

ICON Namelist Overview

1. Juli 2019

Inhaltsverzeichnis

| | | |
|----------|---|----------|
| 1 | ICON Namelists | 3 |
| 1.1 | Scripts, Namelist files and Programs | 3 |
| 1.2 | Namelist parameters | 4 |
| 2 | Namelist parameters defining the atmospheric model | 4 |
| 2.1 | assimilation_nml | 4 |
| 2.2 | coupling_mode_nml | 6 |
| 2.3 | diffusion_nml | 7 |
| 2.4 | dynamics_nml | 8 |
| 2.5 | echam_cld_nml | 8 |
| 2.6 | echam_cnv_nml | 10 |
| 2.7 | echam_gwd_nml | 11 |
| 2.8 | echam_phy_nml | 12 |
| 2.9 | echam_rad_nml | 13 |
| 2.10 | echam_sso_nml | 17 |
| 2.11 | echam_vdf_nml | 17 |
| 2.12 | ensemble_pert_nml | 18 |
| 2.13 | gribout_nml | 20 |
| 2.14 | grid_nml | 22 |
| 2.15 | gridref_nml | 23 |
| 2.16 | ha_dyn_nml | 25 |
| 2.17 | initicon_nml | 26 |
| 2.18 | interpol_nml | 29 |
| 2.19 | io_nml | 31 |
| 2.19.1 | Restart read/write mode: | 33 |
| 2.20 | les_nml (parameters for LES turbulence scheme; valid for inwp_turb=5) | 34 |
| 2.21 | limarea_nml (Scope: l_limited_area=.TRUE. in grid_nml) | 35 |
| 2.22 | lnd_nml | 37 |
| 2.23 | ls_forcing_nml (parameters for large-scale forcing; valid for torus geometry) | 39 |
| 2.24 | master_nml | 40 |
| 2.25 | master_model_nml (repeated for each model) | 40 |

| | | |
|----------|---|-----------|
| 2.26 | master_time_control_nml | 41 |
| 2.27 | meteogram_output_nml | 41 |
| 2.28 | nonhydrostatic_nml (relevant if run_nml:iequations=3) | 42 |
| 2.29 | nudging_nml | 46 |
| 2.30 | nwp_phy_nml | 47 |
| 2.31 | nwp_tuning_nml | 50 |
| 2.32 | output_nml (relevant if run_nml/output='nml') | 51 |
| 2.33 | parallel_nml | 59 |
| 2.34 | psrad_nml | 61 |
| 2.35 | radiation_nml | 61 |
| 2.36 | run_nml | 65 |
| 2.37 | sleve_nml (relevant if nonhydrostatic_nml:ivctype=2) | 68 |
| 2.38 | synsat_nml | 68 |
| 2.39 | time_nml | 69 |
| 2.40 | transport_nml (used if run_nml/ltransport=.TRUE.) | 71 |
| 2.41 | turbdiff_nml | 73 |
| 2.42 | upatmo_nml | 76 |
| 3 | Ocean-specific namelist parameters | 78 |
| 3.1 | ocean_physics_nml | 78 |
| 3.2 | sea_ice_nml (relevant if run_nml/iforcing=2 (ECHAM)) | 78 |
| 4 | Namelist parameters for testcases (NAMELIST_ICON) | 79 |
| 4.1 | ha_testcase_nml (Scope: ltestcase=.TRUE. and iequations=[0,1,2] in run_nml) | 79 |
| 4.2 | nh_testcase_nml (Scope: ltestcase=.TRUE. and iequations=3 in run_nml) | 81 |
| 5 | External data | 88 |
| 5.1 | extpar_nml (Scope: itopo=1 in run_nml) | 88 |
| 6 | External packages | 90 |
| 7 | Information on vertical level distribution | 90 |
| 8 | Compile flag for mixed precision | 90 |
| A | Arithmetic expression evaluation | 90 |
| A.1 | Examples for arithmetic expressions | 90 |
| A.2 | Expression syntax | 91 |
| A.2.1 | List of functions | 91 |
| A.2.2 | List of operators | 91 |
| A.2.3 | List of available constants | 91 |
| A.3 | Usage with Fortran | 91 |
| A.3.1 | Fortran examples | 91 |
| A.3.2 | Error handling | 92 |
| A.4 | Remarks | 92 |
| B | Changes incompatible with former versions of the model code | 92 |

1 ICON Namelists

1.1 Scripts, Namelist files and Programs

Run scripts starting the programs for the models are stored in run/. These scripts write namelist files containing the specified Fortran namelists. Programs are stored in < icon home>/build/<architecture>/bin/.

Tabelle 1: Namelist files

| Namelist file | Purpose | Made by script | Used by program |
|----------------------|---------------------|-------------------------|------------------------|
| NAMELIST_GRAPH | Generate graphs | create_global_grids.run | grid_command |
| NAMELIST_GRID | Generate grids | create_global_grids.run | grid_command |
| NAMELIST_GRIDREF | Gen. nested domains | create_global_grids.run | grid_command |
| NAMELIST_ICON | Run ICON models | exp.<name>.run | control_model |

1.2 Namelist parameters

The following subsections tabulate all available Fortran namelist parameters by name, type, default value, unit, description, and scope:

- *Type* refers to the type of the Fortran variable, in which the value is stored: I=INTEGER, L=LOGICAL, R=REAL, C=character string
- *Default* is the preset value, if defined, that is assigned to this parameter within the programs.
- *Unit* shows the unit of the control parameter, where applicable.
- *Description* explains in a few words the purpose of the parameter.
- *Scope* explains under which conditions the namelist parameter has any effect, if its scope is restricted to specific settings of other namelist parameters.

Information on the file, where the namelist is defined and used, is given at the end of each table.

2 Namelist parameters defining the atmospheric model

Namelist parameters for the ICON models are organized in several thematic Fortran namelists controlling the experiment, and the properties of dynamics, transport, physics etc.

2.1 assimilation_nml

The main switch for the Latent heat nudging scheme is called ldass_lhn and has to be set in run_nml.

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------|-----------|------|--|---|
| nlhn_start | I | -9999 | s | time in seconds when LHN is applied for the first time | run_nml:ldass_lhn = .true. |
| nlhn_end | I | -9999 | s | time in seconds when LHN is applied for the last time | |
| lhn_coef | R | 1.0 | | Nudging coefficient of adding the increments | |
| fac_lhn_up | R | 2.0 | | Upper limit of the scaling factor of the temperature profile. | |
| fac_lhn_down | R | 0.5 | | Lower limit of the scaling factor of the temperature profile. | fac_lhn_down, fac_lhn_up, fac_lhn_artif |
| lhn_logscale | L | .TRUE. | | Apply all scaling factors as logarithmic values | |
| thres_lhn | R | 0.1/3600. | mm/s | Minimal value of precipitation rate, either of model or radar. LHN will be applied first for precipitation above it. | |
| start_fadeout | R | 1.0 | | Value to determine, at which model time step a fading out of the increments might start. | |
| lhn_qrs | L | .TRUE. | | Use a vertical average of precipitation fluxes as reference to compare with radar observed precipitation, to avoid severe overestimation due to displacement of model surface precipitation. If set .FALSE. the model surface precipitation rate is used as reference. | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|-----------|-------------|--|---|
| rqrsgmax | R | 1.0 | K/s | This value determines the height of the vertical averaging, to obtain the reference precipitation rate It is the model layer where the quotion of the maximal precipitation flux occurred for the first time. | lhn_qrs = .TRUE. |
| lhn_hum_adj | L | .TRUE. | | Apply an increment of specific humidity with respect to the estimated temperature increment to maintain the relative humidty | lhn_hum_adj=.TRUE. |
| lhn_no_ttend | L | .FALSE. | | Only apply moisture increments. Temperature increments will only be used for calculation of moisture increments | |
| lhn_incloud | L | .TRUE. | | Apply increments only in model layers where the underlying latent heat release of the model is positive. | lhn_artif_only=.FALSE. |
| lhn_limit | L | .TRUE. | | Limitation of temperature increments | abs_lhn_lim |
| abs_lhn_lim | R | 50./3600. | | Lower and upper limit for temperature increments to be added. | lhn_limit = .TRUE. |
| lhn_filt | L | .TRUE. | | Vertical smoothing of the profile of temperature increments | |
| lhn_relax | L | .FALSE. | | Horizontal smoothing of radar data but also of incorporated model fields | nlhn_relax |
| nlhn_relax | I | 2 | | Number of horizontal grid point, where smoothing is applied. | lhn_relax = .TRUE. |
| lhn_artif | L | .TRUE. | | Apply an artificial temperature profile to estimate increments at model grid points without significant precipitation (determined by fac_lhn_artif). | fac_lhn_artif, tt_artif_max, zlev_artif_max, std_artif_ma |
| fac_lhn_artif | R | 5.0 | grid points | Value of the ratio of radar to model precipitation rate, from which an artificial temperature profile is applied | lhn_artif=.TRUE. |
| lhn_artif_only | L | .FALSE. | | Scaling the artificial temperature profile instead of local model profile of latent heat release for calculation the increments at any model grid point. The scaling factor is still be determined by the ratio of observed to modelled precipitation rate. | tt_artif_max, zlev_artif_max, std_artif_max |
| tt_artif_max | R | 0.0015 | | Maximal temperature of Gaussian shaped function used a artificial temperature profile. | lhn_artif, lhn_artif_only |
| zlev_artif_max | R | 1000.0 | | Height of maximum of Gaussian shaped function used a artificial temperature profile. | lhn_artif, lhn_artif_only |
| std_artif_max | R | 4.0 | m | Parameter defining width of Gaussian shaped function used a artificial temperature profile. | lhn_artif, lhn_artif_only |
| nlhnverif_start | I | -9999 | s | time in seconds when online verification within LHN is active for the first time | run_nml:lclass_lhn = .true. |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|--------------|---------|------|---|----------------------------|
| nlhnverif_end | I | -9999 | s | time in seconds when online verification within LHN is active for the last time | run_nml:ldass_lhn = .true. |
| lhn_diag | L | .FALSE. | | Enable a extensive diagnostic output, writing into file lhn.log. lhn_diag is set .TRUE. automatically, when online verification is active. | |
| lhn_dt_obs | R | 300.0 | s | Frequency of the radar observations | |
| radar_in | C | ',' | | Path where the radar data file is expected. | lhn_black=.TRUE. |
| radardata_file(:) | C (n_dom) | | | Name of the radar data file. This might be either in GRIB2 or in NetCDF (recommended). | |
| lhn_black | L | .FALSE. | | Apply a blacklist information in the radar data obtained by comparison against satellite cloud information | |
| blacklist_file(:) | C (n_dom) | | | Name of blacklist file, containing a mask concerning the quality of the radar data. Value 1: good quality Value 0: bad quality This might be either in GRIB2 or in NetCDF (recommended). | lhn_bright=.TRUE. |
| lhn_bright | L | .FALSE. | | Apply a model intern bright band detection to avoid strong overestimation due to uncertain radar observations. | |
| height_file(:) | C (n_dom) | | | Name of file containing the height of the lowest scan for each possible radar station within the given radar composite. This file is required, when applying bright band detection. This might be either in GRIB2 or in NetCDF (recommended). | |
| nradar | I (n_dom) | 200 | | Maximal number of radar station contained within height_file | lhn_bright=.TRUE. |

Defined and used in: src/namelist/mo_assimilation_nml.f90

2.2 coupling_mode_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|-------|
| coupled_mode | L | .FALSE. | | .TRUE.: if yac coupling routines have to be called | |

Defined and used in: src/namelist/mo_coupling_nml.f90

2.3 diffusion_nml

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|--------------------------|------|---|---|
| lhdiff_temp | L | .TRUE. | | Diffusion on the temperature field | Options 2, 24 and 42 are allowed only in the hydrostatic atm model (iequations = 1 or 2 in dynamics_nml). |
| lhdiff_vn | L | .TRUE. | | Diffusion on the horizontal wind field | |
| lhdiff_w | L | .TRUE. | | Diffusion on the vertical wind field | |
| hdiff_order | I | 4 (hydro) 5 (NH) | | Order of ∇ operator for diffusion: -1: no diffusion 2: ∇^2 diffusion 3: Smagorinsky ∇^2 diffusion 4: ∇^4 diffusion 5: Smagorinsky ∇^2 diffusion combined with ∇^4 background diffusion as specified via hdiff_efdt_ratio | |
| lsmag_3d | L | .FALSE. | | 24 or 42: ∇^2 diffusion from model top to a certain level (cf. k2_pres_max and k2_klev_max below); ∇^4 for the lower levels. .TRUE.: Use 3D Smagorinsky formulation for computing the horizontal diffusion coefficient (recommended at mesh sizes finer than 1 km if the LES turbulence scheme is not used) | hdiff_order=3 or 5; itype_vn_diffu=1 |
| itype_vn_diffu | I | 1 | | Reconstruction method used for Smagorinsky diffusion: 1: u/v reconstruction at vertices only | iequations=3, hdiff_order=3 or 5 |
| itype_t_diffu | I | 2 | | 2: u/v reconstruction at cells and vertices Discretization of temperature diffusion: 1: $K_h \nabla^2 T$ 2: $\nabla \cdot (K_h \nabla T)$ | iequations=3, hdiff_order=3 or 5 |
| k2_pres_max | R | -99. | Pa | Pressure level above which ∇^2 diffusion is applied. | hdiff_order = 24 or 42, and dynamics_nml:iequations = 1 or 2. |
| k2_klev_max | I | 0 | | Index of the vertical level till which (from the model top) ∇^2 diffusion is applied. If a positive value is specified for k2_pres_max, k2_klev_max is reset accordingly during the initialization of a model run. | hdiff_order = 24 or 42, and dynamics_nml:iequations = 1 or 2. |
| hdiff_efdt_ratio | R | 1.0 (hydro) 36.0 (NH) | | ratio of e-folding time to time step (or 2* time step when using a 3 time level time stepping scheme) (for triangular NH model, values above 30 are recommended when using hdiff_order=5) | iequations=3 |
| hdiff_w_efdt_ratio | R | 15.0 | | ratio of e-folding time to time step for diffusion on vertical wind speed | |
| hdiff_min_efdt_ratio | R | 1.0 | | minimum value of hdiff_efdt_ratio near model top | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------------------|------|----------------------|---------|--|--|
| echam_cld_config(jg)% ccwmin | R | 1.0e-7 | kg/kg | cloud water and ice minimum mass mixing ratio for cover>0 | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cqtmin | R | 1.0e-12 | kg/kg | cloud water/ice minimum for microphysical processes | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cthomi | R | Tmelt-35 = 238.15 | K | maximum temperature for homogeneous freezing | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% csecfrl | R | 5.0e-6 | kg/kg | minimum in-cloud water mass mixing ratio in mixed phase clouds | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% ccraut | R | 15. | | coefficient of autoconversion of cloud droplets to rain | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% ccracl | R | 6. | | coefficient of accretion of cloud droplets by falling rain | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cauloc | R | 10. | | coefficient of local rainwater production by autoconversion | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% clmin | R | 0.0 | | minimum for cauloc*dz/5000 | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% clmax | R | 0.5 | | maximum for cauloc*dz/5000 | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cvtfall | R | 2.5 | | coefficient of sedimentation velocity of cloud ice | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% ceffmin | R | 10. | 1.e-6 m | min effective radius for ice cloud | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% ceffmax | R | 150. | 1.e-6 m | max effective radius for ice cloud | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% crhoi | R | 500. | kg/m3 | density of cloud ice | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% crhosno | R | 100. | kg/m3 | bulk density of snow | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% ccsaut | R | 95.0 | | coefficient of autoconversion of cloud ice to snow | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% ccsacl | R | 0.1 | | coefficient of accretion of cloud droplets by falling snow | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cn1lnd | R | defval | 1e6/m3 | cloud droplet number concentration over land, p <= 100 hPa | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cn2lnd | R | defval | 1e6/m3 | cloud droplet number concentration over land, p >= 800 hPa | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cn1sea | R | defval | 1e6/m3 | cloud droplet number concentration over sea, p <= 100 hPa | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cn2sea | R | defval | 1e6/m3 | cloud droplet number concentration over sea, p >= 800 hPa | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cinhomi | R | 0.8 | | ice cloud inhomogeneity factor | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cinhoml1 | R | 0.8 | | liquid cloud inhomogeneity factor, ktype = 0 = stratiform clouds | echam_phy_config(jg)% dt_cld > 0.000s |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------------------|------|---------|------|--|--|
| echam_cld_config(jg)% cinhoml2 | R | 0.4 | | liquid cloud inhomogeneity factor, ktype = 4 = shallow conv. (cf. clwprat) | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cinhoml3 | R | 0.8 | | liquid cloud inhomogeneity factor, ktype = 1 = deep convection and ktype = 2 = shallow conv. (cf. clwprat) and ktype = 3 = mid-level conv. | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% clwprat | R | 4.0 | | critical ratio of cloud liq.+ice paths below and above the top of shallow convection; for ratio > clwprat -> change ktype from 2 to 4 | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% crs | R | 0.968 | | critical relative humidity at surface | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% crt | R | 0.8 | | critical relative humidity aloft | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% nex | I | 2 | | transition parameter for critical relative humidity profile | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% jbmin | I | 40 | | index of highest level for search of top level of inversion layer over sea (ca. 2 km) | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% jbmax | I | 45 | | index of bottom level of inversion layer over sea | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% cinv | R | 0.25 | | fraction of dry adiabatic lapse rate for search of top level of inversion layer over sea | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% csatsc | R | 0.7 | | minimum effective saturation for cloud cover below an invesion layer over sea | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% ncctop | I | 13 | | index of highest level for tropopause calculation | echam_phy_config(jg)% dt_cld > 0.000s |
| echam_cld_config(jg)% nccbot | I | 35 | | index of lowest level for tropopause calculation | echam_phy_config(jg)% dt_cld > 0.000s |

2.6 echam_cnv_nml

The parameterization of convection for the ECHAM physics is configured by a data structure *echam_cnv_config(jg=1:ndom)%<param>*, which is a 1-dimensional array extending over all domains. The structure contains parameters providing control over the parametrized effects:

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------------------|------|---------|------|---|--|
| echam_cnv_config(jg)% lmfpen | L | .TRUE. | | Switch on penetrative convection. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% lmfmid | L | .TRUE. | | Switch on midlevel convection. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% lmfdd | L | .TRUE. | | Switch on cumulus downdraft. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% lmfdudv | L | .TRUE. | | Switch on cumulus friction. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% entrmid | R | 2.0e-4 | | Entrainment rate for midlevel convection. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% entrscv | R | 3.0e-3 | | Entrainment rate for shallow convection. | echam_phy_config(jg)% dt_cnv > 0.000s |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------------------------|------|---------|------|--|--|
| echam_cnv_config(jg)% entrpen | R | 2.0e-4 | | Entrainment rate for penetrative convection. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% entrdd | R | 4.0e-4 | | Entrainment rate for cumulus downdrafts. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cprcon | R | 2.5e-4 | | Coefficient for determining conversion from cloud water to rain. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cmfctop | R | 0.2 | | Fractional convective mass flux across the top of cloud. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cmfdeps | R | 0.3 | | Fractional convective mass flux for downdrafts at lfs. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cminbuoy | R | 0.02 | | Minimum excess buoyancy. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cmaxbuoy | R | 1.0 | | Maximum excess buoyancy. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cbfac | R | 1.0 | | Factor for std dev of virtual pot temp. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% centrmax | R | 3.0e-4 | | Maximum entrainment/detrainment rate. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% dlev_land | R | 0 | Pa | Minimum pressure thickness of clouds for precipitation over land. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% dlev_ocean | R | 0 | Pa | Minimum pressure thickness of clouds for precipitation over ocean. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cmftau | R | 3600. | | Characteristic convective adjustment time scale. | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cmfcmin | R | 1.0e-10 | | Minimum massflux value (for safety). | echam_phy_config(jg)% dt_cnv > 0.000s |
| echam_cnv_config(jg)% cmfcmax | R | 1.0 | | Maximum massflux value for updrafts. | echam_phy_config(jg)% dt_cnv > 0.000s |

2.7 echam_gwd_nml

The parameterization of atmospheric gravity waves for the ECHAM physics is configured by a data structure *echam_gwd_config(jg=1:ndom)%<param>*, which is a 1-dimensional array extending over all domains. The structure contains parameters providing control over the parametrized effects:

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--|--|
| lheatcal | L | .FALSE. | | .TRUE.: compute drag, heating rate and diffusion coefficient from the dissipation of gravity waves .FALSE.: compute drag only | echam_phy_config(jg)% dt_gwd > 0.000s |
| emiss_lev | I | 10 | | Index of model level, counted from the surface, from which the gravity wave spectra are emitted | echam_phy_config(jg)% dt_gwd > 0.000s |
| rmscon | R | 0.87 | m/s | Root mean square gravity wave wind at the emission level | echam_phy_config(jg)% dt_gwd > 0.000s |
| kstar | R | 5.0e-5 | 1/m | Typical gravity wave horizontal wavenumber | echam_phy_config(jg)% dt_gwd > 0.000s |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--------------------------------------|--|
| m_min | R | 0.0 | 1/m | Minimum bound in vertical wavenumber | echam_phy_config(jg)% dt_gwd > 0.000s |

2.8 echam_phy_nml

The ECHAM physics is configured by a data structure `echam_phy_config(jg=1:ndom)%<param>`, which is a 1-dimensional array extending over all domains. The structure contains several parameters providing time control for the atmospheric forcing by the different parameterizations and additional logical switches for controlling the coupling between the parameterizations of the physics and between the dynamics and the physics. Further logical switches control how the atmospheric boundary conditions for the ECHAM physics are determined. Time control parameters are available for the following atmospheric processes:

| <i>prc</i> | <i>parameterized process</i> |
|------------|--|
| rad | LW and SW radiation |
| vdf | vertical diffusion |
| cnv | cumulus convection |
| cld | cloud microphysics |
| gwd | atmospheric gravity wave drag |
| sso | sub grid scale orographic effects |
| mox | methane oxidation and water vapor photolysis |
| car | Cariolle's linearized ozone chemistry |
| art | ART chemistry |

The time control for an atmospheric forcing by a process *prc* consists of three components, the time interval *dt_prc* for re-computing the forcing, and the start and end dates and times defining the interval *[sd_prc,ed_prc]*, in which the forcing is either computed, if the date/time coincides with the interval *dt_prc*, or recycled. Recycling means that the forcing stored from the last computation is used again. Outside of the interval the forcing is set to zero. If *dt_prc* is not specified, or an empty string or a string of blanks or an interval of length 0s, e.g. *"PT0S"* is given, then the forcing is switched off for the entire experiment and the start and end dates and times are irrelevant. If *sd_prc* or *ed_prc* are not specified, or an empty string or a string of blanks are given, then the experiment start date and the experiment stop date are used, respectively. Further the forcing control switch *fc_prc* can be used to decide if an active process (*dt_prc* > 0) is used for the integration (*fc_prc* = 1) or only computed for diagnostic purposes (*fc_prc* = 0).

