esm-runscripts

01000101
01010011
01001101 ESM-Tools

More details – Adapt your runscript

Setups vs Models vs Submodels

```
#!/usr/bin/bash

fesom_standalone_set_defaults()
{
        coupled_setup_component_list="fesom"
        coupled_setup_executable_list="fesom"

        SPINUP_fesom=${SPINUP_fesom:-0}

        fesom_VERSION=${fesom_VERSION:-"1.4"}
        if [[ ${fesom_VERSION} = "CMIP6" ]]; then
                fesom_VERSION="1.4"
        fi

        POST_PROCESSING_fesom_standalone=${POST_PROCESSING_fesom_standalone:-0}

        if [[ ${fesom_VERSION} = "1.4" ]]; then
                RES_fesom=${RES_fesom:-REF87K}
        elif [[ ${fesom_VERSION} = "2.0" ]]; then
                RES_fesom=${RES_fesom:-CORE2}
        fi

        eval cores_per_node_fesom=\$\{cores_per_node_fesom:-${cores_per_compute_node}\}
        omp_num_threads_compute_fesom_standalone=1
        omp_num_threads_post_fesom_standalone=8


        TOTAL_NNODES_post=1

        case $cores_per_compute_node in
                36)
                        case $RES_fesom in
                                REF87K)
                                        nnodes_fesom=${nnodes_fesom:-6}
                                        nproca_fesom=${nproca_fesom:-216}
                                        nprocb_fesom=${nprocb_fesom:-1}
                                        ;;
                                CORE2)
                                        nnodes_fesom=${nnodes_fesom:-8}
                                        nproca_fesom=${nproca_fesom:-288}
                                        nprocb_fesom=${nprocb_fesom:-1}
                                        ;;
                                CAVCORE2)
                                        nnodes_fesom=${nnodes_fesom:-8}
                                        nproca_fesom=${nproca_fesom:-288}
                                        nprocb_fesom=${nprocb_fesom:-1}
                                        ;;
                        esac
                        ;;
        esac
```

In the ESM-Tools, there‘s a distinction between a setup (awicm, fesom_standalone,…) and models (fesom, echam, oasis,…).

The difference between a model and its standalone setup is subtle – the model contains everything that is valid for the model no matter if coupled or not; the setup only contains what is needed to start the model in standalone mode.

A submodel is a model that can not run in standalone mode, and is part of a model (jsbach, hdmodel,…)

Inheritance

Information can be passed down from bigger to smaller entities:

setup -> model -> submodel

E.g.:

`POOL_DIR_awicm=/…/`

also sets

`POOL_DIR_echam, POOL_DIR_fesom, ...`

Only passes on vars that end in `_$setup_name` or `_$model`!

Inheritance **DOES NOT** overwrite fields that are set already!

```
echam_VERSION=echam-6.3.04p1
LCTLIBDEF_jsbach=${MODEL_DIR_awicm}/${echam_VERSION}/lctlib_nlct21.def
CF_NAME_TABLE_oasis3mct=${MODEL_DIR_awicm}/oasis/cf_name_table.txt

BIN_DIR_echam=${MODEL_DIR_awicm}/bin/
BIN_DIR_fesom=${MODEL_DIR_awicm}/bin/
EXE_fesom=fesom

BASE_DIR=/work/ab0995/a270058/esm-experiments/

POOL_DIR_awicm=/pool/data/
POOL_DIR_echam=/pool/data/
POOL_DIR_fesom=/work/bm0944/input/

MESH_DIR_fesom=/pool/data/AWICM/FESOM1/MESHES/core/


NYEAR_awicm=0           # Number of years per run
NMONTH_awicm=1          # Number of months per run
```

Choose scenarios

echam, jsbach and mpiom have pre-defined „scenarios“,
which a typical simulations, not necessarily CMIP-style

SCENARIO_echam=
(SCENARIO_jsbach=)

1. PI-CTRL
2. 1850
3. 1950
4. HIST
5. PALEO
6. RCP26/45/85 – mpiesm-1.0 only
7. SCEN
8. 4CO2
9. 1percCO2

Not all scenarios are available for each version of echam / each setup. Check in the namelist folder!

SCENARIO_mpiom=

1. PI-CTRL
2. HIST
3. PALEO

To create a new scenario, it is sufficient to create a subfolder with the needed namelists in esm-runscripts/namelists,
and specify the forcing / input data in the functions

Available resolutions

Resolution of a model can be chosen by setting
`RES_$model`

RES_echam=

1. T31
2. T63
3. T127

RES_mpiom=

1. GR15
2. GR30

RES_fesom=

FESOM1:
1. CORE2
2. GLOB / MR
3. REF87K
4. REF
5. BOLD
6. fArc
7. PI-GRID
8. CAVCORE2

FESOM2:
1. CORE2
2. GLOB / MR

RES_oifs=

1. T95
2. T159
3. T255
4. T511
5. T799
6. T1279

RES_nemo=

1. ORCA05
2. GYRE_XIOS

Postprocessing

Postprocessing can be turned on by setting:
`POST_PROCESSING_$model=1`

ECHAM / JSBACH:

1. BOT
2. ATM
3. LOG
4. QBO
5. co2
6. tracer
7. mm
8. ym
9. dm
10. land
11. jsbach

For echam/jsbach, you can choose the postprocessing method by setting

```
MEANTAGS_echam="BOT ATM mm"
```

Other models might only have one / no postprocessing method pre-defined, so no need to choose.

