See the mswrite man pages at NCAR for more details.

See Example: 4

mss_rtpd

Default: 365
Required: no
Description: Retention period for data files sent to the MSS. See the mswrite man pages at NCAR for more details.
See Example: 4

diag_glob

Default: 0 (false)
Required: no
Description: If true (non–zero), print global diagnostic information.
See Example: 2

flux_albd

Default: "instant"
Required: no
Description: Albedo calculation selection, one of "instant" or "davg".

- "instant" => atm is sent albedo fields which have a zenith angle dependence.
- "davg" => atm is sent albedo fields which are zenith angle independent.

flux_solar

Default: 1 (true)
Required: no
Description: If true (non–zero): redistribute solar radiation between ice, land, and ocean models proportional to (1 - albedo).

flux_prec

Default: 0 (false)
Required: no
Description: If true (non–zero): adjust the precipitation field over the non–land ocean domain such that $E + P = 0$ over that ocean and ice surface area.

init_solar

Default: 1 (true) for initial runs, 0 (false) otherwise
Required: no
Description: If true (non–zero), the coupler and atmosphere will exchange an additional message pair on startup. This message pair consists of the coupler sending the merged surface albedos, and the atmosphere returning the resulting downward radiation. The default is that this only happens on an initial run.

init_davg

Default: 1 (true) for initial runs, 0 (false) otherwise
Required: no
Description: If true (non–zero), there is an initial one day simulation in which only models that communicate more than once per day are advanced, allowing

the formation of initial daily average fluxes. After this one special day simulation, all models will begin to advance together. The default is that this only happens on an initial run.

info_dbug

Default: 1
Required: no
Description: Debugging information level: 0, 1, 2, or 3.

- 0 => do not write any extra debugging information to stdout
- 1 => write a small amount of extra debugging information to stdout
- 2 => write a medium amount of extra debugging information to stdout
- 3 => write a large amount of extra debugging information to stdout

See Example: 2

info_time

Default: 0 (false)
Required: no (normally not recommended)
Description: If true (non-zero), write message passing timing info to stdout.

msg_group

Default: "unknown"
Required: yes, always
Description: A character string (up to eight characters) used to uniquely identify the message passing group of the simulation. All connected component models must use the same group identifier.
See Example: 1, 2, 3, 4

msg_method

Default: "default"
Required: no
Description: Message passing between the driver and the component models may be implemented using various message passing libraries (this requires some isolated re-coding within the driver, as well as recoding within all connected coupled models). Some libraries may offer a choice of various methods for passing messages. For example, the MCL 1.0 message passing library (developed within SCD at NCAR), allows several choices of methods:

- "pvm3" => use PVM3 (only) as the underlying message passing method.
- "named-pipe" => use a hybrid of PVM3 and UNIX named-pipes as the underlying message passing method. This normally has a faster data transfer rate, but may be less portable than straight PVM3. See the MCL 1.0 document for more details.
- "default" => the same as selecting pvm3

See Example: 4

msg_maxt

Default: 300
Required: no
Description: the number of seconds elapsed before the message passing library gives up waiting for a message to arrive. The coupled system will shut down if a message fails to arrive.
See Example: 4

Examples of input namelists

Example 1: an initial run

```
E$drv_parm
    rest_str    = "case b001.01 (an initial run) "
    rest_dir    = "/DOE/csm/b001.01/drv/ "
    rest_type   = "initial"
    rest_date   = "19800831"
    stop_option = "date"
    stop_date   = 19801001
    msg_group   = "b001.01"
$
```

Here the inputs specify an initial run, starting on 31 Aug 1980, and stopping on 1 Oct 1980. History data and restart files will be created according to the default settings: there will be a restart file for 1 Sep 1980 and 1 Oct 1980, and there will be a history file for 1 Sep 1980 (but not for 1 Oct 1980). The Flux Coupler will communicate with atmosphere, ocean, and ice models that have also joined the message passing group "b001.01."

Example 2: a continuation run

```
E$drv_parm
    rest_str    = "case b001.01 (a continuation run) "
    rest_dir    = "/DOE/csm/b001.01/drv/ "
    rest_type   = "continue"
    rest_freq   = "ndays"
    stop_option = "newyear"
    hist_freq   = "ndays"
    hist_nday   = 10
    hist_odate  = 0
    info_dbug   = 0
    diag_glob   = 1
    msg_group   = "b001.01"
$
```

Here the inputs specify a continuation run. Assuming this run continues from where example 1 finished, it will start on 1 Oct 1980 and the stop_option will cause the simulation to stop on 1 Jan 1981. Restart files will be created according to the default settings (the 1st of every month, including 1 Jan 1981). History data will be created on the 1st, 11th, 21st, and 31st of every month (including 1 Oct 1980, but excluding 1 Jan 1981). Setting info_dbug = 0 will minimize the amount of debugging information written to stdout. Setting diag_glob = 1 will signal the coupler to calculate some global diagnostic information and write it to stdout.

Example 3: a branch run

```
E$drv_parm
    rest_str    = "case b001.02 (a branch run) "
    rest_dir    = "/DOE/csm/b001.02/drv/ "
    rest_type   = "branch"
    rest_bfile  = "/DOE/csm/b001.01/drv/r1981-01-01 "
    stop_option = "ndays"
    stop_nday   = 181
    msg_group   = "b001.02"
$
```

Here the inputs specify a branch run. The branch file (/DOE/.../r1981-01-01) must be specified. The branch file is an existing file from some previous simulation. Since the simulation's starting date (rest_date) is unspecified in the namelist, the date from the branch file will be used (apparently 1 Jan 1981). The stop_option will cause the simulation to stop after 181 days (*i.e.*, on 1 Jul 1981).

Example 4: a regeneration run

```
E$drv_parm
    rest_str    = "case b001.02 (a regen run) "
    rest_dir    = "/DOE/csm/b001.02/drv/ "
    rest_type   = "regeneration"
    rest_bfile  = "/DOE/csm/b001.02/drv/r1981-01-01 "
    rest_freq   = "ndays"
    rest_nday   = 10
    rest_odate  = 19810701
    stop_option = "date"
    stop_date   = 19810701
    mss_rtpd    = 30
    mss_pass    = ",secret"
    mss_opts    = "nowait"
    msg_group   = "b001.02"
    msg_maxt    = 600
    msg_method  = "named-pipe"
$
```

Here the inputs specify a regeneration run. The initial condition file (/DOE/.../r1981-01-01) must be specified. It is an existing file from the same simulation. This simulation's starting date is taken from the branch file (1 Jan 1980). The stop_option will cause the simulation to stop on 1 Jul 1981. Restart files will be created every 10 days, based on and including 1 Jul 1981. Setting msg_maxt = 600 tells the message passing system to allow up to 10 minutes for a message to arrive (although expected wait times are a fraction of a second). If the coupler is using the MCL 1.0 message passing library, which it will be if it was compiled using the makefile called "Makefile.MCL1.0," see section I, subsection *IV*, setting msg_method = "named-pipe" changes the message passing method from the default (standard PVM3) to an optional named-pipe implementation, see the MCL 1.O document for details. The mass store file retention period, write password, and other options have been selected here.