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------------------|------|---------|------|--|--|
| echam_phy_config(jg)% dt_prc | C | | | This is the time interval in ISO 8601-2004 format at which the forcing by the process <i>prc</i> is computed. | run_nml/iforcing = 2 |
| echam_phy_config(jg)% sd_prc | C | | | Defines the start date/time in ISO 8601-2004 format of the interval <i>[sd_prc,ed_prc]</i> , in which the forcing by the process <i>prc</i> is computed in intervals <i>dt_prc</i> . | run_nml/iforcing = 2 and <i>dt_prc</i> > 0.000s |
| echam_phy_config(jg)% ed_prc | C | | | Defines the end date/time in ISO 8601-2004 format of the interval <i>[sd_prc,ed_prc]</i> , in which the forcing by the process <i>prc</i> is computed in intervals <i>dt_prc</i> . | run_nml/iforcing = 2 and <i>dt_prc</i> > 0.000s |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------------------|------|---------|------|--|---|
| echam_phy_config(jg)% fc_prc | I | 1 | | Forcing control for process <i>prc</i> . fc_prc = 0: the forcing of the process is not used in the integration. fc_prc = 1: the forcing of the process is used in the integration. | run_nml/forcing = 2 and <i>dt_prc</i> > 0.000s |
| echam_phy_config(jg)% lice | L | .FALSE. | | .TRUE. for sea-ice temperature calculation | run_nml/forcing = 2 |
| echam_phy_config(jg)% lmlo | L | .FALSE. | | .TRUE. for mixed layer ocean | run_nml/forcing = 2 |
| echam_phy_config(jg)% ljsb | L | .FALSE. | | .TRUE. for using the JSBACH land surface model | run_nml/forcing = 2 |
| echam_phy_config(jg)% lamip | L | .FALSE. | | .TRUE. for AMIP boundary conditions | run_nml/forcing = 2 |
| echam_phy_config(jg)% iqneg_d2p | I | 0 | | If negative tracer mass fractions are found in the dynamics to physics interface, then: 1,3: they are reported; 2,3: they are replaced with zero | run_nml/forcing = 2 |
| echam_phy_config(jg)% iqneg_p2d | I | 0 | | If negative tracer mass fractions are found in the dynamics to physics interface, then: 1,3: they are reported; 2,3: they are replaced with zero | run_nml/forcing = 2 |

2.9 echam_rad_nml

The input from ECHAM physics to the PSrad scheme is configured by a data structure *echam_rad_config(jg=1:ndom)%<param>*, which is a 1-dimensional array extending over all domains. The structure contains parameters providing control over the Earth orbit, the computation of the SW incoming flux at the top of the atmosphere and the atmospheric composition:

| Parameter | Type | Default | Unit | Description | Scope |
|---|------|----------|------|--|---|
| echam_rad_config(jg)% isolrad | I | 0 | | Selects the spectral solar irradiation (SSI) at 1 AU distance from the sun 0: SSI of the SRTM scheme, TSI = 1368.222 Wm2. 1: SSI from an external file containing monthly mean time series 2: Average 1844–1856 of the SSI time series provided for CMIP5, TSI = 1360.875 W/m2 3: Average 1979–1988 of the SSI time series provided for CMIP5, TSI = 1361.371 W/m2 4: SSI for RCE–type simulation with diurnal cycle, TSI = 1069.315 W/m2 5: SSI for RCE–type simulation without diurnal cycle, TSI = 433.3371 W/m2 6: Average 1850-1873 of the SSI time series provided for CMIP6, TSI = 1360.744 W/m2 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% fsolrad | R | 1 | | Scaling factor for the SSI | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% l_orbvsop87 | L | .TRUE. | | .TRUE. for the realistic VSOP87 Earth orbit .FALSE. for the Kepler orbit | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% cecc | R | 0.016715 | | eccentricity of the Kepler orbit | echam_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .FALSE. |
| echam_rad_config(jg)% cobld | R | 23.44100 | deg | obliquity of the Earth rotation axis on the Kepler orbit | echam_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .FALSE. |
| echam_rad_config(jg)% clonp | R | 282.7000 | deg | longitude of perihelion with respect to vernal equinox on the Kepler orbit | echam_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .FALSE. |
| echam_rad_config(jg)% lyr_perp | L | .FALSE. | | .FALSE. for transient VSOP87 Earth orbit .TRUE.: VSOP87 Earth orbit of year yr_perp is perpetuated | echam_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .TRUE. |
| echam_rad_config(jg)% yr_perp | L | -99999 | | year to be used for lyr_perp = .TRUE. | echam_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .TRUE. |
| echam_rad_config(jg)% nmonth | I | 0 | | 0: Earth circles on orbit 1-12: Earth orbit position fixed for specified month | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% ldiur | L | .TRUE. | | .TRUE. for diurnal cycle in solar irradiation .FALSE. for zonally averaged solar irradiation | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% l_sph_symm_irr | L | .FALSE. | | .TRUE. for a horizontally independent solar irradiation; .FALSE. for a horizontally resolved solar irradiation | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------------------|------|---------|------|--|---|
| echam_rad_config(jg)% irad_h2o | I | 1 | | Selects source for concentration of water vapor, cloud water and cloud ice 0: set to zero (or epsilon) 1: from tracer | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% irad_co2 | I | 2 | | Selects source for concentration of CO2 0: set to zero (or epsilon) 1: from tracer 2: constant vol. mixing ration set by 'vmr_co2' 4: spatially constant, time dependent vol. mixing ratio from file | echam_phy_config(jg)% dt_rad > 0.000s and CO2 tracer is defined |
| echam_rad_config(jg)% irad_ch4 | I | 3 | | Selects source for concentration of CH4 0: set to zero (or epsilon) 2: constant vol. mixing ration set by 'vmr_ch4' 3: horizontally constant, vertically decaying, with surface vol. mixing ratio set by 'vmr_ch4' 4: horizontally constant, vertically decaying, time dependent with surface vol. mixing ratio from file | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% irad_n2o | I | 3 | | Selects source for concentration of N2O 0: set to zero (or epsilon) 2: constant vol. mixing ration set by 'vmr_n2o' 3: horizontally constant, vertically decaying, with surface vol. mixing ratio set by 'vmr_n2o' 4: horizontally constant, vertically decaying, time dependent with surface vol. mixing ratio from file | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% irad_o3 | I | 0 | | Selects source for concentration of O3 0: set to zero (or epsilon) 1: from tracer 2: 3-dim concentration, climatological annual cycle, monthly means from an annual file bc_ozone_picontrol.nc 4: 3-dim concentration, constant in time, 1st time slice file bc_ozone_picontrol.nc 8: 3-dim concentration, time dependent, monthly means from yearly files bc_ozone_historical_<year>.nc | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% irad_o2 | I | 2 | | Selects source for concentration of O2 0: set to zero (or epsilon) 2: constant vol. mixing ration set by 'vmr_o2' | echam_phy_config(jg)% dt_rad > 0.000s |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------------------------|------|------------|-------|--|--|
| echam_rad_config(jg)% irad_cfc11 | I | 2 | | Selects source for concentration of CFC11 0: set to zero (or epsilon) 2: constant vol. mixing ration set by 'vmr_cfc11' 4: spatially constant, time dependent vol. mixing ratio from file | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% irad_cfc12 | I | 2 | | Selects source for concentration of CFC12 0: set to zero (or epsilon) 2: constant vol. mixing ration set by 'vmr_cfc12' 4: spatially constant, time dependent vol. mixing ratio from file | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% irad_aero | I | 2 | | Selects source for concentration of XYZ 13: tropospheric 'Kinne' aerosols, time dependent from file 14: stratospheric 'Stenchikov' aerosols, time dependent from file 15: tropospheric 'Kinne' aerosols + stratospheric 'Stenchikov' aerosols, time dependent, both from file 18: tropospheric natural 'Kinne' aerosols for 1850 + time dep. stratospheric 'Stenchikov' aerosols, both from file + param. time dep. antropogenic 'simple plumes' any other: set to zero | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% vmr_co2 | R | 348.0e-06 | m3/m3 | Volume mixing ratio of CO2 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% vmr_ch4 | R | 1650.0e-09 | m3/m3 | Volume mixing ratio of CH4 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% vmr_n2o | R | 306.0e-09 | m3/m3 | Volume mixing ratio of N2O | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% vmr_o2 | R | 0.20946 | m3/m3 | Volume mixing ratio of O2 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% vmr_cfc11 | R | 214.5e-12 | m3/m3 | Volume mixing ratio of CFC11 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% vmr_cfc12 | R | 371.1e-12 | m3/m3 | Volume mixing ratio of CFC11 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% frad_h2o | R | 1.0 | | Scaling factor for concentration of water vapor, cloud water and cloud ice | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% frad_co2 | R | 1.0 | | Scaling factor for concentration of CO2 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% frad_ch4 | R | 1.0 | | Scaling factor for concentration of CH4 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% frad_n2o | R | 1.0 | | Scaling factor for concentration of N2O | echam_phy_config(jg)% dt_rad > 0.000s |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------------------|------|---------|------|---|--|
| echam_rad_config(jg)% frad_o3 | R | 1.0 | | Scaling factor for concentration of O3 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% frad_o2 | R | 1.0 | | Scaling factor for concentration of O2 | echam_phy_config(jg)% dt_rad > 0.000s |
| echam_rad_config(jg)% frad_cfc | R | 1.0 | | Scaling factor for concentration of CFC11 and CFC12 | echam_phy_config(jg)% dt_rad > 0.000s |

2.10 echam_sso_nml

The parameterization of subgrid scale orographic (SSO) effects for the ECHAM physics is configured by a data structure *echam_sso_config(jg=1:ndom)%<param>*, which is a 1-dimensional array extending over all domains. The structure contains parameters providing control over the parametrized effects:

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------------------|------|---------|------|--|--|
| echam_sso_config(jg)% gpicea | R | 40. | m | Minimum height difference of peak height and mean height to activate the SSO parameterization. | echam_phy_config(jg)% dt_sso > 0.000s |
| echam_sso_config(jg)% gstd | R | 10. | m | Minimum standard deviation of the SSO height to activate the SSO parameterization. | echam_phy_config(jg)% dt_sso > 0.000s |
| echam_sso_config(jg)% gkdrag | R | 0.05 | | Coefficient for orographic gravity wave drag. | echam_phy_config(jg)% dt_sso > 0.000s |
| echam_sso_config(jg)% gkwake | R | 0. | | Coefficient for low level blocking. | echam_phy_config(jg)% dt_sso > 0.000s |
| echam_sso_config(jg)% gklift | R | 0. | | Coefficient for low level lift. | echam_phy_config(jg)% dt_sso > 0.000s |
| echam_sso_config(jg)% lsftlf | L | .TRUE. | | .FALSE.: SSO effects are directly applied, for the case that SSO parameters are valid for the full cell area. .TRUE.: SSO effects are scaled with the cell area fraction of land including lakes (field sftlf), for the case that SSO parameters are valid only for this part of the cell area. | echam_phy_config(jg)% dt_vdf > 0.000s |

2.11 echam_vdf_nml

The parameterization of vertical diffusion (VDF) for the ECHAM physics is configured by a data structure *echam_vdf_config(jg=1:ndom)%<param>*, which is a 1-dimensional array extending over all domains. The structure contains parameters providing control over some of the parametrized effects:

| Parameter | Type | Default | Unit | Description | Scope |
|---|------|---------|------|---|--|
| echam_vdf_config(jg)% lsfc_mom_flux | L | .TRUE. | | switch on/off surface momentum flux | echam_phy_config(jg)% dt_vdf > 0.000s |
| echam_vdf_config(jg)% lsfc_heat_flux | L | .TRUE. | | switch on/off surface heat flux | echam_phy_config(jg)% dt_vdf > 0.000s |
| echam_vdf_config(jg)% pr0 | R | 1.0 | | neutral limit Prandtl number, can be varied from about 0.6 to 1.0 | echam_phy_config(jg)% dt_vdf > 0.000s |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------------------|------|---------|------|---|--|
| echam_vdf_config(jg)% f_tau0 | R | 0.17 | | neutral non-dimensional stress factor | echam_phy_config(jg)% dt_vdf > 0.000s |
| echam_vdf_config(jg)% c_f | R | 0.185 | | mixing length: coriolis term tuning parameter | echam_phy_config(jg)% dt_vdf > 0.000s |
| echam_vdf_config(jg)% c_n | R | 2.0 | | mixing length: stability term tuning parameter | echam_phy_config(jg)% dt_vdf > 0.000s |
| echam_vdf_config(jg)% wmc | R | 0.5 | | ratio of typical horizontal velocity to wstar at free convection | echam_phy_config(jg)% dt_vdf > 0.000s |
| echam_vdf_config(jg)% fsl | R | 0.4 | | fraction of first-level height at which surface fluxes are nominally evaluated, tuning param for sfc stress | echam_phy_config(jg)% dt_vdf > 0.000s |
| echam_vdf_config(jg)% fbl | R | 3.0 | | 1/fbl: fraction of BL height at which lmix hat its max | echam_phy_config(jg)% dt_vdf > 0.000s |

2.12 ensemble_pert_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|---|-------------------------|
| use_ensemble_pert | L | .FALSE. | | Main switch to activate physics parameter perturbations for ensemble forecasts / ensemble data assimilation; the perturbations are applied via random numbers depending on the perturbationNumber (ensemble member ID) specified in gribout_nml. Perturbations are always turned off if perturbationNumber ≤ 0 | run_nml:iforcing = inwp |
| itype_pert_gen | I | 1 | | Mode of ensemble perturbation generation 1: Equal distribution within perturbation range 2: Discrete distribution with 50% probability for default value and 25% probability for upper and lower extrema | |
| timedep_pert | I | 0 | | Time-dependence of ensemble perturbations (except tkred_sfc, which oscillates with a time scale of 20 days) 0: None 1: Random seed for perturbation generation depends on initial date | |
| range_gkwake | R | 0.5 | | Variability range for low level wake drag constant | |
| range_gkdrag | R | 0.04 | | Variability range for orographic gravity wave drag constant | |
| range_gfrcrit | R | 0.1 | | Variability range for critical Froude number in SSO scheme | |
| range_gfluxlaun | R | 0.75e-3 | | Variability range for non-orographic gravity wave launch momentum flux | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|--------------------------------|---|---------------------|
| range_zvz0i | R | 0.25 | m/s | Variability range for terminal fall velocity of cloud ice | inwp_gscp = 1 or 2 |
| range_rain_n0fac | R | 4. | | Multiplicative change of intercept parameter of raindrop size distribution | inwp_gscp = 1 or 2 |
| range_entrorg | R | 0.2e-3 | 1/m | Variability range for entrainment parameter in convection scheme | inwp_convection = 1 |
| range_rdepths | R | 5.e3 | Pa | Variability range for maximum allowed shallow convection depth | inwp_convection = 1 |
| range_rprcon | R | 0.25e-3 | | Variability range for tuning parameter controlling conversion of cloud water into precipitation | inwp_convection = 1 |
| range_capdcfac_et | R | 0.75 | | Maximum fraction of CAPE diurnal cycle correction applied in the extratropics | icapdcycl = 3 |
| range_rhebc | R | 0.05 | | Variability range for RH threshold for the onset of evaporation below cloud base | inwp_convection = 1 |
| range_textc | R | 0.05 | K | Variability range for temperature excess value in test parcel ascent | inwp_convection = 1 |
| range_qexc | R | 0.005 | | Variability range for mixing ratio excess value in test parcel ascent | inwp_convection = 1 |
| range_box_liq | R | 0.01 | | Variability range for box width scale of liquid clouds in cloud cover scheme | inwp_cldcover = 1 |
| range_box_liq_asy | R | 0.25 | | Variability range for asymmetry factor for sub-grid scale liquid cloud distribution | inwp_cldcover = 1 |
| range_tkhmin | R | 0.2 | m ² s ⁻¹ | Variability range for minimum vertical diffusion for heat/moisture | inwp_turb = 1 |
| range_tkmmin | R | 0.2 | m ² s ⁻¹ | Variability range for minimum vertical diffusion for momentum | inwp_turb = 1 |
| range_turlen | R | 150 | m | Variability range for turbulent mixing length | inwp_turb = 1 |
| range_a_hshr | R | 1 | | Variability range for scaling factor for extended horizontal shear term | inwp_turb = 1 |
| range_a_stab | R | 0 | | Variability range for stability correction | inwp_turb = 1 |
| range_c_diff | R | 1.0 | | Range for multiplicative change of length scale factor for vertical diffusion | inwp_turb = 1 |
| range_q_crit | R | 0 | | Variability range for critical value for normalized supersaturation in turbulent cloud scheme | inwp_turb = 1 |
| range_tkred_sfc | R | 4.0 | | Range for multiplicative change of reduction of minimum diffusion coefficients near the surface | inwp_turb = 1 |
| range_rlam_heat | R | 3.0 | | Variability range (multiplicative!) of laminar transport resistance parameter | inwp_turb = 1 |
| range_charnock | R | 1.5 | | Variability range (multiplicative!) of upper and lower bound of wind-speed dependent Charnock parameter | inwp_turb = 1 |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|---|---------------------------|
| range_minsnowfrac | R | 0.1 | | Variability range for minimum value to which snow cover fraction is artificially reduced in case of melting snow | idiag_snowfrac = 20/30/40 |
| range_c_soil | R | 0.25 | | Variability range for evaporating fraction of soil | |
| range_cwimax_ml | R | 2.0 | | Variability range for capacity of interception storage (multiplicative) | |
| range_z0_lcc | R | 0.25 | | Variability range (relative change) of roughness length attributed to each landuse class | |
| range_rootdp | R | 0.2 | | Variability range (relative change) of root depth attributed to each landuse class | |
| range_rsmin | R | 0.2 | | Variability range (relative change) of minimum stomata resistance attributed to each landuse class | |
| range_laimax | R | 0.15 | | Variability range (relative change) of leaf area index (maximum of annual cycle) attributed to each landuse class | |
| stdev_sst_pert | R | 0. | K | Inserting the standard deviation of SST perturbations (present in the model input data) activates a correction factor for the saturation vapor pressure over oceans, which compensates the systematic increase of evaporation due to the SST perturbations. | |

Defined and used in: src/namelist/mo_ensemble_pert_nml.f90

2.13 gribout_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|-------------|------|--|------------|
| preset | C | "determ..." | | Setting this different to "none" enables a couple of defaults for the other gribout_nml namelist parameters. If, additionally, the user tries to set any of these other parameters to a conflicting value, an error message is thrown. Possible values are "none", "deterministic", "ensemble". | filetype=2 |
| tablesVersion | I | 15 | | Main switch for Table version | filetype=2 |
| backgroundProcess | I | 0 | | Background process - GRIB2 code table backgroundProcess.table | filetype=2 |
| generatingCenter | I | -1 | | Output generating center. If this key is not set, center information is taken from the grid file DWD: 78 MPIMET: 98 ECMWF: 98 | filetype=2 |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------------------|----------|---------|------|--|--------------|
| generatingSubcenter | I | -1 | | Output generating Subcenter. If this key is not set, subcenter information is taken from the grid file DWD: 255 MPIMET: 232 ECMWF: 0 | filetype=2 |
| generatingProcess Identifier | I(n_dom) | 1 | | generating Process Identifier - GRIB2 code table generatingProcessIdentifier.table | filetype=2 |
| numberOfForecastsIn- Ensemble | I | -1 | | Local definiton for ensemble products, (only set if value changed from default) | filetype=2 |
| perturbationNumber | I | -1 | | Local definiton for ensemble products, (only set if value changed from default) | filetype=2 |
| productionStatusOfPro- cessedData | I | 1 | | Production status of data - GRIB2 code table 1.3 | filetype=2 |
| significanceOfReferenceTime | I | 1 | | Significance of reference time - GRIB2 code table 1.2 | filetype=2 |
| typeOfEnsembleForecast | I | -1 | | Local definiton for ensemble products (only set if value changed from default) | filetype=2 |
| typeOfGeneratingProcess | I | -1 | | Type of generating process - GRIB2 code table 4.3 | filetype=2 |
| typeOfProcessedData | I | -1 | | Type of data - GRIB2 code table 1.4 | filetype=2 |
| localDefinitionNumber | I | -1 | | local Definition Number - GRIB2 code table grib2LocalSectionNumber.78.table | filetype=2 |
| localNumberOfExperiment | I | 1 | | local Number of Experiment | filetype=2 |
| localTypeOfEnsemble- Forecast | I | -1 | | Local definiton for ensemble products (only set if value changed from default) | filetype=2 |
| typeOfGrib2TileTemplate | C | “DWD” | | type of GRIB2 templates which are used for decoding tiled surface fields WMO: official WMO templates (55, 59) DWD: local DWD templates (40455, 40456) | filetype = 2 |
| lspecialdate_invar | L | .FALSE. | | Special reference date for invariant and climatological fields .TRUE.: set special reference date 0001-01-01, 00:00 .FASLE.: no special reference date | filetype = 2 |
| ldate_grib_act | L | .TRUE. | | GRIB creation date .TRUE.: add creation date .FALSE.: add dummy date | filetype=2 |
| lgribout_24bit | L | .FALSE. | | If TRUE, write thermodynamic fields ρ , θ_v , T , p with 24bit precision instead of 16bit | filetype=2 |

Defined and used in: src/namelist/mo_gribout_nml.f90

2.14 grid_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------|----------|---------|-------|---|--|
| lplane | L | .FALSE. | | planar option | |
| is_plane_torus | L | .FALSE. | | f-plane approximation on triangular grid | |
| corio_lat | R | 0.0 | deg | Center of the f-plane is located at this geographical latitude | lplane=.TRUE. and is_plane_torus=.TRUE. |
| grid_angular_velocity | R | Earth's | rad/s | The angular velocity in rad per sec. | |
| l_limited_area | L | .FALSE. | | | |
| grid_rescale_factor | R | 1.0 | | Defined as the inverse of the reduced-size earth reduction factor X . Choose grid_rescale_factor < 1 for a reduced-size earth. The geometry and the timestep will be multiplied by this factor. The angular velocity will be divided by this factor. | |
| lfeedback | L(n_dom) | .TRUE. | | Specifies if feedback to parent grid is performed. Setting lfeedback(1)=.false. turns off feedback for all nested domains; to turn off feedback for selected nested domains, set lfeedback(1)=.true. and set ".false." for the desired model domains | n_dom>1 |
| ifedback_type | I | 2 | | 1: incremental feedback 2: relaxation-based feedback Note: vertical nesting requires option 2 to run numerically stable over longer time periods | n_dom>1 |
| start_time | R(n_dom) | 0. | s | Time when a nested domain starts to be active. Relative time w.r.t. experiment start date (ini_datetime_string / experimentStratDate). (namelist entry is ignored for the global domain) | n_dom>1 |
| end_time | R(n_dom) | 1.E30 | s | Time when a nested domain terminates. Relative time w.r.t. experiment start date (ini_datetime_string / experimentStratDate). (namelist entry is ignored for the global domain) | n_dom>1 |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------------|----------|---------|------|---|----------------------|
| patch_weight | R(n_dom) | 0. | | If patch_weight is set to a value > 0 for any of the first level child patches, processor splitting will be performed, i.e. every of the first level child patches gets a subset of the total number of processors corresponding to its patch_weight. A value of 0. corresponds to exactly 1 processor for this patch, regardless of the total number of processors. For the root patch and higher level childs, patch_weight is not used. However, patch_weight must be set to 0 for these patches to avoid confusion. | n_dom>1 |
| lredgrid_phys | L | .FALSE. | | If set to .true. radiation is calculated on a reduced grid (= one grid level higher) | |
| dynamics_grid_filename | C | | | Array of the grid filenames to be used by the dycore. May contain the keyword <path> which will be substituted by model_base_dir. | |
| dynamics_parent_grid_id | I(n_dom) | $i - 1$ | | Array of the indexes of the parent grid filenames, as described by the dynamics_grid_filename array. Indexes start at 1, an index of 0 indicates no parent. Specification of this namelist parameter is only required if more than one domain is in use <i>and</i> the grid files are rather old s.t. they do not contain a UUID attribute. | |
| radiation_grid_filename | C | | | Grid filename to be used for the radiation model on the coarsest grid. Filled only if the radiation grid is different from the dycore grid. May contain the keyword <path> which will be substituted by model_base_dir. | lredgrid_phys=.TRUE. |
| create_vgrid | L | .FALSE. | | .TRUE.: Write vertical grid files containing (vct_a, vct_b, z_ifc, and z_ifv. | |
| vertical_grid_filename | C(n_dom) | | | Array of filenames. These files contain the vertical grid definition (vct_a, vct_b, z_ifc). If empty, the vertical grid is created within ICON during the setup phase. | |
| use_duplicated_connectivity | L | .TRUE. | | if .TRUE., the zero connectivity is replaced by the last non-zero value | |
| use_dummy_cell_closure | L | .FALSE. | | if .TRUE. then create a dummy cell and connect it to cells and edges with no neighbor | |

Defined and used in: src/namelists/mo_grid_nml.f90

2.15 gridref_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|----------|---------|------|---|---------|
| grf_intmethod_c | I | 2 | | Interpolation method for grid refinement (cell-based dynamical variables): 1: parent-to-child copying 2: gradient-based interpolation | n_dom>1 |
| grf_intmethod_ct | I | 2 | | Interpolation method for grid refinement (cell-based tracer variables): 1: parent-to-child copying 2: gradient-based interpolation | n_dom>1 |
| grf_intmethod_e | I | 6 | | Interpolation method for grid refinement (edge-based variables): 1: inverse-distance weighting (IDW) 2: RBF interpolation 3: combination gradient-based / IDW 4: combination gradient-based / RBF 5/6: same as 3/4, respectively, but direct interpolation of mass fluxes along nest interface edges | n_dom>1 |
| grf_velfbk | I | 1 | | Method of velocity feedback: 1: average of child edges 1 and 2 2: 2nd-order method using RBF interpolation | n_dom>1 |
| grf_scalfbk | I | 2 | | Feedback method for dynamical scalar variables (T, p_{sfc}): 1: area-weighted averaging 2: bilinear interpolation | n_dom>1 |
| grf_tracfbk | I | 2 | | Feedback method for tracer variables: 1: area-weighted averaging 2: bilinear interpolation | n_dom>1 |
| grf_idw_exp_e12 | R | 1.2 | | exponent of generalized IDW function for child edges 1/2 | n_dom>1 |
| grf_idw_exp_e34 | R | 1.7 | | exponent of generalized IDW function for child edges 3/4 | n_dom>1 |
| rbf_vec_kern_grf_e | I | 1 | | RBF kernel for grid refinement (edges): 1: Gaussian 2: $1/(1+r^2)$ 3: inverse multiquadric | n_dom>1 |
| rbf_scale_grf_e | R(n_dom) | 0.5 | | RBF scale factor for grid refinement (lateral boundary interpolation to edges). Refers to the respective parent domain and thus does not need to be specified for the innermost nest. Lower values than the default of 0.5 are needed for child mesh sizes less than about 500 m. | n_dom>1 |
| denom_diffu_t | R | 135 | | Denominator for lateral boundary diffusion of temperature | n_dom>1 |
| denom_diffu_v | R | 200 | | Denominator for lateral boundary diffusion of velocity | n_dom>1 |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|---------|------|---|--|
| l_mass_consvcorr | L | .FALSE. | | .TRUE.: Apply mass conservation correction in feedback routine | n_dom>1 |
| l_density_nudging | L | .FALSE. | | .TRUE.: Apply density nudging near lateral nest boundary if grf_intmethod_e ≤ 4 | n_dom>1 .AND. lfeedback = .TRUE. |
| fbk_relax_timescale | R | 10800 | | Relaxation time scale for feedback | n_dom>1 .AND. lfeedback = .TRUE. .AND. ifedback_type = 2 |

Defined and used in: `src/namelist/mo_gridref_nml.f90`

2.16 ha_dyn_nml

This namelist is relevant if `run_nml:ldynamics=.TRUE.` and `dynamics_nml:iequations=IHS_ATM_TEMP` or `IHS_ATM_THETA`.