Restarting from a former run

To restart from any folder, you need to set `LRESUME`, `INI_RESTART_DIR` and `INI_PARENT_DATE`. MPIESM-models also need to know `INI_PARENT_EXP_ID`.

`LRESUME` must be set to 1 in order for the job to restart in the first run. `LRESUME=0` means initial run.

```
LRESUME_echam=1
INI_RESTART_DIR_echam=${BASE_DIR}/${PARENT_EXP_ID}/restart/echam/
INI_PARENT_DATE_echam='20021231'
INI_PARENT_EXP_ID_echam=${PARENT_EXP_ID}

LRESUME_fesom=1
INI_RESTART_DIR_fesom=${BASE_DIR}/${PARENT_EXP_ID}/restart/fesom/
INI_PARENT_DATE_fesom='2002'

LRESUME_oasis3mct=1
INI_RESTART_DIR_oasis3mct=${BASE_DIR}/${PARENT_EXP_ID}/restart/oasis3mct/
INI_PARENT_DATE_oasis3mct='20021231'
```

Restarting from a former esm-tools run

To restart from a previous ESM-Tools run („branching off“) it is sufficient to define `PARENT_DATE_esmstyle` and `PARENT_EXP_ID_esmstyle`.

```
PARENT_DATE_esmstyle=20080131
PARENT_EXP_ID_esmstyle=ws1
```

Setting restart frequency

To set the amount of times a job is restarted, you need to set:

```
NYEAR_awicm=…
NMONTH_awicm=…
```

That says that after n years or m months, restarts are written, the job stops and restarts itself if the current time is less than the `FINAL_DATE`.

```
NYEAR_awicm=1                  # Number of years per run
NMONTH_awicm=0                 # Number of months per run
```

Changing namelist entries

There are several ways to change namelist entries from the runscript, the easiest is to use something like this:

```
calendar___include_fleapyear___nml_entry=".true."
calendar___include_fleapyear___nml_file="namelist.config"
```

This means: In namelist.config, chapter “calendar“, entry „include_fleapyear“ needs to be set to „.true.“

```
calendar___include_fleapyear___nml_entry="REMOVE_FROM_NAMELIST"
calendar___include_fleapyear___nml_file="namelist.config"
```

Additional or different input/forcing/… files

Use this syntax to change a file path, or to add some additional file:

```
echam_myname1_FORCING_FILES="/work/aa0238/a270118/SST_SIC_Forcing/T127_HadNOAAOI_SIC_HICE.nc unit.96 warn"
echam_myname2_FORCING_FILES="/work/aa0238/a270118/SST_SIC_Forcing/T127_HadNOAAOI_SST_LAMO.nc unit.20 warn"
```

This tells echam to take forcing files unit.96 and unit.20 from non-default locations. myname1 and myname2 are identifiers that can be any name you want – just NOT DEFAULT!

user defined functions

You can define small functions within the runscript, these are called depending on their names:

```
echam_user_prepare_exe()
{
        echo "BEEP BEEP"
}

echam_USER_prepare_exe()
{
        echo "Im a sheep"
}
```

The first one is called directly BEFORE, the second directly AFTER the executables are prepared (= copied to the folder structure).

lctlib /cf_name_table

Two typical files that sometimes go missing are lctlib_nlct21.def for jsbach, and the cf_name_table.txt for oasis. You can normally fix this with something like this:

```
LCTLIBDEF_jsbach=${MODEL_DIR_echam_standalone}/lctlib_nlct21.def
CF_NAME_TABLE_oasis3mct=${MODEL_DIR_awicm}/oasis/cf_name_table.txt
```

hyperthreading / BeeOND / xthi

You can set some very technical features of the machine.

1. hyperthreading: Turned on by default on mistral, unavailable on CRAY machines. Using hyperthreading makes your job slower by about 70-80% (typically), but costs only 50% the number of cores. So, time to solution goes up, resource usage goes down.

```
use_hyperthreading=0/1
```

2. BeeOND: ollie has the ability to move the needed files to local SSDs before the run, and move back the output from the SSDs afterwards. Default: Turned off.

```
use_beeond=0/1
```

3. xthi is a small tool used to display the distribution of tasks on physical cores. Good for debugging if your run is much slower than expected.

```
with_xthi=0/1
```