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|---------|------|---|---|
| itime_scheme | I | 14 | | Time integration scheme: 11: pure advection (no dynamics) 12: 2 time level semi implicit (not yet implemented) 13: 3 time level explicit 14: 3 time level with semi implicit correction 15: standard 4th-order Runge-Kutta method (4-stage) 16: SSPRK(5,4) scheme (5-stage) | |
| ileapfrog_startup | I | 1 | | How to integrate the first time step when the leapfrog scheme is chosen. 1 = Euler forward; 2 = a series of sub-steps. | itime_scheme= 13 or 14 |
| asselin_coeff | R | 0.1 | | Asselin filter coefficient | itime_scheme= 13 or 14 |
| si_2tls | R | 0.6 | | weight of time step n+1. Valid range: [0,1] | itime_scheme=12 |
| si_expl_scheme | I | 2 | | scheme for the explicit part used in the 2 time level semi-implicit time stepping scheme. 1 = Euler forward; 2 = Adams-Bashforth 2nd order | itime_scheme=12 |
| si_cmin | R | 30.0 | m/s | semi implicit correction is done for eigenmodes with speeds larger than si_cmin | itime_scheme=14 and lsi_3d=.FALSE. |
| si_coeff | R | 1.0 | | weight of the semi implicit correction | itime_scheme=14 |
| si_offctr | R | 0.7 | | | itime_scheme=14 |
| si_rtol | R | 1.0e-3 | | relative tolerance for GMRES solver | itime_scheme=14 |
| lsi_3d | L | .FALSE. | | 3D GMRES solver or decomposition into 2D problems | lshallow_water=.FALSE. and itime_scheme=14 |
| ldry_dycore | L | .TRUE. | | Assume dry atmosphere | iequations∈{1,2} |
| lref_temp | L | .FALSE. | | Set a background temperature profile as base state when computing the pressure gradient force | iequations∈{1,2} |

2.17 initicon_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|---|--|
| init_mode | I | 2 | | 1: MODE_DWDANA start from DWD analysis or FG 2: MODE_IFSANA start from IFS analysis 3: MODE_COMBINED IFS atm + ICON/GME soil 4: MODE_COSMO start from prognostic set of variables as used by COSMO 5: MODE_IAU start from DWD analysis with incremental analysis update. Extension of MODE_IAU_OLD including snow increments 6: MODE_IAU_OLD start from DWD analysis with incremental analysis update. NOTE: Extension of mode MODE_DWDANA_INC including W_SO increments. 7: MODE_ICONVREMAP start from DWD first guess with subsequent vertical remapping (work in progress; so far, changing the number of model levels does not yet work) | |
| dt_iau | R | 10800 | s | Duration of incremental analysis update (IAU) procedure. Start time for IAU is the actual model start time (see below). | init_mode=5,6 |
| dt_shift | R | 0 | s | Time by which the actual model start time is shifted ahead of the nominal date. The latter is given by either <code>ini_datetime_string</code> or <code>experimentStartDate</code> . <code>dt_shift</code> must be NEGATIVE, usually -0.5 dt_iau . | init_mode=5,6 |
| iterate_iau | L | .FALSE. | | If .TRUE., the IAU phase is calculated twice with halved <code>dt_shift</code> in first cycle (allows writing a fully initialized analysis at the nominal initialization date while using a centered IAU window for the forecast). | init_mode=5,6 and <code>dt_shift < 0</code> |
| start_time_avg_fg | R | 0 | s | Start time for calculating temporally averaged first guess output for data assimilation. | |
| end_time_avg_fg | R | 0 | s | End time for calculating temporally averaged first guess output for data assimilation. Setting <code>end_time_avg_fg > start_time_avg_fg</code> activates the averaging | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|----------|---------|------|---|---------------------|
| interval_avg_fg | R | 0 | s | Corresponding averaging interval. Note that $\text{end_time_avg_fg} - \text{start_time_avg_fg}$ must not be smaller than the averaging interval | |
| rho_incr_filter_wgt | R | 0 | | Vertical filtering weight on density increments | init_mode=5,6 |
| niter_diffu | I | 10 | | Number of diffusion iterations applied on wind increments | init_mode=5,6 |
| niter_divdamp | I | 25 | | Number of divergence damping iterations applied on wind increments | init_mode=5,6 |
| type_iau_wgt | I | 1 | | Weighting function for performing IAU 1: Top-Hat 2: SIN2 | init_mode=5,6 |
| nlevsoil_in | I | 4 | | number of soil levels of input data | init_mode=2 |
| zpbl1 | R | 500.0 | m | bottom height (AGL) of layer used for gradient computation | |
| zpbl2 | R | 1000.0 | m | top height (AGL) of layer used for gradient computation | |
| lread_ana | L | .TRUE. | | If .FALSE., ICON is started from first guess only. Analysis field is not required, and skipped if provided. | init_mode=1,3 |
| use_lakeiceana | L | .FALSE. | | If .TRUE., analysis data for sea ice fraction are also used for freshwater lakes (for the time being restricted to the Great Lakes; extension to other lakes needs to be tested) | init_mode=5,6 |
| qcana_mode | I | 0 | | If > 0, analysis increments for cloud water are read and processed. 1: QC increments are added to QV increments 2: QC increments are added to QC if clouds are present, otherwise to QV increments | init_mode=5 |
| qiana_mode | I | 0 | | 1: analysis increments for cloud ice are read and processed. | init_mode=5 |
| lconsistency_checks | L | .TRUE. | | If .FALSE., consistency checks for Analysis and First Guess fields are skipped. On default, checks are performed for <i>uuidOfHGrid</i> and <i>validity time</i> . | init_mode=1,3,4,5,6 |
| l_coarse2fine_mode | L(n_dom) | .FALSE. | | If true, apply corrections for coarse-to-fine mesh interpolation to wind and temperature | |
| lp2cintp_incr | L(n_dom) | .FALSE. | | If true, interpolate atmospheric data assimilation increments from parent domain. Can be specified separately for each nested domain; setting the first (global) entry to true activates the interpolation for all nested domains. | init_mode=5,6 |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|----------|---------|------|---|-------------------|
| lp2cintp_sfcana | L(n_dom) | .FALSE. | | If true, interpolate atmospheric surface analysis data from parent domain. Can be specified separately for each nested domain; setting the first (global) entry to true activates the interpolation for all nested domains. | init_mode=5,6 |
| ltile_init | L | .FALSE. | | True: initialize tiled surface fields from a first guess coming from a run without tiles. Along coastlines and lake shores, a neighbor search is executed to fill the variables on previously non-existing land or water points with reasonable values. Should be combined with ltile_coldstart = .TRUE. | init_mode=1,5,6 |
| ltile_coldstart | L | .FALSE. | | If true, tiled surface fields are initialized with tile-averaged fields from a previous run with tiles. | init_mode=1,5,6 |
| lvert_remap_fg | L | .FALSE. | | A neighbor search is applied to subgrid-scale ocean points for SST and sea-ice fraction. If true, vertical remapping is applied to the atmospheric first-guess fields, whereas the analysis increments remain unchanged. The number of model levels must be the same for input and output fields, and the z_ifc (alias HHL) field pertaining to the input fields must be appended to the first-guess file. | init_mode=5,6 |
| ifs2icon_filename | C | | | Filename of IFS2ICON input file, default " <code><path>ifs2icon_R<nroot>B<jlev>_DOM<idom>.nc</code> ". May contain the keywords <code><path></code> which will be substituted by <code>model_base_dir</code> , as well as <code>nroot</code> , <code>nroot0</code> , <code>jlev</code> , and <code>idom</code> defining the current patch. | init_mode=2 |
| dwdfg_filename | C | | | Filename of DWD first-guess input file, default " <code><path>dwdFG_R<nroot>B<jlev>_DOM<idom>.nc</code> ". May contain the keywords <code><path></code> which will be substituted by <code>model_base_dir</code> , as well as <code>nroot</code> , <code>nroot0</code> , <code>jlev</code> , and <code>idom</code> defining the current patch. | init_mode=1,3,5,6 |
| dwdana_filename | C | | | Filename of DWD analysis input file, default " <code><path>dwdana_R<nroot>B<jlev>_DOM<idom>.nc</code> ". May contain the keywords <code><path></code> which will be substituted by <code>model_base_dir</code> , as well as <code>nroot</code> , <code>nroot0</code> , <code>jlev</code> , and <code>idom</code> defining the current patch. | init_mode=1,3,5,6 |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------------|------|-------------|------|---|-----------------|
| filetype | I | -1 (undef.) | | One of CDI's FILETYPE_XXX constants. Possible values: 2 (=FILETYPE_GRB2), 4 (=FILETYPE_NC2). If this parameter has not been set, we try to determine the file type by its extension "*.grb" or ".nc". | init_mode=1,5,6 |
| check_fg(jg)%list | C(:) | | | In ICON a small subset of first guess input fields is declared 'optional', meaning that they are read in if present, but they are not mandatory to start the model. By adding optional fields to this list, they become mandatory for domain jg, such that the model aborts if any of them is missing. This list may include a subset of the optional first guess fields, or even the entire set of first guess fields. On default this list is empty, such that optional fields experience a cold-start initialization if they are missing and the model does not abort. | |
| check_ana(jg)%list | C(:) | | | List of mandatory analysis fields for domain jg that must be present in the analysis file. If these fields are not found, the model aborts. For all other analysis fields, the FG-fields will serve as fallback position. | |
| ana_varnames_map_file | C | | | Dictionary file which maps internal variable names onto GRIB2 shortnames or NetCDF var names. This is a text file with two columns separated by whitespace, where left column: ICON variable name, right column: GRIB2 short name or NetCDF var name. | |
| itype_vert_expol | I | 1 | | Type of vertical extrapolation of initial data: 1: Linear extrapolation (standard) 2: Blend of linear extrapolation and simple climatology. Intended for upper-atmosphere simulations and specific settings can be found in upatmo_nml. Requires: ivctype = 2, 12; l_limited_area = .FALSE. | |

Defined and used in: src/namelists/mo_initicon_nml.f90

2.18 interpol_nml

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------|---------|------|-------------|-------|
| l_intp_c2l | L | .TRUE. | | DEPRECATED | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|---|----------------|
| l_mono_c2l | L | .TRUE. | | Monotonicity can be enforced by demanding that the interpolated value is not higher or lower than the stencil point values. | ihadv_tracer=4 |
| llsq_high_consv | L | .TRUE. | | conservative (T) or non-conservative (F) least-squares reconstruction for high order transport | |
| lsq_high_ord | I | 3 | | polynomial order for high order reconstruction 1: linear 2: quadratic 30: cubic (no 3 rd order cross deriv.) 3: cubic | |
| llsq_lin_consv | L | .FALSE. | | conservative (T) or non-conservative (F) least-squares reconstruction for 2nd order (linear) transport | |
| nudge_efold_width | R | 2.0 | | e-folding width (in units of cell rows) for lateral boundary nudging coefficient. This switch and the following two pertain to one-way nesting and limited-area mode | |
| nudge_max_coeff | R | 0.02 | | Maximum relaxation coefficient for lateral boundary nudging. Recommended range of values for limited-area mode is 0.06 – 0.075. | |
| nudge_zone_width | I | 8 | | Total width (in units of cell rows) for lateral boundary nudging zone. For the limited-area mode, a minimum of 10 is recommended. If < 0 the patch boundary_depth_index is used. | |
| rbf_dim_c2l | I | 10 | | stencil size for direct lon-lat interpolation: 4 = nearest neighbor, 13 = vertex stencil, 10 = edge stencil. | |
| rbf_scale_mode_ll | I | 2 | | Specifies, how the RBF shape parameter is determined for lon-lat interpolation. 1 : lookup table based on grid level 2 : determine automatically. So far, this routine only estimates the smallest value for the shape parameter for which the Cholesky is likely to succeed in floating point arithmetic. 3 : explicitly set shape parameter in each output namelist (namelist parameter <code>output_nml::rbf_scale</code> , p. 56). | |
| rbf_vec_kern_c | I | 1 | | Kernel type for reconstruction at cell centres: 1: Gaussian 3: inverse multiquadric | |
| rbf_vec_kern_e | I | 3 | | Kernel type for reconstruction at edges: 1: Gaussian 3: inverse multiquadric | |
| rbf_vec_kern_ll | I | 1 | | Kernel type for reconstruction at lon-lat-points: | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------------|----------|----------------------|------|---|-------|
| rbf_vec_kern_v | I | 1 | | 1: Gaussian 3: inverse multiquadric Kernel type for reconstruction at vertices: | |
| rbf_vec_scale_c | R(n_dom) | resolution-dependent | | 1: Gaussian 3: inverse multiquadric Scale factor for RBF reconstruction at cell centres | |
| rbf_vec_scale_e | R(n_dom) | resolution-dependent | | Scale factor for RBF reconstruction at edges | |
| rbf_vec_scale_v | R(n_dom) | resolution-dependent | | Scale factor for RBF reconstruction at vertices | |
| support_baryctr_intp | L | .FALSE. | | Flag. If .FALSE. barycentric interpolation is replaced by a fallback interpolation. | |
| lreduced_nestbdry_stencil | L | .FALSE. | | Flag. If .TRUE. then the nest boundary points are taken out from the lat-lon interpolation stencil. | |

Defined and used in: `src/namelist/mo_interpol_nml.f90`

2.19 io_nml

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|---------|------|--|---|
| lkeep_in_sync | L | .FALSE. | | Sync output stream with file on disk after each timestep | |
| dt_diag | R | 86400. | s | diagnostic integral output interval | <code>run_nml:output = "totint"</code> <code>output /= "none" (run_nml)</code> |
| dt_checkpoint | R | 2592000 | s | Time interval for writing restart files. Note that if the value of <code>dt_checkpoint</code> resulting from model default or user's specification is longer than <code>time_nml:dt_restart</code> , it will be reset (by the model) to <code>dt_restart</code> so that at least one restart file is generated during the restart cycle. | |
| inextra_2d | I | 0 | | Number of extra 2D Fields for diagnostic/debugging output. | <code>dynamics_nml:iequations = 3</code> <i>(to be done for 1, 2)</i> |
| inextra_3d | I | 0 | | Number of extra 3D Fields for diagnostic/debugging output. | <code>dynamics_nml:iequations = 3</code> <i>(to be done for 1, 2)</i> |
| lflux_avg | L | .TRUE. | | if .FALSE. the output fluxes are accumulated from the beginning of the run if .TRUE. the output fluxes are average values from the beginning of the run, except of TOT_PREC that would be accumulated | <code>iequations=3</code> <code>iforcing=3</code> |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------|------|--|------------------------------------|
| itype_pres_msl | I | 1 | s | Specifies method for computation of mean sea level pressure (and geopotential at pressure levels below the surface). 1: GME-type extrapolation, 2: stepwise analytical integration, 3: current IFS method, 4: IFS method with consistency correction 5: New DWD method constituting a mixture between IFS and old GME method (departure level for downward extrapolation between 10 m and 150 m AGL depending on elevation) | iforcing=3 output_nml namelists |
| itype_rh | I | 1 | | Specifies method for computation of relative humidity 1: WMO-type: water only (e_s=e_s_water), 2: IFS-type: mixed phase (water and ice), 3: IFS-type with clipping ($rh \leq 100$) | |
| gust_interval | R | 3600. | | Interval over which wind gusts are maximized | |
| output_nml_dict | C | ' ' | | File containing the mapping of variable names to the internal ICON names. May contain the keyword <path> which will be substituted by model_base_dir. The format of this file: One mapping per line, first the name as given in the ml_varlist, hl_varlist, pl_varlist or il_varlist of the output_nml namelists, then the internal ICON name, separated by an arbitrary number of blanks. The line may also start and end with an arbitrary number of blanks. Empty lines or lines starting with # are treated as comments. Names not covered by the mapping are used as they are. | |
| linvert_dict | L | .FALSE. | | If .TRUE., columns in dictionary file output_nml_dict are evaluated in inverse order. This allows using the same dictionary file as for input (ana_varnames_map_file in initicon_nml). | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|---------|------|---|--|
| netcdf_dict | C | ' ' | | File containing the mapping from internal names to names written to NetCDF. May contain the keyword <path> which will be substituted by model_base_dir. The format of this file: One mapping per line, first the name written to NetCDF, then the internal name, separated by an arbitrary number of blanks (<i>inverse to the definition of</i> output_nml_dict). The line may also start and end with an arbitrary number of blanks. Empty lines or lines starting with # are treated as comments. Names not covered by the mapping are output as they are. Note that the specification of output variables, e.g. in ml_varlist, is independent from this renaming, see the namelist parameter output_nml_dict for this. | output_nml namelists, NetCDF output |
| lnetcdf_ft64_output | L | .FALSE. | | If .TRUE. floating point variable output in NetCDF files is written in 64-bit instead of 32-bit accuracy. | |
| restart_file_type | I | 4 | | Type of restart file. One of CDI's FILETYPE_XXX. So far, only 4 (=FILETYPE_NC2) is allowed | |
| restart_write_mode | C | " " | | Restart read/write mode. Allowed settings (character strings!) are listed below. | |
| nrestart_streams | I | 1 | | When using the restart write mode "dedicated procs multifile", it is possible to split the restart output into several files, as if nrestart_streams * num_io_procs restart processes were involved. This speeds up the read-in process, since all the files may then be read in parallel. | restart_write_mode = "dedicated procs multifile" |
| lmask_boundary | L | F | | Set to .TRUE., if interpolation zone should be masked in triangular output. | |

2.19.1 Restart read/write mode:

Allowed settings for restart_write_mode are:

''sync''

'Old' synchronous mode. PE # 0 reads and writes restart files. All other PEs have to wait.

''async''

Asynchronous restart writing: Dedicated PEs (`num_restart_proc > 0`) write restart files while the simulation continues. Restart PEs can only parallelize over different patches. — Read-in: PE # 0 reads while other PEs have to wait.

”joint procs multifile”

All worker PEs write restart files to a dedicated directory. Therefore, the directory itself is called the restart file. The information is stored in a way that it can be read back into the model independent from the processor count and the domain decomposition. — Read-in: All worker PEs read the data in parallel.

”dedicated procs multifile”

In this case, all the restart data is first transferred to memory buffers in dedicated restart writer PEs. After that, the work processes carry on with their work immediately, while the restart writers perform the actual restart writing asynchronously. Restart PEs can parallelize over patches and horizontal indices. — Read-in: All worker PEs read the data in parallel..

” ”

Fallback mode.

If `num_restart_proc == 0 (parallel_nml)`, then this behaves like **”sync”**, otherwise like **”async”**.

Defined and used in: `src/namelist/mo_io_nml.f90`

2.20 `les_nml` (parameters for LES turbulence scheme; valid for `inwp_turb=5`)

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------------|------|---------|------|---|-----------------------------|
| <code>sst</code> | R | 300 | K | sea surface temperature for idealized LES simulations | <code>isrfc_type=5,4</code> |
| <code>shflx</code> | R | 0.1 | Km/s | Kinematic sensible heat flux at surface | <code>isrfc_type = 2</code> |
| <code>lhflx</code> | R | 0 | m/s | Kinematic latent heat flux at surface | <code>isrfc_type = 2</code> |
| <code>isrfc_type</code> | I | 1 | | surface type 0 = No fluxes and zero shear stress 1 = TERRA land physics 2 = fixed surface fluxes 3 = fixed buoyancy fluxes 4 = RICO test case 5 = fixed SST 6 = time varying SST and <code>qv_s</code> case with prescribed roughness length for semi-idealized setups | |
| <code>ufric</code> | R | -999 | m/s | friction velocity for idealized LES simulations; if <code>< 0</code> then it is automatically diagnosed | |
| <code>psfc</code> | R | -999 | Pa | surface pressure for idealized LES simulations; if <code>< 0</code> then it uses the surface pressure from dynamics | |
| <code>min_sfc_wind</code> | R | 1.0 | m/s | Minimum surface wind for surface layer useful in the limit of free convection | |
| <code>is_dry_cbl</code> | L | .FALSE. | | switch for dry convective boundary layer simulations | |
| <code>smag_constant</code> | R | 0.23 | | Smagorinsky constant | |
| <code>km_min</code> | R | 0.0 | | Minimum turbulent viscosity | |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------|------|----------|--------------------------------|--|--------------|
| max_turb_scale | R | 300.0 | m ² /s ³ | Asymtotic maximum turbulence length scale (useful for coarse grid LES and when grid is vertically stretched) | isrfc_type=3 |
| turb_prandtl | R | 0.333333 | | turbulent Prandtl number | |
| bflux | R | 0.0007 | | buoyancy flux for idealized LES simulations (Stevens 2007) | |
| tran_coeff | R | 0.02 | m/s | transfer coefficient near surface for idealized LES simulation (Stevens 2007) | isrfc_type=3 |
| vert_scheme_type | I | 2 | | type of time integration scheme in vertical diffusion 1 = explicit 2 = fully implicit | |
| sampl_freq_sec | R | 60 | s | sampling frequency in seconds for statistical (1D and 0D) output | |
| avg_interval_sec | R | 900 | s | (time) averaging interval in seconds for 1D statistical output | |
| expname | C | ICOLES | | expname to name the statistical output file | |
| ldiag_les_out | L | .FALSE. | | Control for the statistical output in LES mode | |
| les_metric | L | .FALSE. | | Switch to turn on Smagorinsky diffusion with 3D metric terms to account for topography | |

Defined and used in: `src/namelist/mo_les_nml.f90`

2.21 limarea_nml (Scope: l_limited_area=.TRUE. in grid_nml)

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|---|-----------------|
| itype_latbc | I | 0 | s | Type of lateral boundary nudging. 0: constant lateral boundary conditions derived from the initial conditions, 1: time-dependent lateral boundary conditions provided by an external source (IFS, COSMO or a coarser-resolution ICON run), 2: Test mode using time-dependent lateral boundary conditions from a nested ICON run in which the present limited-area domain was operated as a nested grid with identical(!) model level configuration. Available for synchronous read mode (num_prefetch_proc = 0) only! | itype_latbc ≥ 1 |
| dtime_latbc | R | 10800.0 | | Time difference between two consecutive boundary data. | |
| init_latbc_from_fg | L | .FALSE. | | If .TRUE., take lateral boundary conditions for initial time from first guess (or analysis) field | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------------|------|---------|------|--|----------------------|
| nudge_hydro_pres | L | .TRUE. | | If .TRUE., hydrostatic pressure is used to compute lateral boundary nudging (recommended if boundary conditions contain hydrostatic pressure, which is usually the case) | itype_latbc \geq 1 |
| latbc_filename | C | | | Filename of boundary data input file, these files must be located in the latbc_path directory. Default: "prepiconR<nroot>B<jlev>_<y><m><d><h>.nc". The filename may contain keyword tokens (day, hour, etc.) which will be automatically replaced during the run-time. See the table below for a list of allowed keywords. | itype_latbc \geq 1 |
| latbc_path | C | | | Absolute path to boundary data. | itype_latbc \geq 1 |
| latbc_boundary_grid | C | " " | | Grid file defining the lateral boundary. Empty string means: whole domain is read for the lateral boundary. This NetCDF grid file must contain two integer index arrays: int global_cell_index(cell), int global_edge_index(edge), both with attributes nglobal which contains the global size size of the non-sparse cells and edges. | itype_latbc \geq 1 |
| latbc_varnames_map_file | C | | | Dictionary file which maps internal variable names onto GRIB2 shortnames or NetCDF var names. This is a text file with two columns separated by whitespace, where left column: ICON variable name, right column: GRIB2 short name. This list contains variables that are to be read asynchronously for boundary data nudging in a HDCP2 simulation. All new boundary variables that in the future, would be read asynchronously. Need to be added to text file dict.latbc in run folder. | num_prefetch_proc=1 |
| nretries | I | 0 | | If LatBC data is unavailable: number of retries | |
| retry_wait_sec | I | 10 | | If LatBC data is unavailable: idle wait seconds between retries | |

Defined and used in: src/namelist/mo_limarea_nml.f90

Keyword substitution in boundary data filename (latbc_filename):

| | |
|-------|------------------------------------|
| <y> | substituted by year (four digits) |
| <m> | substituted by month (two digits) |
| <d> | substituted by day (two digits) |
| <h> | substituted by hour (two digits) |
| <min> | substituted by minute (two digits) |

<sec> substituted by seconds (two digits)
 <ddhhmmss> substituted by a *relative* day-hour-minute-second string.
 <dddhh> substituted by a *relative* (three-digit) day-hour string.

2.22 Ind_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|---|---|
| nlev_snow | I | 2 | m | number of snow layers | lmulti_snow=.true. |
| ntiles | I | 1 | | number of tiles | ntiles>1 |
| lsnowtile | L | .FALSE. | | .TRUE.: consider snow-covered and snow-free tiles separately | |
| frlnd_thrhd | R | 0.05 | | fraction threshold for creating a land grid point | ntiles>1 |
| frlake_thrhd | R | 0.05 | | fraction threshold for creating a lake grid point | ntiles>1 |
| frsea_thrhd | R | 0.05 | | fraction threshold for creating a sea grid point | ntiles>1 |
| frlndtile_thrhd | R | 0.05 | | fraction threshold for retaining the respective tile for a grid point | ntiles>1 |
| lmelt | L | .TRUE. | | .TRUE. soil model with melting process | init_mode=1 |
| lmelt_var | L | .TRUE. | | .TRUE. freezing temperature dependent on water content | |
| lana_rho_snow | L | .TRUE. | | .TRUE. take rho_snow-values from analysis file | lmulti_snow = .FALSE. |
| lmulti_snow | L | .FALSE. | | .TRUE. for use of multi-layer snow model (default is single-sayer scheme) | |
| l2lay_rho_snow | L | .FALSE. | | .TRUE. predict additional snow density for upper part of the snowpack, having a maximum depth of max_toplaydepth | lmulti_snow=.TRUE. or l2lay_rho_snow=.TRUE. |
| max_toplaydepth | R | 0.25 | | maximum depth of uppermost snow layer | |
| idiag_snowfrac | I | 1 | | Type of snow-fraction diagnosis: 1 = based on SWE only 2–4 = more advanced experimental methods 20, 30, 40 = same as 2, 3, 4, respectively, but with artificial reduction of snow fraction in case of melting snow (shold be used only in combination with lsnowtile=.TRUE. | lsnowtile=.TRUE. |
| itype_snowevap | I | 2 | | Tuning of snow evaporation in vegetated areas: 1: Tuning turned off 2: First level of tuning without additional control variables 3: Second level of tuning with additional I/O variables for snow age and maximum snow depth (should be used only if these additional variables are avaiable from the DWD assimilation cycle) | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|--|------------------------|
| itype_lndtbl | I | 3 | m | Table values used for associating surface parameters to land-cover classes: 1 = defaults from extpar (GLC2000 and GLOBCOVER2009) 2 = Tuned version based on IFS values for globcover classes (GLOBCOVER2009 only) 3 = even more tuned operational version (GLOBCOVER2009 only) 4 = tuned version for new bare soil evaporation scheme (itype_evsl=4) | itype_interception = 1 |
| itype_root | I | 2 | | root density distribution: 1 = constant 2 = exponential | |
| itype_evsl | I | 2 | | type of bare soil evaporation parameterization 2 = Dickinson (1984) 3 = Noilhan and Planton (1989) 4 = Resistance-based scheme by Jan-Peter Schulz | |
| itype_trvg | I | 2 | | type of plant transpiration parameterization 2 = BATS scheme, Dickinson (1984) 3 = Extended BATS scheme with additional prognostic variable for integrated plant transpiration since sunrise; should be used only with an appropriate first guess for this variable coming from the DWD assimilation cycle | |
| itype_heatcond | I | 2 | | type of soil heat conductivity 1 = constant soil heat conductivity 2 = moisture dependent soil heat conductivity 3 = variant of option 2 with reduced near-surface heat conductivity in the presence of plant cover | |
| itype_interception | I | 1 | | type of plant interception 1 = standard scheme, effectively switched off by tiny value cwimax_ml 2 = Rain and snow interception (to be removed) | |
| cwimax_ml | R | 1.e-6 | | scaling parameter for maximum interception storage (almost switched off); use 5.e-4 to activate interception storage | |
| c_soil | R | 1. | | surface area density of the (evaporative) soil surface allowed range: 0 – 2 | |
| c_soil_urb | R | 1. | | surface area density of the (evaporative) soil surface, urban areas allowed range: 0 – 2 | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------|------|--|----------------------------|
| itype_hydbound | I | 1 | | type of hydraulic lower boundary condition 1 = none 3 = ground water as lower boundary of soil column | lseaice=.TRUE. |
| lstomata | L | .TRUE. | | If .TRUE., use map of minimum stomatal resistance | |
| l2tls | L | .TRUE. | | If .FALSE., use constant value of 150 s/m. If .TRUE., forecast with 2-Time-Level integration scheme (mandatory in ICON) | |
| lseaice | L | .TRUE. | | .TRUE. for use of sea-ice model | |
| lprog_albsi | L | .FALSE. | | If .TRUE., sea-ice albedo is computed prognostically | |
| llake | L | .TRUE. | | .TRUE. for use of lake model | iequations=3 iforcing=3 |
| sstice_mode | I | 1 | | 1: SST and sea ice fraction are read from the analysis. The SST is kept constant whereas the sea ice fraction can be modified by the seaice model. 2: SST and sea ice fraction are read from the analysis. The SST is updated by climatological increments on a daily basis. The sea ice fraction can be modified by the seaice model. 3: SST and sea ice fraction are updated daily, based on climatological monthly means 4: SST and sea ice fraction are updated daily, based on actual monthly means 5: SST and sea ice fraction are updated daily, based on actual daily means (not yet implemented) | |
| sst_td_filename | C | | | Filename of SST input files for time dependent SST. Default is "<path>SST_<year>_<month>_<gridfile>". May contain the keyword <path> which will be substituted by model_base_dir | sstice_mode=3,4,5 |
| ci_td_filename | C | | | Filename of sea ice fraction input files for time dependent sea ice fraction. Default is "<path>CI_<year>_<month>_<gridfile>". May contain the keyword <path> which will be substituted by model_base_dir | sstice_mode=3,4,5 |

Defined and used in: src/namelist/mo_lnd_nwp_nml.f90

2.23 ls_forcing_nml (parameters for large-scale forcing; valid for torus geometry)

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|---------|------|---|--|
| is_subsidence_moment | L | .FALSE. | | switch for enabling LS vertical advection due to subsidence for momentum equations | is_plane_torus=.TRUE. |
| is_subsidence_heat | L | .FALSE. | | switch for enabling LS vertical advection due to subsidence for thermal equations | is_plane_torus=.TRUE. |
| is_advection | L | .FALSE. | | switch for enabling LS horizontal advection (currently only for thermal equations) | is_plane_torus=.TRUE. |
| is_nudging | L | .FALSE. | | switch for enabling LS Newtonian relaxation (nudging) for horizontal winds, temperature and specific humidity | is_plane_torus=.TRUE. |
| is_geowind | L | .FALSE. | | switch for enabling geostrophic wind | is_plane_torus=.TRUE. |
| is_rad_forcing | L | .FALSE. | | switch for enabling radiative forcing | is_plane_torus=.TRUE. inwp_rad=.FALSE. |
| is_theta | L | .FALSE. | | switch to indicate that the prescribed radiative forcing is for potential temperature | is_plane_torus=.TRUE. is_rad_forcing=.TRUE. |

Defined and used in: `src/namelist/mo_ls_forcing_nml.f90`

2.24 master_nml

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------------|------|---------|------|--|-------|
| institute | C | ' ' | | Acronym of the institute for which the full institute name is printed in the log file. Options are DWD, MPIM, KIT, or CSCS. Otherwise the full names of MPIM and DWD are printed. | |
| lrestart | L | .FALSE. | | If .TRUE.: Current experiment is started from a restart. | |
| read_restart_namelist | L | .TRUE. | | If .TRUE.: Namelists are read from the restart file to override the default namelist settings, before reading new namelists from the run script. Otherwise the namelists stored in the restart file are ignored. | |
| lrestart_write_last | L | .FALSE. | | If .TRUE.: model run should create restart at experiment end. This is independent from the settings of the restart interval. | |
| model_base_dir | C | ' ' | | General path which may be used in file names of other name lists: If a file name contains the keyword "<path>", then this model_base_dir will be substituted. | |

2.25 master_model_nml (repeated for each model)

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|---|-------|
| model_name | C | | | Character string for naming this component. | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------------|------|---------|------|--|-------|
| model_namelist_filename | C | | | File name containing the model namelists. | |
| model_type | I | -1 | | Identifies which component to run. 1=atmosphere 2=ocean 3=radiation 99=dummy_model | |
| model_min_rank | I | 0 | | Start MPI rank for this model. | |
| model_max_rank | I | -1 | | End MPI rank for this model. | |
| model_inc_rank | I | 1 | | Stride of MPI ranks. | |

2.26 master_time_control_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------------|------|--------------------------|----------------------|--|-------|
| calendar | I | 1 | | Selects the calendar type to use: 0 = calendar is not defined yet 1 = proleptic Gregorian calendar 2 = 365 day year without leap years 3 = 360 day year with 30 day months | |
| experimentReferenceDate | C | experiment start date | | This specifies the reference date for the calendar in use. It is an anchor date for cycling of events on the time line. If this namelist parameter is unspecified, then the reference date is set to the experiment start date. | |
| experimentStartDate | C | ISO8601 formatted string | date of initial file | This is the start date of an experiment, which remains valid for the whole experiment. The start date is also the reference date of the experiment, which is the anchor point for cycling events. In special cases the reference date might be reset. Reasons might be debugging purposes or spinning off experiments from an existing restart of an other experiment. | |
| experimentStopDate | C | ISO8601 formatted string | n/a | This is the date an experiment is finished. | |
| forecastLeadTime | C | ISO8601 formatted string | n/a | Specifies the time span for a numerical weather forecast. It is used to set the experiment stop time with respect to the experiment start date. | |
| checkpointTimeIntVal | C | ISO8601 formatted string | n/a | Time interval for writing checkpoints. | |
| restartTimeIntVal | C | ISO8601 formatted string | n/a | Time interval for writing a restart file and interrupt the current running job. | |

2.27 meteogram_output_nml

Nearest neighbour 'interpolation' is used for all variables.

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|----------|--------------------------|------|---|-------|
| lmeteogram_enabled | L(n_dom) | .FALSE. | | Flag. True, if meteogram of output variables is desired. | |
| zprefix | C(n_dom) | “METEOGRAM_” | | string with file name prefix for output file | |
| ldistributed | L(n_dom) | .TRUE. | | Flag. Separate files for each PE. | |
| loutput_tiles | L | .FALSE. | | Write tile-specific output for some selected surface/soil fields | |
| n0_mtgrm | I(n_dom) | 0 | | initial time step for meteogram output. | |
| ninc_mtgrm | I(n_dom) | 1 | | output interval (in time steps) | |
| stationlist_tot | | 53.633, 9.983, 'Hamburg' | | list of meteogram stations (triples with lat, lon, name string) | |
| silent_flush | L(n_dom) | 1 | | do not warn about flushing to disk if .TRUE. | |
| max_time_stamps | I(n_dom) | 1 | | number of output time steps to record in memory before flushing to disk | |
| var_list | C(:) | | | Positive-list of variables (optional). Only variables contained in this list are included in the meteogram. If the default list is not changed by user input, then all available variables are added to the meteogram | |

Defined and used in: `src/namelist/mo_mtgrm_nml.f90`

2.28 nonhydrostatic_nml (relevant if run_nml:iequations=3)

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|---------|------|--|--------------|
| itime_scheme | I | 4 | | Options for predictor-corrector time-stepping scheme: 4: Contravariant vertical velocity is computed in the predictor step only, velocity tendencies are computed in the corrector step only (most efficient option) 5: Contravariant vertical velocity is computed in both substeps (beneficial for numerical stability in very-high resolution setups with extremely steep slopes, otherwise no significant impact) 6: As 5, but velocity tendencies are also computed in both substeps (no apparent benefit, but more expensive) | iequations=3 |
| rayleigh_type | I | 2 | | Type of Rayleigh damping 1: CLASSICAL (requires velocity reference state!) 2: Klemp (2008) type | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------|----------|-----------|------|---|----------------------------------|
| rayleigh_coeff | R(n_dom) | 0.05 | | Rayleigh damping coefficient $1/\tau_0$ (Klemp, Dudhia, Hassiotis: MWR136, pp.3987-4004); higher values are recommended for R2B6 or finer resolution | |
| damp_height | R(n_dom) | 45000 | m | Height at which Rayleigh damping of vertical wind starts (needs to be adjusted to model top height; the damping layer should have a depth of at least 20 km when the model top is above the stratopause) | |
| htop_moist_proc | R | 22500.0 | m | Height above which moist physics and advection of cloud and precipitation variables are turned off | |
| hbot_qvsubstep | R | 22500.0 | m | Height above which QV is advected with substepping scheme (must be at least as large as htop_moist_proc) | ihadv_tracer=22, 32, 42 or 52 |
| htop_tracer_proc | R | 1000000.0 | m | Height above which physical processes and advection of additional tracer variables are turned off; the default value is set to an very high value, i.e. by default this possible restriction is not active. This value is taken for all additional tracers in the tracer container with an index equal or greater than iqt; it may be overwritten for specific ART tracers by the tag 'htop_proc' in the XML file when defining the individual ART tracers. | tracers with an index \geq iqt |
| vwind_offctr | R | 0.15 | | Off-centering in vertical wind solver. Higher values may be needed for R2B5 or coarser grids when the model top is above 50 km. Negative values are not allowed | |
| rhotheta_offctr | R | -0.1 | | Off-centering of density and potential temperature at interface level (may be set to 0.0 for R2B6 or finer grids; positive values are not recommended) | |
| veladv_offctr | R | 0.25 | | Off-centering of velocity advection in corrector step. Negative values are not recommended | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|---------|------|--|---------------------|
| ivctype | I | 2 | | Type of vertical coordinate: 1: Gal-Chen hybrid 2: SLEVE (uses sleve_nml) 12: as 2, but nominal interface heights (vct_a (& vct_b)) from file, as in case of 1. Requires: ldeepatmo = .TRUE. (and layer_thickness < 0, to trigger read-in of vertical coordinates from file). Please, see <icon_home>/vertical_coord_tables/README: section “atm_hyb_sz_<nlev>” for the format of the coordinate file, and <icon_home>/src/atm_dyn_iconam... .../mo_init_vgrid: init_sleve_coord for the entrie to the column “vct_b” of the file. (Please, use with care. It has not been thoroughly checked for all possible negative interferences with other parts of the code.) | |
| ndyn_substeps | I | 5 | | number of dynamics substeps per fast-physics / transport step | |
| lhdiff_rcf | L | .TRUE. | | .TRUE.: Compute diffusion only at advection time steps (in this case, divergence damping is applied in the dynamical core) | |
| lextra_diffu | L | .TRUE. | | .TRUE.: Apply additional momentum diffusion at grid points close to the stability limit for vertical advection (becomes effective extremely rarely in practice; this is mostly an emergency fix for pathological cases with very large orographic gravity waves) | |
| divdamp_fac | R | 0.0025 | | Scaling factor for divergence damping | lhdiff_rcf = .TRUE. |
| divdamp_order | I | 4 | | Order of divergence damping: 2 = second-order divergence damping 4 = fourth-order divergence damping 24 = combined second-order and fourth-order divergence damping and enhanced vertical wind off-centering during the initial spinup phase (does not allow checkpointing/restarting earlier than 2.5 hours of integration) | lhdiff_rcf = .TRUE. |
| divdamp_type | I | 3 | | Type of divergence damping: 2 = divergence damping acting on 2D divergence 3 = divergence damping acting on 3D divergence 32 = combination of 3D div. damping in the troposphere with transition to 2D div. damping in the stratosphere | lhdiff_rcf = .TRUE. |
| divdamp_trans_start | R | 12500. | | Lower bound of transition zone between 2D and 3D divergence damping | divdamp_type = 32 |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|---|---|
| divdamp_trans_end | R | 17500. | m | Upper bound of transition zone between 2D and 3D divergence damping | divdamp_type = 32 |
| nest_substeps | I | 2 | | Number of dynamics substeps for the child patches. DO NOT CHANGE!!! The code will not work correctly with other values | ifedback_type=1 |
| l_masscorr_nest | L | .FALSE. | | .TRUE.: Apply mass conservation correction also in nested domain | |
| iadv_rhotheta | I | 2 | | Advection method for rho and rhotheta: 1: simple second-order upwind-biased scheme 2: 2nd order Miura horizontal 3: 3rd order Miura horizontal (not recommended) | |
| igradp_method | I | 3 | | Discretization of horizontal pressure gradient: 1: conventional discretization with metric correction term 2: Taylor-expansion-based reconstruction of pressure (advantageous at very high resolution) 3: Similar discretization as option 2, but uses hydrostatic approximation for downward extrapolation over steep slopes 4: Cubic/quadratic polynomial interpolation for pressure reconstruction 5: Same as 4, but hydrostatic approximation for downward extrapolation over steep slopes | |
| l_zdiffu_t | L | .TRUE. | | .TRUE.: Compute Smagorinsky temperature diffusion truly horizontally over steep slopes | hdiff_order=3/5 .AND. lhdiff_temp = .true. hdiff_order=3/5 .AND. lhdiff_temp=.true. .AND. l_zdiffu_t=.true. hdiff_order=3/5 .AND. lhdiff_temp=.true. .AND. l_zdiffu_t=.true. |
| thslp_zdiffu | R | 0.025 | | Slope threshold above which truly horizontal temperature diffusion is activated | |
| thhgt_d_zdiffu | R | 200 | | Threshold of height difference between neighboring grid points above which truly horizontal temperature diffusion is activated (alternative criterion to thslp_zdiffu) | |
| exner_expol | R | 1./3. | | Temporal extrapolation (fraction of dt) of Exner function for computation of horizontal pressure gradient. This damps horizontally propagating sound waves. For R2B5 or coarser grids, values between 1/2 and 2/3 are recommended. Model will be numerically unstable for negative values. | |
| l_open_ubc | L | .FALSE. | | .TRUE.: Use open upper boundary condition (rather than w=0) to allow vertical motions related to diabatic heating to extend beyond the model top | |

2.29 nudging_nml

Parameters for the upper boundary nudging in the limited-area mode (grid_nml: l_limited_area = .TRUE.). For the lateral boundary nudging, please see interpol_nml and limarea_nml. The characteristics of the driving data for the nudging can be specified in limarea_nml.

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------------------|------|---------|------|---|---|
| nudge_type | I | 0 | | Nudging type: 0: none 1: upper boundary nudging Please note: - nudge_type = 1 requires l_limited_area = .TRUE. - nudging is applied in primary domain only | run_nml:iforcing = 3 (NWP) ivctype = 2 (SLEVE) |
| max_nudge_coeff_vn | R | 0.04 | 1 | Max. nudging coefficient for the horizontal wind (i.e. the edge-normal wind component v_n). Given the wind update due to the nudging term on the rhs: $v_n(t) = v_n^*(t) + \text{nudge_coeff_vn}(z) * \text{ndyn_substeps} * [\overline{v_n}(t) - v_n^*(t)],$ where t and z denote time and height, respectively, $\overline{v_n}(t)$ is the target wind to nudge to, and v_n^* is the value before the nudging, the vertical profile of the nudging coefficient reads: $\text{nudge_coeff_vn}(z) = \text{max_nudge_coeff_vn} * [(z - \text{nudge_start_height}) / (\text{top_height} - \text{nudge_start_height})]^2,$ for $\text{nudge_start_height} \leq z \leq \text{top_height}$ (see nudge_start_height below), and is zero elsewhere. The range of validity is $\text{max_nudge_coeff_vn} \in [0, \sim 1/\text{ndyn_substeps}]$, where the lower boundary is mandatory. | nudge_type > 0 |
| max_nudge_coeff_... ...thermdyn | R | 0.075 | 1 | Max. nudging coefficient for the thermodynamic variables selected by limarea_nml: nudge_hydro_pres. The range of validity is $\text{max_nudge_coeff_thermdyn} \in [0, \sim 1/\text{ndyn_substeps}]$, where the lower boundary is mandatory. | nudge_type > 0 |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|---|----------------|
| nudge_start_height | R | 12000 | m | Nudging is applied for: $\text{nudge_start_height} \leq z \leq \text{top_height}$, where z denotes the nominal height of the grid layer center, and top_height is the height of the model top (see sleeve_nml). | nudge_type > 0 |

Defined and used in: `src/namelist/mo_nudging_nml.f90`

2.30 nwp_phy_nml

The switches for the physics schemes and the time steps can be set for each model domain individually. If only one value is specified, it is copied to all child domains, implying that the same set of parameterizations and time steps is used in all domains. If the number of values given in the namelist is larger than 1 but less than the number of model domains, then the settings from the highest domain ID are used for the remaining model domains.

If the time steps are not an integer multiple of the advective time step (dtime), then the time step of the respective physics parameterization is automatically rounded to the next higher integer multiple of the advective time step. If the radiation time step is not an integer multiple of the cloud-cover time step it is automatically rounded to the next higher integer multiple of the cloud cover time step.

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------|-------------|---------|-------|--|---|
| inwp_gscp | I (max_dom) | 1 | | cloud microphysics and precipitation 0: none 1: hydci (COSMO-EU microphysics, 2-cat ice: cloud ice, snow) 2: hydci_gr (COSMO-DE microphysics, 3-cat ice: cloud ice, snow, graupel) 3: as 1, but with improved ice nucleation scheme by C. Koehler 4: Two-moment microphysics by A. Seifert 9: Kessler scheme | run_nml:iforcing = inwp |
| qi0 | R | 0.0 | kg/kg | cloud ice threshold for autoconversion | inwp_gscp=1 |
| qc0 | R | 0.0 | kg/kg | cloud water threshold for autoconversion | inwp_gscp=1 |
| mu_rain | R | 0.0 | | shape parameter in gamma distribution for rain | inwp_gscp>0 |
| rain_n0_factor | R | 1.0 | | tuning factor for intercept parameter of raindrop size distribution | inwp_gscp>0 |
| mu_snow | R | 0.0 | | shape parameter in gamma distribution for snow | inwp_gscp>0 |
| icpl_aero_gscp | I | 0 | | 0: off 1: simple coupling between autoconversion and Tegen aerosol climatology; requires irad_aero=6 More advanced options are in preparation | inwp_gscp>0 currently only for inwp_gscp = 1 |
| inwp_convection | I (max_dom) | 1 | | convection 0: none 1: Tiedtke/Bechtold convection | run_nml:iforcing = inwp |
| lshallowconv_only | L (max_dom) | .FALSE. | | .TRUE.: use shallow convection only | inwp_convection = 1 |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------|-----------------|---------|------|---|-------------------------|
| ldetrain_conv_prec | L (max_ dom) | .FALSE. | | .TRUE.: Activate detrainment of convective rain and snow | inwp_convection = 1 |
| icapdcycl | I | 0 | | Type of CAPE correction to improve diurnal cycle for convection: 0 = none (IFS default prior to autumn 2013) 1 = intermediate testing option 2 = correctoins over land and water now operational at ECMWF 3 = correction over land as in 2 restricted to the tropics, no correction over water (this choice optimizes the NWP skill scores) | inwp_convection = 1 |
| icpl_aero_conv | I | 0 | | 0: off 1: simple coupling between autoconversion and Tegen aerosol climatology; requires irad_aero=6 | irad_aero=6 |
| iprog_aero | I | 0 | | 0: off 1: simple prognostic aerosol scheme for mineral dust, based on 2D aerosol optical depth fields of Tegen climatology 2: as option 1, but for all 5 aerosol types | |
| icpl_o3_tp | I | 1 | | 0: off 1: simple coupling between the ozone mixing ratio and the thermal tropopause, restricted to the extratropics | irad_o3 = 7 or 9 |
| inwp_cldcover | I (max_ dom) | 1 | | cloud cover scheme for radiation 0: no clouds (only QV) 1: diagnostic cloud cover (by Martin Koehler) 2: prognostic total water variance (not yet started) 3: clouds from COSMO SGS cloud scheme 4: clouds as in turbulence (turbdiff) 5: grid scale clouds | run_nml:iforcing = inwp |
| inwp_radiation | I (max_ dom) | 1 | | radiation 0: none 1: RRTM radiation 2: Ritter-Geleyn radiation 3: PSRAD radiation | run_nml:iforcing = inwp |
| inwp_satad | I | 1 | | saturation adjustment 0: none 1: saturation adjustment at constant density | run_nml:iforcing = inwp |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|-----------------|---------------|------|--|--|
| inwp_turb | I (max_ dom) | 1 | | vertical diffusion and transfer 0: none 1: COSMO diffusion and transfer 2: GME turbulence scheme 3: EDMF-DUALM (work in progress) 5: Classical Smagorinsky diffusion | run_nml:iforcing = inwp |
| inwp_sso | I (max_ dom) | 1 | | subgrid scale orographic drag 0: none 1: Lott and Miller scheme (COSMO) | run_nml:iforcing = inwp inwp_turb > 0 |
| inwp_gwd | I (max_ dom) | 1 | | non-orographic gravity wave drag 0: none 1: Orr-Ern-Bechtold-scheme (IFS) | run_nml:iforcing = inwp inwp_turb > 0 |
| inwp_surface | I (max_ dom) | 1 | | surface scheme 0: none 1: TERRA | run_nml:iforcing = inwp |
| ustart_raylfric | R | 160.0 | m/s | wind speed at which extra Rayleigh friction starts | inwp_gwd > 0 |
| efdt_min_raylfric | R | 10800. | s | minimum e-folding time of Rayleigh friction (effective for u > ustart_raylfric + 90 m/s) | inwp_gwd > 0 |
| latm_above_top | L (max_ dom) | .FALSE. | | .TRUE.: take into account atmosphere above model top for radiation computation | inwp_radiation > 0 |
| itype_z0 | I | 2 | | Type of roughness length data used for turbulence scheme: 1 = land-cover-related roughness including contribution from sub-scale orography (does not account for tiles) 2 = land-cover-related roughness based on tile-specific landuse class 3 = land-cover-related roughness based on tile-specific landuse class including contribution from sub-scale orography | inwp_turb > 0 |
| dt_conv | R (max_ dom) | 600. | s | time interval of convection and cloud-cover call. If convection is switched off, dt_conv controls the time intervall of cloud-cover, only. | run_nml:iforcing = inwp |
| dt_rad | R (max_ dom) | 1800. | s | currently each subdomain has the same value time interval of radiation call | run_nml:iforcing = inwp |
| dt_sso | R (max_ dom) | 1200. | s | currently each subdomain has the same value time interval of sso call | run_nml:iforcing = inwp |
| dt_gwd | R (max_ dom) | 1200. | s | currently each subdomain has the same value time interval of gwd call | run_nml:iforcing = inwp |
| lrtm_filename | C(:) | “rrtmg_lw.nc” | | currently each subdomain has the same value NetCDF file containing longwave absorption coefficients and other data for RRTMG_LW k-distribution model. | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------------------------------|------|--|-------|
| cldopt_filename | C(:) | “ECHAM 6_CldOpt Props.nc” | | NetCDF file with RRTM Cloud Optical Properties for ECHAM6. | |

Defined and used in: `src/namelist/mo_nwp_phy_nml.f90`

2.31 nwp_tuning_nml

Please note: These tuning parameters are NOT domain specific.

| Parameter | Type | Default | Unit | Description | Scope |
|---|-----------------|---------|------|---|-------------------------|
| SSO (Lott and Miller) | | | | | |
| tune_gkwake | R (max_ dom) | 1.5 | | low level wake drag constant | run_nml:iforcing = inwp |
| tune_gkdrag | R (max_ dom) | 0.075 | | gravity wave drag constant | run_nml:iforcing = inwp |
| tune_gfrcrit | R (max_ dom) | 0.4 | | critical Froude number (controls depth of blocking layer) | run_nml:iforcing = inwp |
| tune_grcrit | R (max_ dom) | 0.25 | | critical Richardson number (controls onset of wave breaking) | run_nml:iforcing = inwp |
| GWD (Warner McIntyre) | | | | | |
| tune_gfluxlaun | R | 2.50e-3 | | total launch momentum flux in each azimuth (rho_o x F_o) | run_nml:iforcing = inwp |
| Grid scale microphysics (one moment) | | | | | |
| tune_zceff_min | R | 0.01 | | Minimum value for sticking efficiency | run_nml:iforcing = inwp |
| tune_v0snow | R | 25.0 | | factor in the terminal velocity for snow | run_nml:iforcing = inwp |
| tune_zvz0i | R | 1.25 | m/s | Terminal fall velocity of ice | run_nml:iforcing = inwp |
| tune_icesedi_exp | R | 0.33 | | Exponent for density correction of cloud ice sedimentation | run_nml:iforcing = inwp |
| Convection scheme | | | | | |
| tune_entrorg | R | 1.85e-3 | 1/m | Entrainment parameter valid for dx=20 km (depends on model resolution) | run_nml:iforcing = inwp |
| tune_rprcon | R | 1.4e-3 | | Coefficient for conversion of cloud water into precipitation | run_nml:iforcing = inwp |
| tune_rdepths | R | 2.e4 | Pa | Maximum allowed depth of shallow convection | run_nml:iforcing = inwp |
| tune_capdcfac_et | R | 0.5 | | Fraction of CAPE diurnal cycle correction applied in the extratropics | icapdcycl = 3 |
| tune_rhebc_land | R | 0.75 | | RH threshold for onset of evaporation below cloud base over land | run_nml:iforcing = inwp |
| tune_rhebc_land_trop | R | 0.75 | | RH threshold for onset of evaporation below cloud base over land in the tropics | run_nml:iforcing = inwp |
| tune_rhebc_ocean | R | 0.85 | | RH threshold for onset of evaporation below cloud base over sea | run_nml:iforcing = inwp |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------|------|---------|------|--|---|
| tune_rhebc_ocean_trop | R | 0.80 | | RH threshold for onset of evaporation below cloud base over sea in the tropics | run_nml:iforcing = inwp |
| tune_rcucov | R | 0.05 | | Convective area fraction used for computing evaporation below cloud base | run_nml:iforcing = inwp |
| tune_rcucov_trop | R | 0.05 | | Convective area fraction used for computing evaporation below cloud base in the tropics | run_nml:iforcing = inwp |
| tune_textc | R | 0.125 | K | Excess value for temperature used in test parcel ascent | run_nml:iforcing = inwp |
| tune_qexc | R | 0.0125 | | Excess fraction of grid-scale QV used in test parcel ascent | run_nml:iforcing = inwp |
| tune_box_liq | R | 0.05 | | Box width for liquid cloud diagnostic in cloud cover scheme | run_nml:iforcing = inwp; inwp_cldcover = 1 |
| tune_thicklayfac | R | 0.0025 | 1/m | Factor for enhancing the box width for model layer thicknesses exceeding 150 m | run_nml:iforcing = inwp; inwp_cldcover = 1 |
| tune_box_liq_asy | R | 2.5 | | Asymmetry factor for liquid cloud cover diagnostic | run_nml:iforcing = inwp; inwp_cldcover = 1 |
| lcalib_clcov | L | .TRUE. | | Apply calibration of layer-wise cloud cover diagnostics | run_nml:iforcing = inwp |
| Misc | | | | | |
| tune_gust_factor | R | 8.0 | | Multiplicative factor for friction velocity in gust parameterization | run_nml:iforcing = inwp |
| itune_albedo | I | 0 | | MODIS albedo tuning 0: None 1: dimmed sahara | run_nml:iforcing = inwp albedo_type=2 |
| tune_minsnowfrac | R | 0.2 | | Minimum value to which the snow cover fraction is artificially reduced in case of melting show | lnd_nml:idiag_snowfrac = 20/30/40 |
| IAU | | | | | |
| max_freshsnow_inc | R | 0.025 | | Maximum allowed freshsnow increment per analysis cycle (positive or negative) | init_mode=5 (MODE_IAU) |

Defined and used in: `src/namelist/mo_nwp_tuning_nml.f90`

2.32 output_nml (relevant if run_nml/output='nml')

Please note: There may be several instances of output_nml in the namelist file, every one defining a list of variables with separate attributes for output.

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------|---------|------|---|-------|
| dom | I(:) | -1 | | Array of domains for which this name-list is used. If not specified (or specified as -1 as the first array member), this name-list will be used for all domains. Attention: Depending on the setting of the parameter l_output_phys_patch these are either logical or physical domain numbers! | |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------|------|------------------|------|--|-------|
| file_interval | C | ?? | | Defines the length of a file in terms of an ISO-8601 duration string. An example for this time stamp format is given below. This namelist parameter can be set instead of steps_per_file . | |
| filename_format | C | see description. | | Output filename format. Includes keywords path , output_filename , physdom , etc. (see below). Default is <output_filename>_DOM<physdom>_<levtype>_<jfile> | |
| filename_extn | C | "default" | | User-specified filename extension (empty string also possible). If this namelist parameter is chosen as "default", then we have ".nc" for NetCDF output files, and ".grb" for GRIB1/2. | |
| filetype | I | 4 | | One of CDI's FILETYPE_XXX constants. Possible values: 2=FILETYPE_GRB2, 4=FILETYPE_NC2, 5=FILETYPE_NC4 | |
| m_levels | C | None | | Model level indices (optional). Allowed is a comma- (or semicolon-) separated list of integers, and of integer ranges like "10...20". One may also use the keyword "nlev" to denote the maximum integer (or, equivalently, "nör "N"). Furthermore, arithmetic expressions like "(nlev - 2)" are possible. Basic example: m_levels = "1,3,5...10,20...(nlev-2)" | |
| h_levels | R(:) | None | m | height levels | |
| p_levels | R(:) | None | Pa | pressure levels | |
| i_levels | R(:) | None | K | isentropic levels | |
| ml_varlist | C(:) | None | | Name of model level fields to be output. | |
| hl_varlist | C(:) | None | | Name of height level fields to be output. | |
| pl_varlist | C(:) | None | | Name of pressure level fields to be output. | |
| il_varlist | C(:) | None | | Name of isentropic level fields to be output. | |
| include_last | L | .TRUE. | | Flag whether to include the last time step | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------------|------------|---------|------|--|--------|
| mode | I | 2 | | 1 = forecast mode, 2 = climate mode In climate mode the time axis of the output file is set to TAXIS_ABSOLUTE. In forecast mode it is set to TAXIS_RELATIVE. Till now the forecast mode only works if the output is at multiples of 1 hour | mode=1 |
| taxis_tunit | I | 2 | | Time unit of the TAXIS_RELATIVE time axis. 1 = TUNIT_SECOND 2 = TUNIT_MINUTE 5 = TUNIT_HOUR 9 = TUNIT_DAY | |
| output_bounds | R($k*3$) | None | | For a complete list of possible values see cdilib.c Post-processing times: start, end, increment. We choose the advection time step matching or following the requested output time, therefore we require <code>output_bounds(3) > dtime</code> . Multiple triples are possible in order to define multiple starts/ends/intervals. See namelist parameters <code>output_start</code> , <code>output_end</code> , <code>output_interval</code> for an alternative specification of output events. | |
| output_time_unit | I | 1 | | Units of output bounds specification. 1 = second 2 = minute 3 = hour 4 = day 5 = month 6 = year | |
| output_filename | C | None | | Output filename prefix (which may include path). Domain number, level type, file number and extension will be added, according to the format given in namelist parameter "filename_format". | |
| output_grid | L | .FALSE. | | Flag whether grid information is added to output. | |
| output_start | C(:) | ' ' | | ISO8601 time stamp for begin of output. An example for this time stamp format is given below. More than one value is possible in order to define multiple start/end/interval triples. See namelist parameter <code>output_bounds</code> for an alternative specification of output events. | |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------|------|---------------|------|---|-------|
| output_end | C(:) | τ τ | | ISO8601 time stamp for end of output. An example for this time stamp format is given below. More than one value is possible in order to define multiple start/end/interval triples. See namelist parameter output_bounds for an alternative specification of output events. | |
| output_interval | C(:) | τ τ | | ISO8601 time stamp for repeating output intervals. We choose the advection time step matching or following the requested output time, therefore we require output_bounds(3) > dtime . An example for this time stamp format is given below. More than one value is possible in order to define multiple start/end/interval triples. See namelist parameter output_bounds for an alternative specification of output events. | |
| operation | C | None | | Use this variable for internal diagnostics applied on all given output variables or groups except time-constant ones: mean for generating time averaged, square for time averaged square values, max or min for maximum and minimum values within the corresponding interval, i.e. output_interval . Supported are 2D, 3D and single values like global means on model levels of all components. All operations can be used on global and nested grids. | |
| pe_placement_il | I(:) | -1 | | Advanced output option: Explicit assignment of output MPI ranks to the isentropic level output file. At most stream_partitions_il different ranks can be specified. See namelist parameter pe_placement_ml for further details. | |
| pe_placement_hl | I(:) | -1 | | Advanced output option: Explicit assignment of output MPI ranks to the height level output file. At most stream_partitions_hl different ranks can be specified. See namelist parameter pe_placement_ml for further details. | |
| pe_placement_ml | I(:) | -1 | | Advanced output option: Explicit assignment of output MPI ranks to the model level output file. At most stream_partitions_ml different ranks can be specified, out of the following list: 0 ... (num_io_procs - 1). If this namelist parameters is not provided, then the output ranks are chosen in a Round-Robin fashion among those ranks that are not occupied by explicitly placed output files. | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|-----------|------|--|---------|
| pe_placement_pl | I(:) | -1 | | Advanced output option: Explicit assignment of output MPI ranks to the pressure level output file. At most stream_partitions_pl different ranks can be specified. See namelist parameter pe_placement_ml for further details. | |
| ready_file | C | 'default' | | A <i>ready file</i> is a technique for handling dependencies between the NWP processes. The completion of the write process is signalled by creating a small file with name ready_file . Different output_nml 's may be joined together to form a single ready file event. The setting of ready_file = "default" does not create a ready file. The ready file name may contain string tokens <path>, <datetime>, <ddhhmmss>, <dddhhmmss> which are substituted as described for the namelist parameter filename_format . | |
| reg_def_mode | I | 0 | | Specify if the "delta" value prescribes an interval size or the total *number* of intervals: 0: switch automatically between increment and no. of grid points, 1: reg_lon/lat_def (2) specifies increment, 2: reg_lon/lat_def (2) specifies no. of grid points. | remap=1 |
| remap | I | 0 | | interpolate horizontally 0: none 1: to regular lat-lon grid | |
| north_pole | R(2) | 0,90 | | definition of north pole for rotated lon-lat grids ([longitude , latitude]. | |
| reg_lat_def | R(3) | None | | start, increment, end latitude in degrees. Alternatively, the user may set the number of grid points instead of an increment. Details for the setting of regular grids is given below together with an example. | remap=1 |
| reg_lon_def | R(3) | None | | The regular grid points are specified by three values: start, increment, end given in degrees. Alternatively, the user may set the number of grid points instead of an increment. Details for the setting of regular grids is given below together with an example. | remap=1 |
| steps_per_file | I | -1 | | Max number of output steps in one output file. If this number is reached, a new output file will be opened. Setting steps_per_file to 1 enforces a flush when writing is completed, so that the file is immediately accessible for reading. | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------|------|------------|------|---|----------------------------------|
| steps_per_file_inclfirst | L | see descr. | | Defines if first step is counted wrt. steps_per_file files count. The default is .FALSE. for GRIB2 output, and .TRUE. otherwise. | interpol_nml:rbf_scale_mode_ll=3 |
| stream_partitions_hl | I | 1 | | Splits height level output of this namelist into several concurrent alternating files. See namelist parameter stream_partitions_ml for details. | |
| stream_partitions_il | I | 1 | | Splits isentropic level output of this namelist into several concurrent alternating files. See namelist parameter stream_partitions_ml for details. | |
| stream_partitions_ml | I | 1 | | Splits model level output of this namelist into several concurrent alternating files. The output is split into N files, where the start date of part i gets an offset of $(i - 1) * \text{output_interval}$. The output interval is then replaced by $N * \text{output_interval}$, the include_last flag is set to .FALSE. , the steps_per_file_inclfirst flag is set to .FALSE. , and the steps_per_file counter is set to 1. | |
| stream_partitions_pl | I | 1 | | Splits pressure level output of this namelist into several concurrent alternating files. See namelist parameter stream_partitions_ml for details. | |
| rbf_scale | R | -1. | | Explicit setting of RBF shape parameter for interpolated lon-lat output. This namelist parameter is only active in combination with interpol_nml:rbf_scale_mode_ll=3 . | |

Defined and used in: `src/io/shared/mo_name_list_output_init.f90`

Interpolation onto regular grids: Horizontal interpolation onto regular grids is possible through the namelist setting **remap=1**, where the mesh is defined by the parameters

- **reg_lon_def**: mesh latitudes in degrees,
- **reg_lat_def**: mesh longitudes in degrees,
- **north_pole**: definition of north pole for rotated lon-lat grids.

The regular grid points in **reg_lon_def**, **reg_lat_def** are each specified by three values, given in degrees: *start*, *increment*, *end*. The mesh then contains all grid points $start + k * increment \leq end$, where k is an integer. Instead of defining an increment it is also possible to prescribe the number of grid points.

- Setting the namelist parameter **reg_def_mode=0**: Switch automatically from increment specification to no. of grid points, when the **reg_lon/lat_def(2)** value is larger than 5.0.
- 1: **reg_lon/lat_def(2)** specifies increment
- 2: **reg_lon/lat_def(2)** specifies no. of grid points

For longitude values the last grid point is omitted if the end point matches the start point, e.g. for 0 and 360 degrees.

Examples

local grid with 0.5 degree increment:

```
reg_lon_def = -30.,0.5,30.  
reg_lat_def = 90.,-0.5, -90.
```

global grid with 720x361 grid points:

```
reg_lon_def = 0.,720,360.  
reg_lat_def = -90.,360,90.
```

Time stamp format: The namelist parameters `output_start`, `output_end`, `output_interval` allow the specification of time stamps according to ISO 8601. The general format for time stamps is YYYY-MM-DDThh:mm:ss where Y: year, M: month, D: day for dates, and hh: hour, mm: minute, ss: second for time strings. The general format for durations is PnYnMnDTnHnMnS. See, for example, http://en.wikipedia.org/wiki/ISO_8601 for details and further specifications.

NOTE: as the mtime library underlying the output driver currently has some restrictions concerning the specification of durations:

1. Any number n in PnYnMnDTnHnMnS must have two digits. For instance use "PT06H" instead of "PT6H"
2. In a duration string PnyearYnmonMndayDTnhrHnminMnsecS the numbers nxyz must not pass the carry over number to the next larger time unit: 0<=nmon<=12, 0<=nhr<=23, 0<=nmin<=59, 0<=nsec<=59.999. For instance use "P01D" instead of "PT24H", or "PT01M" instead of "PT60S".

Soon the formatting problem will be resolved and the valid number ranges will be enlarged. (2013-12-16).

Examples

date and time representation (`output_start`, `output_end`)

```
2013-10-27T13:41:00Z
```

duration (`output_interval`)

```
P00DT06H00M00S
```

Variable Groups

Keyword "group:": Using the "group:" keyword for the namelist parameters `ml_varlist`, `hl_varlist`, `pl_varlist`, sets of common variables can be added to the output:

| | |
|------------------------------|--|
| group:all | output of all variables (caution: do not combine with <u>mixed</u> vertical interpolation) |
| group:atmo_ml_vars | basic atmospheric variables on model levels |
| group:atmo_pl_vars | same set as atmo_ml_vars, but except pres |
| group:atmo_zl_vars | same set as atmo_ml_vars, but expect height |
| group:nh_prog_vars | additional prognostic variables of the nonhydrostatic model |
| group:atmo_derived_vars | derived atmospheric variables |
| group:rad_vars | |
| group:precip_vars | |
| group:cloud_diag | |
| group:pbl_vars | |
| group:phys_tendencies | |
| group:land_vars | |
| group:snow_vars | snow variables |
| group:multisnow_vars | multi-layer snow variables |
| group:additional_precip_vars | |
| group:dwd_fg_atm_vars | DWD first guess fields (atmosphere) |
| group:dwd_fg_sfc_vars | DWD first guess fields (surface/soil) |
| group:ART_AERO_VOLC | ART volcanic ash fields |
| group:ART_AERO_RADIO | ART radioactive tracer fields |
| group:ART_AERO_DUST | ART mineral dust aerosol fields |

| | |
|-----------------------|--|
| group:ART_AERO_SEAS | ART sea salt aerosol fields |
| group:prog_timemean | time mean output: temp, u, v, rho |
| group:tracer_timemean | time mean output: qv, qc, qi |
| group:echam_timemean | time mean output: most echam surface variables |
| group:atmo_timemean | time mean variables from prog_timemean,tracer_timemean, echam_timemean |

Keyword "tiles:" The "tiles:" keyword allows to add all tiles of a specific variable to the output, without the need to specify all tile fields separately. E.g. "tiles:t_g" (read: "tiles of t_g") automatically adds all t_g-t_X fields to the output. Here, X is a placeholder for the tile number. Make sure to specify the name of the aggregated variable rather than the name of the corresponding tile container (i.e. in the given example it must be t_g, and not t_g-t!).

Note:

There exists a special syntax which allows to remove variables from the output list, e. g. if these undesired variables were contained in a previously selected group.

Typing <varname>" (for example temp") removes the variable from the union set of group variables and other selected variables. Note that typos are not detected but that the corresponding variable is simply not removed!

Keyword substitution in output filename (filename_format):

| | |
|-----------------|---|
| path | substituted by model_base_dir |
| output_filename | substituted by output_filename |
| physdom | substituted by physical patch ID |
| levtype | substituted by level type "ML", "PL", "HL", "IL" |
| levtype_l | like levtype, but in lower case |
| jfile | substituted by output file counter |
| datetime | substituted by ISO-8601 date-time stamp in format YYYY-MM-DDThh:mm:ss.sssZ |
| datetime2 | substituted by ISO-8601 date-time stamp in format YYYYMMDDThhmmssZ |
| datetime3 | substituted by ISO-8601 date-time stamp in format YYYYMMDDThhmmss.sssZ |
| ddhhmmss | substituted by <i>relative</i> day-hour-minute-second string |
| dddhhmmss | substituted by <i>relative</i> three-digit day-hour-minute-second string |
| hhhhmmss | substituted by <i>relative</i> hour-minute-second string |
| npartitions | If namelist is split into concurrent files: number of stream partitions. |
| ifile_partition | If namelist is split into concurrent files: stream partition index of this file. |
| total_index | If namelist is split into concurrent files: substituted by the file counter (like in jfile), which an "unsplit"namelist would have produced |

2.33 parallel_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|--|---------------------|
| nproma | I | 1 | | chunk length | |
| n_ghost_rows | I | 1 | | number of halo cell rows | |
| division_method | I | 1 | | method of domain decomposition 0: read in from file 1: use built-in geometric subdivision | |
| division_file_name | C | | | Name of division file | division_method = 0 |
| ldiv_phys_dom | L | .TRUE. | | .TRUE.: split into physical domains before computing domain decomposition (in case of merged domains) (This reduces load imbalance; turning off this option is not recommended except for very small processor numbers) | division_method = 1 |
| p_test_run | L | .FALSE. | | .TRUE. means verification run for MPI parallelization (PE 0 processes full domain) | p_test_run = .TRUE. |
| num_test_pe | I | -1 | | If set to more than 1, use this many ranks for testing and switch to different consistency test. This enables tests for identity in setups which are too big to run on a single rank but is limited to comparing one MPI parallelization setup vs. another, obviously. | |
| l_test_openmp | L | .FALSE. | | if .TRUE. is combined with p_test_run=.TRUE. and OpenMP parallelization, the test PE gets only 1 thread in order to verify the OpenMP parallelization | p_test_run = .TRUE. |
| l_log_checks | L | .FALSE. | | if .TRUE. messages are generated during each synchronization step (use for debugging only) | |
| l_fast_sum | L | .FALSE. | | if .TRUE., use fast (not processor-configuration-invariant) global summation | |
| use_dycore_barrier | L | .FALSE. | | if .TRUE., set an MPI barrier at the beginning of the nonhydrostatic solver (do not use for production runs!) | |
| itype_exch_barrier | I | 0 | | 1: set an MPI barrier at the beginning of each MPI exchange call 2: set an MPI barrier after each MPI WAIT call 3: 1+2 (do not use for production runs!) | |
| iorder_sendrecv | I | 1 | | Sequence of send/receive calls: 1 = irecv/send 2 = isend/recv 3 = isend/irecv | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------------|------|---------|------|--|----------------------|
| default_comm- _pattern_type | I | 1 | | Default implementation of mo_communication to be used: 1 = original 2 = YAXT | itype_latbc \geq 1 |
| itype_comm | I | 1 | | 1: use local memory for exchange buffers 3: asynchronous halo communication for dynamical core (currently deactivated) | |
| num_io_procs | I | 0 | | Number of I/O processors (running exclusively for doing I/O) | |
| num_restart_procs | I | 0 | | Number of restart processors (running exclusively for doing restart) | |
| num_prefetch_proc | I | 1 | | Number of processors for prefetching of boundary data asynchronously for a limited area run (running exclusively for reading Input boundary data. Maximum no of processors used for it is limited to 1). | |
| pio_type | I | 1 | | Type of parallel I/O. 1: Classical async I/O processors 2: CDI-PIO (Experimental!) Experimental! | |
| use_icon_comm | L | .FALSE. | | Enable the use of MPI bulk communication through the icon_comm_lib | |
| icon_comm_debug | L | .FALSE. | | Enable debug mode for the icon_comm_lib | |
| max_send_recv- _buffer_size | I | 131072 | | Size of the send/receive buffers for the icon_comm_lib. | |
| use_dp_mpi2io | L | .FALSE. | | Enable this flag if output fields shall be gathered by the output processes in DOUBLE PRECISION. | |
| restart_chunk_size | I | 1 | | (Advanced namelist parameter:) Number of levels to be buffered by the asynchronous restart process. The (asynchronous) restart is capable of writing and communicating more than one 2D slice at once. | |
| num_dist_array_replicas | I | 1 | | (Advanced namelist parameter:) Number of replicas of the distributed array used for the pre_patch. | |
| io_process_stride | I | -1 | | (Advanced namelist parameter:) Stride of processes taking part in reading of data. (Few reading processes, i.e. a large stride, often gives best performance.) | |
| io_process_rotate | I | 0 | | (Advanced namelist parameter:) Rotate of processes taking part in reading of data. (Process taking part if p_pe_work % stride == rotate) | |

Defined and used in: src/namelists/mo_parallel_nml.f90

2.34 psrad_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|-------|
| lradforcing | L(2) | .FALSE. | | switch for diagnostics of aerosol forcing in the solar spectral range (lradforcing(1)) and the thermal spectral range (lradforcing(2)). | |
| lw_gpts_ts | I | 1 | | number of g-points in Monte-Carlo spectral integration for thermal radiation, see lw_spec_samp | |
| lw_spec_samp | I | 1 | | sampling of spectral bands in radiation calculation for thermal radiation lw_spec_samp = 1: standard broad band sampling lw_spec_samp = 2: Monte-Carlo spectral integration (MSCI); lw_gpts_ts randomly chosen g-points per column and radiation call lw_spec_samp = 3: choose g-points not completely randomly in order to reduce errors in the surface radiative fluxes | |
| rad_perm | I | 0 | | integer number that influences the perturbation of the random seed from column to column | |
| sw_gpts_ts | I | 1 | | number of g-points in Monte-Carlo spectral integration for solar radiation, see sw_spec_samp | |
| sw_spec_samp | I | 1 | | sampling of spectral bands in radiation calculation for solar radiation sw_spec_samp = 1: standard broad band sampling sw_spec_samp = 2: Monte-Carlo spectral integration (MSCI); lw_gpts_ts randomly chosen g-points per column and radiation call sw_spec_samp = 3: choose g-points not completely randomly in order to reduce errors in the surface radiative fluxes | |

Defined and used in: src/echam_phy_psradiation/mo_psradiation.f90

2.35 radiation_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--|-------|
| ldiur | L | .TRUE. | | switch for solar irradiation: .TRUE.:diurnal cycle, .FALSE.:zonally averaged irradiation | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------|------|---------|------|--|---------------|
| nmonth | I | 0 | | 0: Earth circles on orbit 1-12: Earth orbit position fixed for specified month | |
| lyr_perp | L | .FALSE. | | .FALSE.: transient Earth orbit following VSOP87 .TRUE.: Earth orbit of year yr_perp of the VSOP87 orbit is perpetuated | |
| yr_perp | L | -99999 | | year used for lyr_perp = .TRUE. | |
| isolrad | I | 0 | | Insolation scheme 0: Use original SRTM insolation. 1: Use insolation from external file containing the spectrally resolved insolation (monthly means) 2: Use preindustrial insolation as in CMIP5 (average from 1844–1856) 3: Use insolation for AMIP-type CMIP5 simulation (average from 1979–1988) 4: Use insolation for RCE-type simulation with $\cos(\text{zenith angle}) = \pi/4$ (with PSRAD: use “4” if the diurnal cycle is switched on) 5: Use insolation for RCE-type simulation with PSRAD if the diurnal cycle is switched off. | |
| izenith | I | 4 | | Choice of zenith angle formula for the radiative transfer computation. 0: Sun in zenith everywhere 1: Zenith angle depends only on latitude 2: Zenith angle depends only on latitude. Local time of day fixed at 07:14:15 for radiative transfer computation ($\sin(\text{time of day}) = 1/\pi$) 3: Zenith angle changing with latitude and time of day 4: Zenith angle and irradiance changing with season, latitude, and time of day (iforcing=inwp only) | |
| islope_rad | I | 0 | | Slope correction for surface radiation: 0: None 1: Slope correction for direct solar radiation without shading effects | |
| albedo_type | I | 1 | | Type of surface albedo 1: based on soil type specific tabulated values (dry soil) 2: MODIS albedo | iforcing=inwp |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|---------|------|--|---|
| direct_albedo | I | 4 | | <p>Direct beam surface albedo. Options mainly differ in terms of their solar zenith angle (SZA) dependency)</p> <p>1: SZA dependency following Ritter-Geleyn; applied to unconditionally all grid points</p> <p>2: SZA dependency following Zaengl (pers. comm.). Same as 1 for water, but for 'rough surfaces' over land the direct albedo is not allowed to exceed the corresponding broadband diffuse albedo.</p> <p>3: SZA dependency following Yang (2008) for snow-free land points. Same as 1 for water/ice and 2 for snow.</p> <p>4: SZA dependency following Briegleb (1992) for snow-free land points. Same as 1 for water/ice and 2 for snow.</p> | <p>iforcing=inwp albedo_type=2</p> |
| icld_overlap | I | 2 | | <p>Method for cloud overlap calculation in shortwave part of RRTM</p> <p>1: maximum-random overlap</p> <p>2: generalized overlap (Hogan, Illingworth, 2000)</p> <p>3: maximum overlap</p> <p>4: random overlap</p> | <p>iforcing=inwp inwp_radiation=1</p> |

| Parameter | Type | Default | Unit | Description | Scope |
|--|------|--|------|---|-------|
| irad_h2o irad_co2 irad_ch4 irad_n2o irad_o3 irad_o2 irad_cfc11 irad_cfc12 | I | 1 2 3 3 0 2 2 2 | | Switches for the concentration of radiative agents irad_xyz = 0: set to zero irad_h2o = 1: vapor, cloud water and cloud ice from tracer variables irad_co2 = 1: CO ₂ from tracer variable irad_co2/ch4/n2o/o2/cfc11/cfc12 = 2: concentration given by vmr_co2/ch4/n2o/o2/cfc11/cfc12 irad_ch4/n2o = 3: tanh-profile with surface concentration given by vmr_ch4/n2o irad_co2/cfc11/cfc12 = 4: time dependent concentration from greenhouse gas file irad_ch4/n2o = 4: time dependent tanh-profile with surface concentration from greenhouse gas file irad_o3 = 2: ozone climatology from MPI irad_o3 = 4: ozone clim for Aqua Planet Exp irad_o3 = 6: ozone climatology with T5 geographical distribution and Fourier series for seasonal cycle for run_nml/forcing = 3 (NWP) irad_o3 = 7: GEMS ozone climatology (from IFS) for run_nml/forcing = 3 (NWP) irad_o3 = 8: ozone climatology for AMIP irad_o3 = 9: MACC ozone climatology (from IFS) for run_nml/forcing = 3 (NWP) irad_o3 = 79: Blending between GEMS and MACC ozone climatologies (from IFS) for run_nml/forcing = 3 (NWP) ; MACC is used over Antarctica irad_o3 = 97: As 79, but MACC is also used above 1 hPa with transition zone between 5 hPa and 1 hPa irad_o3 = 10: Linearized ozone chemistry (ART extension necessary) for run_nml/forcing = 3 (NWP) | |
| vmr_co2 vmr_ch4 vmr_n2o vmr_o2 vmr_cfc11 vmr_cfc12 | R | 348.0e-6 1650.0e-9 306.0e-9 0.20946 214.5e-12 371.1e-12 | | Volume mixing ratio of the radiative agents | |

| Parameter | Type | Default | Unit | Description | Scope |
|--|------|--|------|---|----------------------------|
| fh2o fco2 fch4 fn2o fo3 fo2 fcfc | R | 1. 1. 1. 1. 1. 1. 1. | | Scaling factors for concentrations used in radiation | run_nml/iforcing=2 (ECHAM) |
| irad_aero | I | 2 | | Aerosols 1: prognostic variable 2: global constant 3: externally specified 5: Tanre aerosol climatology for run_nml/iforcing = 3 (NWP) 6: Tegen aerosol climatology for run_nml/iforcing = 3 (NWP) .AND. itopo =1 9: ART online aerosol radiation interaction, uses Tegen for aerosols not chosen to be represented in ART for run_nml/iforcing = 3 (NWP) .AND. itopo =1 .AND. lart=TRUE .AND. iart_ari=1 | |
| lrad_aero_diag | L | .FALSE. | | writes actual aerosol optical properties to output | |
| ighg | I | 0 | | Select dynamic greenhouse gases scenario (read from file) 0 : select default gas volume mixing ratios - 1990 values (CMIP5) 1 : transient CMIP5 scenario from file | run_nml/iforcing=2 (ECHAM) |

Defined and used in: src/namelist/mo_radiation_nml.f90

2.36 run_nml

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------|---------|------|---|-------|
| nsteps | I | -999 | | Number of time steps of this run. Allowed range is ≥ 0 ; setting a value of 0 allows writing initial output (including internal remapping) without calculating time steps. | |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------------|---------|------|--|-------------------|
| dtime | R | 600.0 | s | time step. For real case runs the maximum allowable time step can be estimated as $1.8 \cdot \text{ndyn_substeps} \cdot \overline{\Delta x} \text{ s km}^{-1}$, where $\overline{\Delta x}$ is the average resolution in km and ndyn_substeps is the number of dynamics substeps set in nonhydrostatic_nml. ndyn_substeps should not be increased beyond the default value 5. | |
| ltestcase | L | .TRUE. | | Idealized testcase runs | |
| ldynamics | L | .TRUE. | | Compute adiabatic dynamic tendencies | |
| iforcing | I | 0 | | Forcing of dynamics and transport by parameterized processes. Use positive indices for the atmosphere and negative indices for the ocean. 0: no forcing 1: Held-Suarez forcing 2: ECHAM forcing 3: NWP forcing 4: local diabatic forcing without physics 5: local diabatic forcing with physics -1: MPIOM forcing (to be done) | |
| ltransport | L | .FALSE. | | Compute large-scale tracer transport | |
| ntracer | I | 0 | | Number of advected tracers handled by the large-scale transport scheme | |
| lvert_nest | L | .FALSE. | | If set to .true. vertical nesting is switched on (i.e. variable number of vertical levels) | |
| num_lev | I(max_dom) | 31 | | Number of full levels (atm.) for each domain | lvert_nest=.TRUE. |
| nshift | I(max_dom) | 0 | | vertical half level of parent domain which coincides with upper boundary of the current domain required for vertical refinement, which is not yet implemented | lvert_nest=.TRUE. |
| ltimer | L | .TRUE. | | TRUE: Timer for monitoring the runtime of specific routines is on (FALSE = off) | |
| timers_level | I | 1 | | | |
| activate_sync_timers | L | F | | TRUE: Timer for monitoring runtime of communication routines (FALSE = off) | |
| msg_level | I | 10 | | controls how much printout is written during runtime. For values less than 5, only the time step is written. | |
| msg_timestamp | L | .FALSE. | | If .TRUE., precede output messages by time stamp. | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------|------|-----------------|------|--|----------------|
| test_mode | I | 0 | | Setting a value larger than 0 activates a dummy mode in which time stepping is changed into just doing iterations, and MPI communication is replaced by copying some value from the send buffer into the receive buffer (does not work with nesting and reduced radiation grid because the send buffer may then be empty on some PEs) | iequations = 3 |
| debug_check_level | I | 0 | | Setting a value larger than 0 activates debug checks. | |
| output | C(:) | "nml", "totint" | | <p>Main switch for enabling/disabling components of the model output. One or more choices can be set (as an array of string constants). Possible choices are:</p> <ul style="list-style-type: none"> • "none": switch off all output; • "nml": new output mode (cf. output_nml); • "totint": computation of total integrals. • "maxwinds": write max. winds to separate ASCII file "maxwinds.log". <p>If the output namelist parameter is not set explicitly, the default setting "nml","totint" is assumed.</p> | |
| restart_filename | C | | | File name for restart/checkpoint files (containing keyword substitution patterns <gridfile>, <idom>, <rsttime>, <mtype>). default: "<gridfile>_restart_<mtype>_<rsttime>.nc". | |
| profiling_output | I | 1 | | controls how profiling printout is written: TIMER_MODE_AGGREGATED=1, TIMER_MODE_DETAILED=2, TIMER_MODE_WRITE_FILES=3. | |
| lart | L | .FALSE. | | Main switch which enables the treatment of atmospheric aerosol and trace gases (The ART package of KIT is needed for this purpose) | |
| ldass_lhn | L | .FALSE. | | Main switch which enables the assimilation of radar derived precipitation rate via Latent Heat Nudging | |
| check_uuid_gracefully | L | .FALSE. | | If this flag is set to .TRUE. we give only warnings for non-matching UUIDs. | |

Defined and used in: src/namelists/mo_run_nml.f90

2.37 sleve_nml (relevant if nonhydrostatic_nml:ivctype=2)

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|---------|------|---|-------|
| min_lay_thckn | R | 50 | m | Layer thickness of lowermost layer; specifying zero or a negative value leads to constant layer thicknesses determined by top_height and nlev | |
| max_lay_thckn | R | 25000 | m | Maximum layer thickness below the height given by htop_thcknlimit (NWP recommendation: 400 m) <i>Use with caution! Too ambitious settings may result in numerically unstable layer configurations.</i> | |
| htop_thcknlimit | R | 15000 | m | Height below which the layer thickness does not exceed max_lay_thckn | |
| itype_laydistr | I | 1 | | Type of analytical function used to specify the distribution of the vertical coordinate surfaces 1: transformed cosine, 2: third-order polynomial; in this case, stretch_fac should be less than 1, particularly for large numbers of model levels; the algorithm always works for stretch_fac=0.5 | |
| top_height | R | 23500.0 | m | Height of model top | |
| stretch_fac | R | 1.0 | | Stretching factor to vary distribution of model levels; values <1 increase the layer thickness near the model top | |
| decay_scale_1 | R | 4000 | m | Decay scale of large-scale topography component | |
| decay_scale_2 | R | 2500 | m | Decay scale of small-scale topography component | |
| decay_exp | R | 1.2 | | Exponent of decay function | |
| flat_height | R | 16000 | m | Height above which the coordinate surfaces are flat | |
| lread_smt | L | .FALSE. | | read smoothed topography from file (TRUE) or compute internally (FALSE) | |

Defined and used in: src/namelist/mo_sleve_nml.f90

2.38 synsat_nml¹

This namelist enables the RTTOV library incorporated into ICON for simulating satellite radiance and brightness temperatures. RTTOV is a radiative transfer model for nadir-viewing passive visible, infrared and microwave satellite radiometers, spectrometers and interferometers, see

<https://nwpsaf.eu/deliverables/rtm>

for detailed information.

¹Important note: This feature is currently active for configuration **dwd+cray** only.

| Parameter | Type | Default | Unit | Description | Scope |
|------------|----------------|---------|------|---|-------|
| lsynsat | L (max_dom) | .FALSE. | | Main switch: Enables/disables computation of synthetic satellite imagery for each model domain. | |
| nlev_rttov | I | 51 | | Number of RTTOV levels. | |

Enabling the synsat module makes the following 32 two-dimensional output fields available:

| | | | |
|----------------------|---------------------|----------------------|---------------------|
| SYNMSG_RAD_CL_IR3.9 | SYNMSG_BT_CL_IR3.9 | SYNMSG_RAD_CL_WV6.2 | SYNMSG_BT_CL_WV6.2 |
| SYNMSG_RAD_CL_WV7.3 | SYNMSG_BT_CL_WV7.3 | SYNMSG_RAD_CL_IR8.7 | SYNMSG_BT_CL_IR8.7 |
| SYNMSG_RAD_CL_IR9.7 | SYNMSG_BT_CL_IR9.7 | SYNMSG_RAD_CL_IR10.8 | SYNMSG_BT_CL_IR10.8 |
| SYNMSG_RAD_CL_IR12.1 | SYNMSG_BT_CL_IR12.1 | SYNMSG_RAD_CL_IR13.4 | SYNMSG_BT_CL_IR13.4 |
| SYNMSG_RAD_CS_IR3.9 | SYNMSG_BT_CS_IR3.9 | SYNMSG_RAD_CS_WV6.2 | SYNMSG_BT_CS_WV6.2 |
| SYNMSG_RAD_CS_WV7.3 | SYNMSG_BT_CS_WV7.3 | SYNMSG_RAD_CS_IR8.7 | SYNMSG_BT_CS_IR8.7 |
| SYNMSG_RAD_CS_IR9.7 | SYNMSG_BT_CS_IR9.7 | SYNMSG_RAD_CS_IR10.8 | SYNMSG_BT_CS_IR10.8 |
| SYNMSG_RAD_CS_IR12.1 | SYNMSG_BT_CS_IR12.1 | SYNMSG_RAD_CS_IR13.4 | SYNMSG_BT_CS_IR13.4 |

Here, RAD denotes radiance, BT brightness temperature, CL cloudy, and CS clear sky, supplemented by the channel name.

Defined and used in: `src/namelist/mo_synsat_nml.f90`

2.39 time_nml

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------|------------|------|--|-------|
| calendar | I | 1 | | Calendar type: 0=Julian/Gregorian 1=proleptic Gregorian 2=30day/month, 360day/year | |
| dt_restart | R | 86400.*30. | s | Length of restart cycle in seconds. This namelist parameter specifies how long the model runs until it saves its state to a file and stops. Later, the model run can be resumed, s. t. a simulation over a long period of time can be split into a chain of restarted model runs. Note that the frequency of writing restart files is controlled by <code>io_nml:dt_checkpoint</code> . Only if the value of <code>dt_checkpoint</code> resulting from model default or user's specification is longer than <code>dt_restart</code> , it will be reset (by the model) to <code>dt_restart</code> so that at least one restart file is generated during the restart cycle. If <code>dt_restart</code> is larger than but not a multiple of <code>dt_checkpoint</code> , restart file will <i>not</i> be generated at the end of the restart cycle. | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|-----------------------------|------|--|-------|
| ini_datetime_string | C | '2008- 09-01T 00:00:00Z' | | Initial date and time of the simulation | |
| end_datetime_string | C | '2008- 09-01T 01:40:00Z' | | End date and time of the simulation | |
| is_relative_time | L | .FALSE. | | .TRUE., if time loop shall start with step 0 regardless whether we are in a standard run or in a restarted run (which means re-initialized run). | |

Length of the run If "nsteps" in run_nml is positive, then nsteps*dtime is used to compute the end date and time of the run. Else the initial date and time, the end date and time, dt_restart, as well as the time step are used to compute "nsteps".

2.40 transport_nml (used if run_nml/ltransport=.TRUE.)

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------------|---------|------|--|--|
| lvadv_tracer | L | .TRUE. | | TRUE : compute vertical tracer advection FALSE: do not compute vertical tracer advection | |
| ihadv_tracer | I(ntracer) | 2 | | Tracer specific method to compute horizontal advection: 0: no horiz. transport (note that the specific tracer quantity q is kept constant and not tracer mass ρq) 1: upwind (1st order) 2: Miura (2nd order, linear reconstr.) 3: Miura3 (quadr. or cubic reconstr.) 4: FFSL (quadr. or cubic reconstr.) 5: hybrid Miura3/FFSL (quadr. or cubic reconstr.) 20: miura (2nd order, lin. reconstr.) with subcycling 22: combination of miura and miura with subcycling 32: combination of miura3 and miura with subcycling 42: combination of FFSL and miura with subcycling 52: combination of hybrid FFSL/Miura3 with subcycling Subcycling means that the integration from time step n to $n+1$ is splitted into substeps to meet the stability requirements. For NWP runs, substepping is generally applied above $z = 22$ km (see nonhydrostatic_nml/hbot_qvsubstep). Tracer specific method to compute vertical advection: 0: no vert. transport (note that tracer mass ρq instead of the specific tracer quantity q is kept constant. This differs from the behaviour in horizontal direction!) 1: upwind (1st order) 3: Piecewise parabolic method (PPM): handles $CFL > 1$ 4: Parabolic Spline Method (PSM): (handles $CFL > 1$) 5: Piecewise parabolic method (PPM): GPU-enabled version, handles $CFL > 1$ | lsq_high_ord $\in [2,3]$ lsq_high_ord $\in [2,3]$ lsq_high_ord $\in [2,3]$ |
| ivadv_tracer | I(ntracer) | 3 | | Tracer specific method to compute vertical advection: 0: no vert. transport (note that tracer mass ρq instead of the specific tracer quantity q is kept constant. This differs from the behaviour in horizontal direction!) 1: upwind (1st order) 3: Piecewise parabolic method (PPM): handles $CFL > 1$ 4: Parabolic Spline Method (PSM): (handles $CFL > 1$) 5: Piecewise parabolic method (PPM): GPU-enabled version, handles $CFL > 1$ | lvadv_tracer=TRUE |
| itype_hlimit | I(ntracer) | 4 | | Type of limiter for horizontal transport: | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------|------------|--------------|------|---|--|
| itype_vlimit | I(ntracer) | 1 | | 0: no limiter 3: monotonic flux limiter (FCT) 4: positive definite flux limiter Type of limiter for vertical transport: | |
| ivlimit_selective | I(ntracer) | 0 | | 0: no limiter 1: semi-monotonic reconstruction filter 2: monotonic reconstruction filter 3: positive definite flux limiter Reduce detrimental effect of vertical limiter by applying a method for identifying and avoiding spurious limiting of smooth extrema. | |
| beta_fct | R | 1.005 | | 1: on 0: off global boost factor for range of permissible values $[q_{max}, q_{min}]$ in (semi-) monotonic flux limiter. A value larger than 1 allows for (small) over and undershoots, while a value of 1 gives strict monotonicity (at the price of increased diffusivity). | itype_vlimit=1, 2 itype_hlimit = 3, 4 |
| iadv_tke | I | 0 | | Type of TKE advection 0: no TKE advection 1: vertical advection only 2: vertical and horizontal advection | inwp_turb=1 |
| lstrang | L | .FALSE. | | Time splitting method FALSE: first order Godunov splitting TRUE: second order Strang splitting | |
| tracer_names | C(:) | 'Int2Str(i)' | | Tracer-specific name suffixes. When running idealized cases or the hydrostatic ICON, this variable is used to specify tracer names. If nothing is specified, the tracer name is given as PREFIX+Int2String(i) , where i is the tracer index. Note that this namelist variable has no effect for nonhydrostatic real-case runs, if the NWP- or ECHAM physics packages are switched on. | iforcing \neq inwp, iechem' |
| npassive_tracer | I | 0 | | number of additional passive tracers which have no sources and are transparent to any physical process (no effect). Passive tracers are named Qpassive_ID , where ID is a number between ntracer and ntracer+npassive_tracer . NOTE: By default, limiters are switched off for passive tracers and the scheme 52 is selected for horizontal advection. | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------|---------|------|--|----------------------|
| init_formula | C | ' ' | | Comma-separated list of initialization formulas for additional passive tracers. | npassive_tracer > 0 |
| iord_backtraj | I | 1 | | order of backward trajectory calculation: 1: first order 2: second order (iterative; currently 1 iteration hardcoded; experimental!) | ihadv_tracer='miura' |
| igrad_c_miura | I | 1 | | Method for gradient reconstruction at cell center for 2nd order miura scheme 1: Least-squares (linear, non-consv) 2: Green-Gauss | ihadv_tracer=2 |
| ivcfl_max | I | 5 | | determines stability range of vertical PPM/PSM-scheme in terms of the maximum allowable CFL-number | ivadv_tracer=3,4 |
| llsq_svd | L | .TRUE. | | use QR decomposition (FALSE) or SV decomposition (TRUE) for least squares design matrix A | |
| lclip_tracer | L | .FALSE. | | Clipping of negative values | |

Defined and used in: `src/namelist/mo_advection_nml.f90`

2.41 turbdiff_nml

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------|---------|------|---|-------|
| imode_turb | I | 1 | | Mode of solving the TKE equation for atmosph. layers: 0: diagnostic equation 1: prognostic equation (current version) 2: prognostic equation (intrinsically positive definite) | |
| imode_tran | I | 0 | | Same as <i>imode_turb</i> but only for the transfer layer | |
| icldm_turb | I | 2 | | Mode of water cloud representation in turbulence for atmosph. layers: -1: ignoring cloud water completely (pure dry scheme) 0: no clouds considered (all cloud water is evaporated) 1: only grid scale condensation possible 2: also sub grid (turbulent) condensation considered | |
| icldm_tran | I | 2 | | Same as <i>icldm_turb</i> but only for the transfer layer | |
| q_crit | R | 1.6 | | critical value for normalized super-saturation | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|---------------------------------|
| itype_wcd | I | 2 | | type of water cloud diagnosis within the turbulence scheme: 1: employing a scheme based on relative humidity 2: employing a statistical saturation adjustment | icldm_turb=2 or icldm_tran=2 |
| itype_sher | I | 0 | | Type of shear forcing used in turbulence: 0: only vertical shear of horizontal wind 1: previous plus horizontal shear correction 2: previous plus shear from vertical velocity 3: same as option 1, but (when combined with ltkschs=.TRUE.) scaling of coarse-grid horizontal shear production term with $\frac{1}{\sqrt{Ri}}$ | |
| ltkschs | L | .FALSE. | | Include correction term for coarse grids in horizontal shear production term (needed at non-convection-resolving model resolutions in order to get a non-negligible impact) | itype_sher \geq 1 |
| ltkesso | L | .TRUE. | | Consider TKE-production by sub grid SSO wakes | inwp_sso = 1 |
| imode_tkesso | I | 1 | | mode of calculat. the SSO source term for TKE production: 1: original implementation 2: Ri-dependent reduction factor for Ri>1 | |
| ltkecon | L | .FALSE. | | Consider TKE-production by sub grid convective plumes (inactive) | inwp_conv = 1 |
| ltkschs | L | .FALSE. | | Consider TKE-production by separated horizontal shear eddies (inactive) | |
| ltmpcor | L | .FALSE. | | Consider thermal TKE sources in enthalpy equation | |
| lsflcnd | L | .TRUE. | | Use lower flux condition for vertical diffusion calculation (TRUE) instead of a lower concentration condition (FALSE) | |
| lexpcor | L | .FALSE. | | Explicit corrections of implicitly calculated vertical diffusion of non-conservative scalars that are involved in sub grid condensation processes | |
| tur_len | R | 500.0 | m | Asymptotic maximal turbulent distance ($\kappa * tur_len$ is the integral turbulent master length scale) | |
| pat_len | R | 100.0 | m | Effective length scale of thermal surface patterns controlling TKE-production by sub grid kata/ana-batic circulations. In case of $pat_len = 0$, this production is switched off. | |
| c_diff | R | 0.2 | 1 | Length scale factor for vertical diffusion of TKE. In case of $c_diff = 0$, TKE is not diffused vertically. | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|-------------------|--|----------------|
| a_stab | R | 0.0 | 1 | Factor for stability correction of turbulent length scale. In case of $a_stab = 0$, the turbulent length scale is not reduced for stable stratification. | ltkeshs=.TRUE. |
| a_hshr | R | 0.20 | 1 | Length scale factor for the separated horizontal shear mode. In case of $a_hshr = 0$, this shear mode has no effect. | |
| alpha0 | R | 0.0123 | 1 | Lower bound of velocity-dependent Charnock parameter | |
| alpha0_max | R | 0.0335 | 1 | Upper bound of velocity-dependent Charnock parameter. Setting this parameter to 0.0335 or higher values implies unconstrained velocity dependence | |
| alpha1 | R | 0.75 | 1 | Scaling parameter for molecular roughness of ocean waves | |
| tkhmin | R | 0.75 | m ² /s | Scaling factor for minimum vertical diffusion coefficient (proportional to $Ri^{-2/3}$) for heat and moisture | |
| tkmmin | R | 0.75 | m ² /s | Scaling factor for minimum vertical diffusion coefficient (proportional to $Ri^{-2/3}$) for momentum | |
| tkmmin_strat | R | 4 | m ² /s | Scaling factor for stratospheric minimum vertical diffusion coefficient (proportional to $Ri^{-1/3}$) for momentum, valid above 17.5 km (tropics above 22.5 km) | |
| tkhmin_strat | R | 0.75 | m ² /s | Scaling factor for stratospheric minimum vertical diffusion coefficient (proportional to $Ri^{-1/3}$) for heat and moisture, valid above 17.5 km (tropics above 22.5 km) | |
| itype_synd | I | 2 | | Type of diagnostics of synoptic near surface variables: 1: Considering the mean surface roughness of a grid box 2: Considering a fictive surface roughness of a SYNOP lawn | |
| rlam_heat | R | 1.0 | 1 | Scaling factor of the laminar boundary layer for heat (scalars). The larger rlam_heat, the larger is the laminar resistance. | |
| rat_sea | R | 10.0 | 1 | Ratio of laminar scaling factors for scalars over sea and land. The larger rat_sea, the larger is the laminar resistance for a sea surface compared to a land surface. | |
| tkesmot | R | 0.15 | 1 | Time smoothing factor within [0, 1] for TKE. In case of $tkesmot = 0$, no smoothing is active. | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|---|------------------|
| frcsmot | R | 0.0 | 1 | Vertical smoothing factor within $[0, 1]$ for TKE forcing terms. In case of $frcmot = 0$, no smoothing is active. | lconst_z0=.TRUE. |
| imode_frsmot | I | 1 | | 1 = apply vertical smoothing (if frcsmot>0) uniformly over the globe 2 = restrict vertical smoothing to the tropics (reduces the moist bias in the tropics while avoiding adverse effects on NWP skill scores in the extratropics) | |
| impl_s | R | 1.20 | 1 | Implicit weight near the surface (maximal value) | |
| impl_t | R | 0.75 | 1 | Implicit weight near top of the atmosphere (minimal value) | |
| lconst_z0 | L | .FALSE. | | TRUE: horizontally homogeneous roughness length z0 | |
| const_z0 | R | 0.001 | m | value for horizontally homogeneous roughness length z0 | |
| ldiff_qi | L | .FALSE. | | Turbulent diffusion of cloud ice, if .TRUE. | |
| itype_tran | I | 2 | | type of surface-atmosphere transfer | |
| lprfcor | L | .FALSE. | | using the profile values of the lowest main level instead of the mean value of the lowest layer for surface flux calculations | |
| lnonloc | L | .FALSE. | | nonlocal calculation of vertical gradients used for turbul. diff. | |
| lfreeslip | L | .FALSE. | | .TRUE.: use a free-slip lower boundary condition, i.e. neither momentum nor heat/moisture fluxes (use for idealized runs only!) | |
| lcpfluc | L | .FALSE. | | consideration of fluctuations of the heat capacity of air | |

Defined and used in: `src/namelist/mo_turbdiff_nml.f90`

2.42 upatmo_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|--|---|
| lnontrad | L | .TRUE. | | .TRUE.: Non-traditional terms in horizontal and vertical components of momentum budget (underlined) are switched on (standard for deep atmosphere): $\partial v_n / \partial t + w[v_n / (a + z) - f_t] + \dots = \dots$ $\partial w / \partial t + v_n[-v_n / (a + z) + f_t] +$ $v_t[-v_t / (a + z) - f_n] + \dots = \dots,$ where a is radius of model Earth, $f_{n,t} = 2\Omega \cos(\varphi) \mathbf{e}_\varphi \cdot \mathbf{e}_{n,t}$ are non-traditional Coriolis parameters, with edge-normal and edge-tangential components denoted by n and t, the angular velocity of the model Earth Ω , the latitude φ , and unit vectors \mathbf{e}_{\dots} . | ldeepatmo = .TRUE. |
| lconstgrav | L | .FALSE. | | .FALSE.: gravitational acceleration varies with height (standard for deep atmosphere) .TRUE.: gravitational acceleration is constant (as in case of shallow atmosphere). I.e. underlined factor in gravitational acceleration is set to 1: $\text{grav} = \text{const.} * [a / (a + z)]^2.$ | ldeepatmo = .TRUE. |
| lcentrifugal | L | .FALSE. | | .TRUE.: Explicit centrifugal acceleration is switched on. I.e. underlined terms in horizontal and vertical components of momentum budget are taken into account: $\partial v_n / \partial t + \Omega^2(a + z) \sin(\varphi) \cos(\varphi) \mathbf{e}_\varphi \cdot \mathbf{e}_n + \dots = \dots$ $\partial w / \partial t - \Omega^2(a + z) \cos^2(\varphi) + \dots = \dots.$ (If the factor const. in the gravitational acceleration of the model Earth, $\text{grav} = \text{const.} * [a / (a + z)]^2$, is assumed to be implicitly composed of a purely gravitational part and a centrifugal part, the latter is not subtracted out for lcentrifugal = .TRUE.!) .FALSE.: input fields to the physics parameterizations are computed in accordance with the shallow-atmosphere approximation in any case .TRUE.: input fields to the physics parameterizations are modified for the deep atmosphere. (Please note: the physics parameterizations themselves are not explicitly modified for the deep atmosphere!) | ldeepatmo = .TRUE. |
| ldeepatmo2phys | L | .FALSE. | | .FALSE.: input fields to the physics parameterizations are computed in accordance with the shallow-atmosphere approximation in any case .TRUE.: input fields to the physics parameterizations are modified for the deep atmosphere. (Please note: the physics parameterizations themselves are not explicitly modified for the deep atmosphere!) | ldeepatmo = .TRUE. .AND. iforcing = 2 (ECHAM) |
| expol_start_height | R | 70000 | m | Height above which extrapolation of initial data starts. | itype_vert_expol = 2 |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|---------|------|---|----------------------|
| expol_blending_scale | R | 10000 | m | Vertical distance above expol_start_height within which blending of linearly extrapolated state and climatological state takes place. Scale height of vertically exponentially decaying factor multiplied to the extrapolated horizontal wind (to alleviate stability-endangering wind magnitudes). Exospheric mean reference temperature of the climatology for the extrapolation blending. .TRUE.: Apply some rudimentary sanity check to the extrapolated atmospheric state in the region above expol_start_height (e.g., temperature values everywhere > 0). (Please, apply with care, since it is computationally relatively expensive.) | itype_vert_expol = 2 |
| expol_vn_decay_scale | R | 10000 | m | | itype_vert_expol = 2 |
| expol_temp_infty | R | 400 | K | | itype_vert_expol = 2 |
| lexpol_sanitycheck | L | .FALSE. | | | itype_vert_expol = 2 |

Defined and used in: `src/namelist/mo_upatmo_nml.f90`

3 Ocean-specific namelist parameters

3.1 ocean_physics_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------|------|---------|------|---|-------|
| i_sea_ice | I | 1 | | 0: No sea ice, 1: Include sea ice .FALSE.: compute drag only | |
| richardson_factor_tracer | I | 0.5e-5 | m/s | | |
| richardson_factor_veloc | I | 0.5e-5 | m/s | | |
| l_constant_mixing | L | .FALSE. | | | |

3.2 sea_ice_nml (relevant if run_nml/forcing=2 (ECHAM))

| Parameter | Type | Default | Unit | Description | Scope |
|-------------|------|---------|------|---|--|
| i_ice_therm | I | 2 | | Switch for thermodynamic model: 1: Zero-layer model 2: Two layer Winton (2000) model 3: Zero-layer model with analytical forcing (for diagnostics) 4: Zero-layer model for atmosphere-only runs (for diagnostics) | In an ocean run i_sea_ice must be >=1. In an atmospheric run the ice surface type must be defined. |
| i_ice_dyn | I | 0 | | Switch for sea-ice dynamics: 0: No dynamics 1: FEM dynamics (from AWI) | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|---|
| i_ice_albedo | I | 1 | | Switch for albedo model. Only one is implemented so far. | Defaults to 1 when i_ice_dyn=0 and 2 otherwise. |
| i_Qio_type | I | 2 | | Switch for ice-ocean heat-flux calculation method: 1: Proportional to ocean cell thickness (like MPI-OM) 2: Proportional to speed difference between ice and ocean | |
| kice | I | 1 | | Number of ice classes (must be one for now) | |
| hnull | R | 0.5 | m | Hibler's h_0 parameter for new-ice growth. | |
| hmin | R | 0.05 | m | Minimum sea-ice thickness allowed. | |
| ramp_wind | R | 10 | days | Number of days it takes the wind to reach correct strength. Only used at the start of an OMIP/NCEP simulation (not after restart). | |

4 Namelist parameters for testcases (NAMELIST _ICON)

The ICON model code includes several experiments, so-called test cases, for the shallow water model as well as the 3-dimensional atmosphere. Depending on the specified experiment, initial conditions and boundary conditions are computed internally.

4.1 ha_testcase_nml (Scope: ltestcase=.TRUE. and iequations=[0,1,2] in run_nml)

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------|---------|------|---|--|
| ctest_name | C | 'JWw' | | Name of test case: 'SW_GW': gravity wave 'USBR': unsteady solid body rotation 'Will_2': Williamson test 2 'Will_3': Williamson test 3 'Will_5': Williamson test 5 'Will_6': Williamson test 6 'GW': gravity wave (nlev=20 only!) 'LDF': local diabatic forcing test without physics 'LDF-Moist': local diabatic forcing test with physics initialised with zonal wind field 'HS': Held-Suarez test 'JWs': Jablonowski-Will. steady state 'JWw': Jablonowski-Will. wave test 'JWw-Moist': Jablonowski-Will. wave test including moisture 'APE': aqua planet experiment 'MRW': mountain induced Rossby wave 'MRW2': modified mountain induced Rossby wave | lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.FALSE. lshallow_water=.FALSE. and iforcing=4 lshallow_water=.FALSE., and iforcing=5 lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|-----------|---------|---|--|
| tracer_inidist_list | I(:) | 1 | | 'PA': pure advection 'SV': stationary vortex 'DF1': deformational flow test 1 'DF2': deformational flow test 2 'DF3': deformational flow test 3 'DF4': deformational flow test 4 'RH': Rossby-Haurwitz wave test For a subset of testcases pre-defined initial tracer distributions are available. This namelist parameter specifies the initial distribution for each tracer. In the following the testcases and the pre-defined numbers are given: 'PA': 4,5,6,7,8 'JABW':1,2,3,4 'DF': 5,6,7,8,9 For more details on the initial distributions, please have a look into the code. | lshallow_water=.FALSE. lshallow_water=.FALSE., ntracer = 2 lshallow_water=.FALSE. ha_testcase_nml='PA', 'JABW','DF' |
| rotate_axis_deg | R | 0.0 | deg | Earth's rotation axis pitch angle | ctest_name= 'Will_2', 'Will_3', 'JWs', 'JWw', 'PA', 'DF1234' |
| gw_brunt_vais | R | 0.01 | 1/s | Brunt Vaisala frequency | ctest_name= 'GW' |
| gw_u0 | R | 0.0 | m/s | zonal wind parameter | ctest_name= 'GW' |
| gw_lon_deg | R | 180.0 | deg | longitude of initial perturbation | ctest_name= 'GW' |
| gw_lat_deg | R | 0.0 | deg | latitude of initial perturbation | ctest_name= 'GW' |
| jw_uptb | R | 1.0 | m/s (?) | amplitude of the wave perturbation | ctest_name= 'JWw' |
| mountctr_lon_deg | R | 90.0 | deg | longitude of mountain peak | ctest_name= 'MRW(2)' |
| mountctr_lat_deg | R | 30.0 | deg | latitude of mountain peak | ctest_name= 'MRW(2)' |
| mountctr_height | R | 2000.0 | m | mountain height | ctest_name= 'MRW(2)' |
| mountctr_half_width | R | 1500000.0 | m | mountain half width | ctest_name= 'MRW(2)' |
| mount_u0 | R | 20.0 | m/s | wind speed for MRW cases | ctest_name= 'MRW(2)' |
| rh_wavenum | I | 4 | | wave number | ctest_name= 'RH' |
| rh_init_shift_deg | R | 0.0 | deg | pattern shift | ctest_name= 'RH' |
| ih_s_init_type | I | 1 | | Choice of initial condition for the Held-Suarez test. 1: the zonal state defined in the JW test case; other integers: isothermal state (T=300 K, ps=1000 hPa, u=v=0.) | ctest_name= 'HS' |
| lhs_vn_ptb | L | .TRUE. | | Add random noise to the initial wind field in the Held-Suarez test. | ctest_name= 'HS' |
| hs_vn_ptb_scale | R | 1. | m/s | Magnitude of the random noise added to the initial wind field in the Held-Suarez test. | ctest_name= 'HS' |
| lrh_linear_pres | L | .FALSE. | | Initialize the relative humidity using a linear function of pressure. | ctest_name= 'JWw-Moist','APE', 'LDF-Moist' |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------|------|---|---|
| rh_at_1000hpa | R | 0.75 | | relative humidity 0, 1 at 1000 hPa | ctest_name= 'JWw-Moist', 'APE', 'LDF-Moist' |
| linit_tracer_fv | L | .TRUE. | | Finite volume initialization for tracer fields | ctest_name='PA' |
| ape_sst_case | C | 'sst1' | | SST distribution selection 'sst1': Control experiment 'sst2': Peaked experiment 'sst3': Flat experiment 'sst4': Control-5N experiment 'sst_qobs': Qobs SST distribution exp 'sst_ice': Control SST distribution with -1.8 C above 64 N/S. | ctest_name='APE' |
| ildf_init_type | I | 0 | | Choice of initial condition for the Local diabatic forcing test. 1: the zonal state defined in the JWs test case; other: isothermal state (T=300 K, ps=1000 hPa, u=v=0.) | ctest_name= 'LDF' |
| ldf_symm | L | .TRUE. | | Shape of local diabatic forcing: .TRUE.: local diabatic forcing symmetric about the equator (at 0 N) .FALSE.: local diabatic forcing asym. about the equator (at 30 N) | ctest_name= 'LDF', 'LDF-Moist' |

Defined and used in: `src/testcases/mo_ha_testcases.f90`

4.2 nh_testcase_nml (Scope: ltestcase=.TRUE. and iequations=3 in run_nml)

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|-------|
| nh_test_name | C | 'jabw' | | testcase selection 'zero': no orography 'bell': bell shaped mountain at 0E,0N 'schaer': hilly mountain at 0E,0N 'jabw': Initializes the full Jablonowski Williamson test case. 'jabw_s': Initializes the Jablonowski Williamson steady state test case. 'jabw_m': Initializes the Jablonowski Williamson test case with a mountain instead of the wind perturbation (specify mount_height). 'mrw_nh': Initializes the full Mountain-induced Rossby wave test case. 'mrw2_nh': Initializes the modified mountain-induced Rossby wave test case. | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--|---|
| | | | | <p>'mwbr_const': Initializes the mountain wave with two layers test case. The lower layer is isothermal and the upper layer has constant brunt vaisala frequency. The interface has constant pressure.</p> <p>'PA': Initializes the pure advection test case.</p> <p>'HS_nh': Initializes the Held-Suarez test case. At the moment with an isothermal atmosphere at rest (T=300K, ps=1000hPa, u=v=0, topography=0.0).</p> <p>'HS_jw': Initializes the Held-Suarez test case with Jablonowski Williamson initial conditions and zero topography.</p> <p>'APE_nwp, APE_echam, APE_nh, APEc_nh, ': Initializes the APE experiments. With the jabw test case, including moisture.</p> <p>'wk82': Initializes the Weisman Klemp test case</p> <p>'g_lim_area': Initializes a series of general limited area test cases: itype_atmos_ana determines the atmospheric profile, itype_anaprof_uv determines the wind profile and itype_topo_ana determines the topography</p> <p>'dcmip_bw_11': Initializes (moist) baroclinic instability/wave (DCMIP2016)</p> <p>'dcmip_pa_12': Initializes Hadley-like meridional circulation pure advection test case.</p> <p>'dcmip_rest_200': atmosphere at rest test (Schaer-type mountain)</p> <p>'dcmip_mw_2x': nonhydrostatic mountain waves triggered by Schaer-type mountain</p> <p>'dcmip_gw_31': nonhydrostatic gravity waves triggered by a localized perturbation (nonlinear)</p> <p>'dcmip_gw_32': nonhydrostatic gravity waves triggered by a localized perturbation (linear)</p> <p>'dcmip_tc_51': tropical cyclone test case with 'simple physics' parameterizations (not yet implemented)</p> <p>'dcmip_tc_52': tropical cyclone test case with full physics in Aqua-planet mode</p> <p>'CBL': convective boundary layer simulations for LES package on torus (doubly periodic) grid</p> | <p>l_limited_area = .TRUE.</p> <p>lcoriolis = .FALSE.</p> <p>lcoriolis = .FALSE.</p> <p>l_limited_area = .TRUE. and lcoriolis = .FALSE.</p> <p>lcoriolis = .TRUE.</p> <p>lcoriolis = .TRUE.</p> <p>is_plane_torus= .TRUE.</p> |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|-----------|------|---|---|
| is_toy_chem | L | .FALSE. | | 'lahade': deep-atmosphere sound wave testcase providing comparison of numerical with analytical solution according to method of Laeuter, Handorf and Dethloff, J. Comp. Phys.(2005) (requires to set src/shared/mo_physical_constants: grav to a very small value, e.g. grav = 1.0E-30) Terminator toy chemistry activated when .TRUE. | ldeepatmo = .TRUE. .AND. lcoriolis = .TRUE. .AND. lcentrifugal = .TRUE. |
| tracer_inidist_list | I(:) | 1 | | For a subset of testcases pre-defined initial tracer distributions are available. This namelist parameter specifies the initial distribution for each tracer. In the following the testcases and the pre-defined numbers are given: 'PA': 4,5,6,7,8 'JABW':1,2,3,4 'DF': 5,6,7,8,9 For more details on the initial distributions, please have a look into the code. | nh_test_name='PA', 'JABW','DF' |
| dcmip_bw% | | | | DCMIP2016 baroclinic wave test | 'dcmip_bw_11' |
| deep | I | 0 | | deep atmosphere (1 = yes or 0 = no) | |
| moist | I | 0 | | include moisture, i.e. $qv \neq 0$ (1 = yes or 0 = no) | |
| pertt | I | 0 | | type of initial perturbation (0 = exponential, 1 = stream function) | |
| toy_chem% | | | | terminator toy chemistry | is_toy_chem=.TRUE. |
| dt_chem | R | 300 | s | chemistry tendency update interval | |
| dt_cpl | R | 300 | s | chemistry-transport coupling interval | |
| id_cl | I | 1 | | Tracer container slice index for species CL | |
| id_cl2 | I | 2 | | Tracer container slice index for species CL2 | |
| jw_up | R | 1.0 | m/s | amplitude of the u-perturbation in jabw test case | nh_test_name='jabw' |
| u0_mrw | R | 20.0 | m/s | wind speed for mrw(2) and mwbr_const cases | nh_test_name= 'mrw(2)_nh' and 'mwbr_const' |
| mount_height_mrw | R | 2000.0 | m | maximum mount height in mrw(2) and mwbr_const | nh_test_name= 'mrw(2)_nh' and 'mwbr_const' |
| mount_half_width | R | 1500000.0 | m | half width of mountain in mrw(2), mwbr_const and bell | nh_test_name= 'mrw(2)_nh', 'mwbr_const' and 'bell' |
| mount_lonctr_mrw_deg | R | 90. | deg | lon of mountain center in mrw(2) and mwbr_const | nh_test_name= 'mrw(2)_nh' and 'mwbr_const' |
| mount_latctr_mrw_deg | R | 30. | deg | lat of mountain center in mrw(2) and mwbr_const | nh_test_name= 'mrw(2)_nh' and 'mwbr_const' |
| temp_i_mwbr_const | R | 288.0 | K | temp at isothermal lower layer for mwbr_const case | nh_test_name= 'mwbr_const' |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------|------|----------|-------|--|--|
| p_int_mwbr_const | R | 70000. | Pa | pres at the interface of the two layers for mwbr_const case | nh_test_name= 'mwbr_const' |
| bruntvais_u_mwbr_const | R | 0.025 | 1/s | constant brunt vaissala frequency at upper layer for mwbr_const case | nh_test_name= 'mwbr_const' |
| mount_height | R | 100.0 | m | peak height of mountain | nh_test_name= 'bell' |
| layer_thickness | R | -999.0 | m | thickness of vertical layers | If layer_thickness < 0, the vertical level distribution is read in from externally given HYB_PARAMS_XX. layer_thickness > 0 |
| n_flat_level | I | 2 | | level number for which the layer is still flat and not terrain-following | |
| nh_u0 | R | 0.0 | m/s | initial constant zonal wind speed | nh_test_name= 'bell' |
| nh_t0 | R | 300.0 | K | initial temperature at lowest level | nh_test_name= 'bell' |
| nh_brunt_vais | R | 0.01 | 1/s | initial Brunt-Vaisala frequency | nh_test_name= 'bell' |
| torus_domain_length | R | 100000.0 | m | length of slice domain | nh_test_name= 'bell', lplane=.TRUE. |
| rotate_axis_deg | R | 0.0 | deg | Earth's rotation axis pitch angle | nh_test_name= 'PA' |
| lhs_nh_vn_ptb | L | .TRUE. | | Add random noise to the initial wind field in the Held-Suarez test. | nh_test_name= 'HS_nh' |
| lhs_fric_heat | L | .FALSE. | | add frictional heating from Rayleigh friction in the Held-Suarez test. | nh_test_name= 'HS_nh' |
| hs_nh_vn_ptb_scale | R | 1. | m/s | Magnitude of the random noise added to the initial wind field in the Held-Suarez test. | nh_test_name= 'HS_nh' |
| rh_at_1000hpa | R | 0.7 | 1 | relative humidity at 1000 hPa | nh_test_name= 'jabw', nh_test_name= 'mrw' |
| qv_max | R | 20.e-3 | kg/kg | specific humidity in the tropics | nh_test_name= 'jabw', nh_test_name= 'mrw' |
| ape_sst_case | C | 'sst1' | | SST distribution selection 'sst1': Control experiment 'sst2': Peaked experiment 'sst3': Flat experiment 'sst4': Control-5N experiment 'sst_qobs': Qobs SST distribution exp. 'sst_const': constant SST | nh_test_name='APE_nwp', 'APE_echam' |
| ape_sst_val | R | 29.0 | degC | aqua planet SST for ape_sst_case='sst_const' | nh_test_name= 'APE_nwp', 'APE_echam' |
| limit_tracer_fv | L | .TRUE. | | Finite volume initialization for tracer fields | pure advection tests, only |
| lcoupled_rho | L | .FALSE. | | Integrate density equation 'offline' | pure advection tests, only |
| qv_max_wk | R | 0.014 | Kg/kg | maximum specific humidity near the surface, range 0.012 - 0.016 | nh_test_name='wk82' |
| u_infty_wk | R | 20. | m/s | used to vary the buoyancy zonal wind at infinity height range 0. - 45. | nh_test_name='wk82' |
| bub_amp | R | 2. | K | used to vary the wind shear maximum amplitud of the thermal perturbation | nh_test_name='wk82' |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|-------------------|-------------------|------|---|---|
| bubctr_lat | R | 0. | deg | latitude of the center of the thermal perturbation | nh_test_name='wk82' |
| bubctr_lon | R | 90. | deg | longitude of the center of the thermal perturbation | nh_test_name='wk82' |
| bubctr_z | R | 1400. | m | height of the center of the thermal perturbation | nh_test_name='wk82' |
| bub_hor_width | R | 10000. | m | horizontal radius of the thermal perturbation | nh_test_name='wk82' |
| bub_ver_width | R | 1400. | m | vertical radius of the thermal perturbation | nh_test_name='wk82' |
| itype_atmo_ana | I | 1 | | kind of atmospheric profile: 1 piecewise N constant layers 2 piecewise polytropic layers | nh_test_name= 'g_lim_area' |
| itype_anaprof_uv | I | 1 | | kind of wind profile: 1 piecewise linear wind layers 2 constant zonal wind 3 constant meridional wind | nh_test_name= 'g_lim_area' |
| itype_topo_ana | I | 1 | | kind of orography: 1 schaefer test case mountain 2 gaussian_2d mountain 3 gaussian_3d mountain any other no orography | nh_test_name= 'g_lim_area' |
| nlayers_nconst | I | 1 | | Number of the desired layers with a constant Brunt-Vaisala-frequency | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| p_base_nconst | R | 100000. | Pa | pressure at the base of the first N constant layer | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| theta0_base_nconst | R | 288. | K | potential temperature at the base of the first N constant layer | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| h_nconst | R(nlayers_nconst) | 0., 1500., 12000. | m | height of the base of each of the N constant layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| N_nconst | R(nlayers_nconst) | 0.01 | 1/s | Brunt-Vaisala-frequency at each of the N constant layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| rh_nconst | R(nlayers_nconst) | 0.5 | % | relative humidity at the base of each N constant layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| rhgr_nconst | R(nlayers_nconst) | 0. | % | relative humidity gradient at each of the N constant layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| nlayers_poly | I | 2 | | Number of the desired layers with constant gradient temperature | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 |
| p_base_poly | R | 100000. | Pa | pressure at the base of the first polytropic layer | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|--------------------|------------|------|---|---|
| h_poly | R(nlayers_poly) | 0., 12000. | m | height of the base of each of the polytropic layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 nh_test_name= 'g_lim_area' and itype_atmo_ana=2 nh_test_name= 'g_lim_area' and itype_atmo_ana=2 nh_test_name= 'g_lim_area' and itype_atmo_ana=2 nh_test_name= 'g_lim_area' and itype_anaprof_uv=1 nh_test_name= 'g_lim_area' and itype_anaprof_uv=1 nh_test_name= 'g_lim_area' and itype_anaprof_uv=1 nh_test_name= 'g_lim_area' and itype_anaprof_uv=2,3 nh_test_name= 'g_lim_area' nh_test_name= 'g_lim_area' nh_test_name= 'g_lim_area' and itype_topo_ana=1 nh_test_name= 'g_lim_area' and itype_topo_ana=1,2 nh_test_name= 'g_lim_area' and itype_topo_ana=1 nh_test_name= 'dcmip_mw_2x' |
| t_poly | R(nlayers_poly) | 288., 213. | K | temperature at the base of each of the polytropic layers | |
| rh_poly | R(nlayers_poly) | 0.8, 0.2 | % | relative humidity at the base of each of the polytropic layers | |
| rhgr_poly | R(nlayers_poly) | 5.e-5, 0. | % | relative humidity gradient at each of the polytropic layers | |
| nlayers_linwind | I | 2 | | Number of the desired layers with constant U gradient | |
| h_linwind | R(nlayers_linwind) | 0., 2500. | m | height of the base of each of the linear wind layers | |
| u_linwind | R(nlayers_linwind) | 5, 10. | m/s | zonal wind at the base of each of the linear wind layers | |
| ugr_linwind | R(nlayers_linwind) | 0., 0. | 1/s | zonal wind gradient at each of the linear wind layers | |
| vel_const | R | 20. | m/s | constant zonal/meridional wind (itype_anaprof_uv=2,3) | |
| mount_lonc_deg | R | 90. | deg | longitud of the center of the mountain | |
| mount_latc_deg | R | 0. | deg | latitud of the center of the mountain | |
| schaer_h0 | R | 250. | m | h0 parameter for the schaar mountain | |
| schaer_a | R | 5000. | m | -a- parameter for the schaar mountain, also half width in the north and south side of the finite ridge to round the sharp edges | |
| schaer_lambda | R | 4000. | m | lambda parameter for the schaar mountain | |
| lshear_dcmip | L | FALSE | | run dcmip_mw_2x with/without vertical wind shear FALSE: dcmip_mw_21: non-sheared TRUE : dcmip_mw_22: sheared | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|-----------|----------------|---|---|
| halfwidth_2d | R | 10000. | m | half lenght of the finite ridge in the north-south direction | nh_test_name= 'g_lim_area' and itype_topo_ana=1,2 nh_test_name= 'g_lim_area' and itype_topo_ana=2,3 nh_test_name= 'g_lim_area' and itype_topo_ana=2,3 |
| m_height | R | 1000. | m | height of the mountain | |
| m_width_x | R | 5000. | m | half width of the gaussian mountain in the east-west direction half width in the north-south direction in the rounding of the finite ridge (gaussian_2d) | |
| m_width_y | R | 5000. | m | half width of the gaussian mountain in the north-south direction | nh_test_name= 'g_lim_area' and itype_topo_ana=2,3 |
| gw_u0 | R | 0. | m/s | maximum amplitude of the zonal wind | nh_test_name= 'dcmip_gw_3X' |
| gw_clat | R | 90. | deg | Lat of perturbation center | nh_test_name= 'dcmip_gw_3X' |
| gw_delta_temp | R | 0.01 | K | maximum temperature perturbation | nh_test_name= 'dcmip_gw_32' |
| u_cbl(2) | R | 0:0 | m/s and 1/s | to prescribe initial zonal velocity profile for convective boundary layer simulations where u_cbl(1) sets the constant and u_cbl(2) sets the vertical gradient | nh_test_name=CBL |
| v_cbl(2) | R | 0:0 | m/s and 1/s | to prescribe initial meridional velocity profile for convective boundary layer simulations where v_cbl(1) sets the constant and v_cbl(2) sets the vertical gradient | nh_test_name=CBL |
| th_cbl(2) | R | 290:0.006 | K and K/m | to prescribe initial potential temperature profile for convective boundary layer simulations where th_cbl(1) sets the constant and th_cbl(2) sets the gradient | nh_test_name=CBL |
| lahade%icase | I | 1 | | lahade sub-cases: 1: spherical sound wave (currently the only sub-case) | nh_test_name='lahade' |
| lahade%omega | R | 0 | m/s | Model Earth's angular velocity in units of the velocity the center of the sound wave is advected according to the rotation | |
| lahade%bkg_temp | R | 250 | K | Temperature of background atmosphere | |
| lahade%bkg_pres | R | 100000 | Pa | Pressure of background atmosphere | |
| lahade%ptb_ctr_lat | R | 0 | deg | Center latitude of spherical sound wave perturbation | |
| lahade%ptb_ctr_lon | R | 0 | deg | Center longitude of spherical sound wave perturbation | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------|------|---------|------|---|-------|
| lahade%ptb_ctr_hgt | R | 0.5 | -> | Center height of spherical sound wave perturbation, in units of the model top height [top_height] | |
| lahade%ptb_rad_min | R | 0.04 | -> | Min. radius of spherical shell within which initial perturbation is non-zero, in units of distance from center to model bottom or model top, whichever is shorter [min{ptb_ctr_hgt,(1-ptb_ctr_hgt)} * top_height] | |
| lahade%ptb_rad_max | R | 0.6 | -> | Max. radius of spherical shell ... [min{ptb_ctr_hgt,(1-ptb_ctr_hgt)} * top_height] | |
| lahade%ptb_amp_temp | R | 0.05 | K | Temperature amplitude of initial sound wave perturbation | |
| lahade%ptb_n_rad | R | 1 | 1 | Number of radial wave crests of initial perturbation = (ptb_rad_max - ptb_rad_min) / radial wave length | |
| lahade%output_ptb_var | C | | | Select, if the numerical and analytical solutions of a sound-wave-perturbation-variable shall be output. Currently available variables are: <ul style="list-style-type: none"> "temp": temperature perturbation "rho": density perturbation "pres": pressure perturbation Requirements: the fields extra_3d1 and extra_3d2 will contain the numerical and the analytical solutions, respectively. Both have to be added to the ml_varlist of the output_nml of your choice in combination with inextra_3d = 2. | |

Defined and used in: src/testcases/mo_nh_testcases.f90

5 External data

5.1 extpar_nml (Scope: itopo=1 in run_nml)

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|-------|
| itopo | I | 0 | | 0: analytical topography/ext. data 1: topography/ext. data read from file | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------------|----------|----------|------|---|------------------------|
| itype_vegetation_cycle | I | 1 | | 1: annual cycle of LAI solely based on NDVI climatology 2: additional use of monthly T2M climatology to get more realistic values in extratropics (requires external parameter data containing this field) 3: as 2 with additional coupling of vegetation parameters to T2M bias in transitional seasons (requires DWD assimilation cycle including soil moisture analysis) | |
| n_iter_smooth_topo | I(n_dom) | 0 | | iterations of topography smoother | itopo = 1 |
| fac_smooth_topo | R | 0.015625 | | pre-factor of topography smoother | n_iter_smooth_topo > 0 |
| hgtdiff_max_smooth_topo | R | 0. | m | RMS height difference to neighbor grid points at which the smoothing pre-factor | n_iter_smooth_topo > 0 |
| | | | | fac_smooth_topo reaches its maximum value (linear proportionality for weaker slopes) | |
| heightdiff_threshold | R(n_dom) | 3000. | m | height difference between neighboring grid points above which additional local nabla2 diffusion is applied | |
| lrevert_sea_height | L | .FALSE. | | If .TRUE., sea point heights will be reverted to original (raw data) heights after topography smoothing was applied. | n_iter_smooth_topo > 0 |
| l_emiss | L | .TRUE. | | read and use external surface emissivity map | itopo = 1 |
| extpar_filename | C | | | Filename of external parameter input file, default: "<path>extpar_<gridfile>". May contain the keyword <path> which will be substituted by model_base_dir. | |
| read_nc_via_cdi | L | .FALSE. | | .TRUE.: read NetCDF input data via cdi library .FALSE.: read NetCDF input data using parallel NetCDF library Note: GRIB2 input data is always read via cdi library / GRIB API. For NetCDF input, this switch allows optimizing the input performance, but there is no general rule which option is faster. | |
| extpar_varnames_map_file | C | , , | | Filename of external parameter dictionary, This is a text file with two columns separated by whitespace, where left column: NetCDF name, right column: GRIB2 short name. It is required, if external parameter are read from a file in GRIB2 format. | |

Defined and used in: src/namelist/mo_extpar_nml.f90

6 External packages

7 Information on vertical level distribution

If no vertical sleeve coordinate is chosen (`ivctype / =2`), the hydrostatic and nonhydrostatic models need hybrid vertical level information to generate the terrain following coordinates. The hybrid level specification is stored in `<icon home>/hyb_params/HYB_PARAMS_<nlev>`. The **hydrostatic** model assumes to get **pressure based** coordinates, the **nonhydrostatic** model expects **height based** coordinates. For further information see `<icon home>/hyb_params/README`.

8 Compile flag for mixed precision

To speed up code parts strongly limited by memory bandwidth (primarily the dynamical core and the tracer advection), an option exists to use single precision for variables that are presumed to be insensitive to computational accuracy. This affects most local arrays in the dynamical core routines (`solve_nonhydro` and `velocity_advection`), some local arrays in the tracer transport routines, the metrics coefficients, arrays used for storing tendencies or differenced fields (gradients, divergence etc.), reference atmosphere fields, and interpolation coefficients. Prognostic variables and intermediate variables affecting the accuracy of mass conservation are still treated in double precision. To activate the mixed-precision option, the cpp flags `'-D__MIXED_PRECISION'` and `'-D__MIXED_PRECISION_2'` need to be specified in the configuration settings used for generating the Makefile. The latter flag is used for physics tendencies. Note that interpolation to a latitude-longitude grid is not supported for single-precision variables; if you desire to output physics tendency fields on a regular grid for diagnostic purposes, do not set `'-D__MIXED_PRECISION_2'`.

A Arithmetic expression evaluation

The `mo_expression` module evaluates basic arithmetic expressions specified by character-strings. It is possible to include mathematical functions, operators, and constants. An application of this module is the evaluation of arithmetic expressions provided as namelist parameters.

Besides, Fortran variables can be linked to the expression and used in the evaluation. The implementation supports scalar input variables as well as 2D and 3D fields.

From a users' point of view, the basic usage of this module is described in Section [A.1](#) below. Technically, infix expressions are processed based on a Finite State Machine (FSM) and Dijkstra's shunting yard algorithm. A more detailed description of the Fortran interface is given in Section [A.3](#).

A.1 Examples for arithmetic expressions

Basic examples:

- `sqrt(2.0)`
- `sin(45*pi/180.) * 10 + 5`
- `if(1. > 2, 99, -1.*pi)`
- `min(1,2)`

Variables are used with a bracket notation:

- `sqrt([u]^2 + [v]^2)`

Note that the use of variables requires that these are enabled ("linked") by the Fortran routine that calls the `mo_expression` module.

A.2 Expression syntax

A.2.1 List of functions

| name | #args | description |
|---|-------|---|
| <code>log()</code> , <code>exp()</code> | 1 | natural logarithm and its inverse function. |
| <code>sin()</code> , <code>cos()</code> | 1 | trigonometric functions |
| <code>sqrt()</code> | 1 | square root |
| <code>min()</code> , <code>max()</code> | 2 | minimum and maximum of two values |
| <code>if(<i>value</i>, <i>then</i>, <i>else</i>)</code> | 3 | conditional expression (<i>value</i> > 0.) |

A.2.2 List of operators

| name | evaluates to |
|---|--|
| $a + b$, $a - b$, $a * b$, a / b $a ^ b$ | $(a + b)$, $(a - b)$, $(a * b)$, (a / b) a^b |
| $a > b$ | $\begin{cases} 1, & \text{if } a > b, \\ 0, & \text{otherwise.} \end{cases}$ |
| $a < b$ | $\begin{cases} 1, & \text{if } a < b, \\ 0, & \text{otherwise.} \end{cases}$ |

A.2.3 List of available constants

| name of constant | assigned value | description |
|------------------|-----------------------|---|
| pi | 4 atan(1) | mathematical constant equal to a circle's circumference divided by its diameter |
| r | $6.371229 \cdot 10^6$ | Earth's radius ¹ |

A.3 Usage with Fortran

The minimal Fortran interface is as follows:

1. The `TYPE expression` which is initialized with the character-string that specifies the arithmetic expression.
2. The type-bound procedure `evaluate()`, which returns the result (scalar or array-shaped) as a `POINTER`.
3. The type-bound procedure `link()` connecting a variable to a name in the character-string expression.

A.3.1 Fortran examples

The following examples illustrate the arithmetic expression parser. The calls to `DEALLOCATE` the data structures have been omitted for the sake of brevity:

1. Scalar arithmetic expression:

```
formula = expression("sin(45*pi/180.) * 10 + 5")
CALL formula%evaluate(val)
... use "val" for some purpose ...
```

2. Masking of a 2D array as an example for the link procedure:

```
formula = expression("if([z_sfc] > 2., [z_sfc], 0. )")
CALL formula%link("z_sfc", z_sfc)
CALL formula%evaluate(val_2D)
... use "val_2D(:,:)" for some purpose ...
```

A.3.2 Error handling

Invalid arithmetic expressions yield empty expression objects. When these are evaluated, a `NULL()` pointer is returned. A successful expression evaluation can be tested with the `err_no` variable:

```
IF (formula%err_no == ERR_NONE) THEN
...
END IF
```

In case of error, the `err_no` variable also provides the reason for the aborted evaluation process.

A.4 Remarks

- Variable names are treated case-sensitive!
- For 3D array input it is implicitly assumed that 2D fields are embedded in 3D fields as `"3D(:,level,:) = 2D(:,:)"`.

B Changes incompatible with former versions of the model code

| | |
|------------------------|--|
| <i>Change:</i> | <code>var_names_map_file</code> , <code>out_varnames_map_file</code> |
| <i>Date of Change:</i> | 2013-04-25 |
| <i>Revision:</i> | 12016 |

- Renamed `var_names_map_file` → `output_nml_dict`.
- Renamed `out_varnames_map_file` → `netcdf_dict`.
- The dictionary in `netcdf_dict` is now reversed, s.t. the same map file as in `output_nml_dict` can be used to translate variable names to the ICON internal names and back.

¹This number seems to be based on Hayford's 1910 estimate of the Earth. It is used in ICON as well as MPAS and was almost certainly taken from the Jablonowski and Williamson test case (QJRMS, 2006).

Change: **output_nml: namespace**
Date of Change: **2013-04-26**
Revision: **12051**

- Removed obsolete namelist variable **namespace** from **output_nml**.

Change: **gribout_nml: generatingCenter, generatingSubcenter**
Date of Change: **2013-04-26**
Revision: **12051**

- Introduced new namelist variables **generatingCenter** and **generatingSubcenter**.
- If not set explicitly, center and subcenter information is copied from the input grid file

Change: **radiation_nml: albedo_type**
Date of Change: **2013-05-03**
Revision: **12118**

- Introduced new namelist variable **albedo_type**
- If set to 2, the surface albedo will be based on the MODIS data set.

Change: **initicon_nml: dwdinc_filename**
Date of Change: **2013-05-24**
Revision: **12266**

- Renamed dwdinc_filename to dwdana_filename

Change: **initicon_nml: l_ana_sfc**
Date of Change: **2013-06-25**
Revision: **12582**

- Introduced new namelist flag **l_ana_sfc**
- If true, soil/surface analysis fields are read from the analysis field dwdfg_filename. If false, surface analysis fields are not read. Soil and surface are initialized with the first guess instead.

Change: **new_nwp_phy_tend_list: output names consistent with variable names**
Date of Change: **2013-06-25**
Revision: **12590**

- temp_tend_radlw → ddt_temp_radlw
- temp_tend_turb → ddt_temp_turb
- temp_tend_drag → ddt_temp_drag

Change: prepicon_nml, remap_nml, input_field_nml
Date of Change: 2013-06-25
Revision: 12597

- Removed the sources for the "prepicon" binary!
- The "prepicon" functionality (and most of its code) has become part of the ICON tools.

Change: initicon_nml
Date of Change: 2013-08-19
Revision: 13311

- The number of vertical input levels is now read from file. The namelist parameter **nlev_in** has become obsolete in r12700 and has been removed.

Change: parallel_nml
Date of Change: 2013-10-14
Revision: 14160

- The namelist parameter **exch_msgsize** has been removed together with the option **iorder_sendrecv=4**.

Change: parallel_nml
Date of Change: 2013-08-14
Revision: 14164

- The namelist parameter **use_sp_output** has been replaced by an equivalent switch **use_dp_mpi2io** (with an inverse meaning, i.e. we have **use_dp_mpi2io = .NOT. use_sp_output**).

Change: parallel_nml
Date of Change: 2013-08-15
Revision: 14175

- The above-mentioned namelist parameter **use_dp_mpi2io** got the default **.FALSE**. By this, the output data are sent now in single precision to the output processes.

Change: **initicon_nml: l_ana_sfc**
Date of Change: **2013-10-21**
Revision: **14280**

- The above-mentioned namelist parameter **l_ana_sfc** has been replaced by **lread_ana**. The default is set to `.TRUE.`, meaning that analysis fields are required and read on default. With `lread_ana=.FALSE.` ICON is able to start from first guess fields only.

Change: **output_nml: lwrite_ready, ready_directory**
Date of Change: **2013-10-25**
Revision: **14391**

- The namelist parameters **lwrite_ready** and **ready_directory** have been replaced by a single namelist parameter **ready_file**, where `ready_file /= 'default'` enables writing ready files.
- Different `output_nml`'s may be joined together to form a single ready file event – they share the same `ready_file`.

Change: **output_nml: output_bounds**
Date of Change: **2013-10-25**
Revision: **14391**

- The namelist parameter **output_bounds** specifies a start, end, and increment of output intervals. It does no longer allow multiple triples.

Change: **output_nml: steps_per_file**
Date of Change: **2013-10-30**
Revision: **14422**

- The default value of the namelist parameter **steps_per_file** has been changed to `-1`.

Change: **run_nml**
Date of Change: **2013-11-13**
Revision: **14759**

- The dump/restore functionality for domain decompositions and interpolation coefficients has been removed from the model code. This means, that the parameters
 - `ldump_states`,
 - `lrestore_states`,
 - `ldump_dd`,
 - `lread_dd`,

- nproc_dd,
- dd_filename,
- dump_filename,
- l_one_file_per_patch

have been removed together with the corresponding functionality from the ICON model code.

Change: **output_nml: filename_format**
Date of Change: **2013-12-02**
Revision: **15068**

- The string token <ddhhmmss> is now substituted by the *relative* day-hour-minute-second string, whereas the absolute date-time stamp can be inserted using <datetime>.

Change: **output_nml: ready_file**
Date of Change: **2013-12-03**
Revision: **15081**

- The ready file name has been changed and may now contain string tokens <path>, <datetime>, <ddhhmmss> which are substituted as described for the namelist parameter filename_format.

Change: **interpl_nml: rbf_vec_scale_ll**
Date of Change: **2013-12-06**
Revision: **15156**

- The real-valued namelist parameter **rbf_vec_scale_ll** has been removed.
- Now, there exists a new integer-valued namelist parameter, **rbf_scale_mode_ll** which specifies the mode, how the RBF shape parameter is determined for lon-lat interpolation.

Change: **io_nml**
Date of Change: **2013-12-06**
Revision: **15161**

- Removed remaining vlist-related namelist parameter. This means that the parameters
 - out_filetype
 - out_expname
 - dt_data
 - dt_file
 - lwrite_dbldprec, lwrite_decomposition, lwrite_vorticity, lwrite_divergence, lwrite_pres, lwrite_z3, lwrite_tracer, lwrite_tend_phy, lwrite_radiation, lwrite_precip, lwrite_cloud, lwrite_tke, lwrite_surface, lwrite_omega, lwrite_initial, lwrite_oce_timestepping

are no longer available.

Change: **gridref_nml**
Date of Change: **2014-01-07**
Revision: **15436**

- Changed namelist defaults for nesting: `grf_intmethod_e`, `l_mass_consvcorr`, `l_density_nudging`.

Change: **interpol_nml**
Date of Change: **2014-02-10**
Revision: **16047**

- Changed namelist default for `rbf_scale_mode_ll`: The RBF scale factor for lat-lon interpolation is now determined automatically by default.

Change: **echam_phy_nml**
Date of Change: **2014-02-27**
Revision: **16313**

- Replace the logical switch `lcover` by the integer switch `icover` that is used in ECHAM-6.2. Values are transferred as follows: `.FALSE.` = 1 (=default), `.TRUE.` = 2.

Change: **turbdiff_nml**
Date of Change: **2014-03-12**
Revision: **16527**

- Change constant minimum vertical diffusion coefficients to variable ones proportional to $1/\sqrt{Ri}$ for `inwp_turb` = 10; at the same time the defaults for `tkhmin` and `tkmmin` are increased from 0.2 m²/s to 0.75 m²/s.

Change: **nwp_phy_nml**
Date of Change: **2014-03-13**
Revision: **16560**

- Removed namelist parameter `dt_ccov`, since practically it had no effect. For the quasi-operational NWP-setup, the calling frequency of the cloud cover scheme is the same as that of the convection scheme. I.e. both are synchronized.

Change: **nwp_phy_nml**
Date of Change: **2014-03-24**
Revision: **16668**

- Changed namelist default for **itype_z0**: use land cover related roughness only (itype_z0=2).

Change: **nonhydrostatic_nml**
Date of Change: **2014-05-16**
Revision: **17293**

- Removed switch for vertical TKE advection in the dynamical core (**lvadv_tke**). TKE advection has been moved into the transport scheme and can be activated with **iadv_tke=1** in the **transport_nml**.

Change: **nonhydrostatic_nml**
Date of Change: **2014-05-27**
Revision: **17492**

- Removed namelist parameter **model_restart_info_filename** in namelist **master_model_nml**.

Change: **transport_nml**
Date of Change: **2014-06-05**
Revision: **17654**

- Changed namelist default for **itype_hlimit** from monotonous limiter (3) to positive definite limiter (4).

Change: **nh_pzlev_nml**
Date of Change: **2014-08-28**
Revision: **18795**

- Removed namelist **nh_pzlev_nml**. Instead, each output namelist specifies its separate list of **p_levels**, **h_levels**, and **i_levels**.

Change: **nonhydrostatic_nml**
Date of Change: **2014-10-27**
Revision: **19670**

- Removed namelist parameter **l_nest_rcf** in namelist **nonhydrostatic_nml**.

Change: **nonhydrostatic_nml**
Date of Change: **2014-11-24**
Revision: **20073**

- Removed namelist parameter **iadv_rcf** in namelist **nonhydrostatic_nml**. The number of dynamics substeps per advective step are now specified via **ndyn_substeps**. The meaning of **run_nml:dttime** has changed and denotes the advective time step.

Change: **io_nml**
Date of Change: **2015-03-25**
Revision: **21501**

- Namelist parameter `lzaxis_reference` is deprecated and has no effect anymore. However, users are not forced to modify their scripts instantaneously: `lzaxis_reference=.FALSE.` is still a valid namelist setting, but it has no effect and a warning will be issued. `lzaxis_reference` finally removed in `r24606`.

Change: **limarea_nml**
Date of Change: **2016-02-08**
Revision: **26390**

- Namelist parameter `dt_latbc` has been removed. Its value is now identical to the namelist parameter `dtime_latbc`.

Change: **interpol_nml**
Date of Change: **2016-02-11**
Revision: **26423**

- Namelist parameter `l_intp_c2l` is deprecated and has no effect anymore.

Change: **lnd_nml**
Date of Change: **2016-07-21**
Revision: **28536**

- The numbering of the various options for `sstice_mode` has changed. Former option 2 became 3, former option 3 became 4, and former option 4 became 5. This was necessary, because a new option was introduced (option 2).

Change: **initicon_nml**
Date of Change: **2016-07-22**
Revision: **28556**

- Namelist parameter `latbc_varnames_map_file` has been moved to the namelist `limarea_nml`.

Change: **transport_nml**
Date of Change: **2016-09-22**
Revision: **29339**

- Namelist parameter `niter_fct` has been removed, since the functionality of iterative flux correction is no longer available.

Change: **initicon_nml**
Date of Change: **2016-10-07**
Revision: **29484**

- Namelist parameter `l_sst_in` has been removed. In case of `init_mode=2` (IFSINIT), sea points are now initialized with SST, if provided in the input file. Otherwise sea points are initialized with the skin temperature. The possibility to use the skin temperature despite having the SST available has been dropped.

Change: **initicon_nml**
Date of Change: **2016-12-14**
Revision: **62288ed77b2975182204a2ec6fa210a3fb1ad8a7**

- Namelist parameters `ana_varlist`, `ana_varlist_n2` have been renamed to `check_ana(jg)%list`, with `jg` indicating the patch ID.

Change: **initicon_nml**
Date of Change: **2017-01-27**
Revision: **ae1be66f**

- The default value of the namelist parameter `num_prefetch_proc` has been changed to 1, i.e. asynchronous read-in of lateral boundary data is now enabled.

Change: **interpol_nml**
Date of Change: **2017-01-31**
Revision: **e1c56104**

- With the introduction of the namelist parameter `lreduced_nestbdry_stencil` in the namelist `interpol_nml` the nest boundary points are no longer removed from lat-lon interpolation stencil by default.

Change: **limarea_nml**
Date of Change: **2017-03-14**
Revision: **631b731627**

- The namelist parameter `nlev_latbc` is now deprecated. Information about the vertical level number is taken directly from the input file.

Change: **echam_phy_nml / mpi_phy_nml**
Date of Change: **2017-04-19**
Revision: **icon-aes:icon-aes-mag 9ecee54f69108716308029d8d7aa0296c343a3c2**

- The namelist `echam_phy_nml` is replaced by the namelist `mpi_phy_nml`, which extends the control to multiple domains and introduces time control in terms of start and end date/time [`sd_prc`,`ed_prc`] and time interval `dt_prc` for individual atmospheric processes *prc*.

Change: mpi_phy_nml / echam_phy_nml and mpi_sso_nml / echam_sso_nml
Date of Change: 2017-11-22
Revision: icon-aes:icon-aes-cfgnml f84219511329281d441d81923fe97ce1d7ecf007

- The namelists, configuration variables and related modules are renamed from ...mpi_phy... to ...echam_phy... because programmers felt that the acronym "mpi" for "Max Planck Institute" in relation to physics cannot be distinguished from "mpi" for "Message Passing Interface" as used in the parallelization.

Change: gw_hines_nml / echam_gwd_nml
Date of Change: 2017-11-24
Revision: icon-aes:icon-aes-cfgnml 699346b5d318d53be215e0b8e8b5ba8631d44c48

- The namelists gw_hines_nml is replaced by the namelist echam_gwd_nml, which extends the control to multiple domains.

Change: vdiff_nml / echam_vdf_nml
Date of Change: 2017-11-27
Revision: icon-aes:icon-aes-cfgnml f1dec0a0d3b8ec506861975cd59a729fe43fdf8e

- The namelists vdiff_nml is replaced by the namelist echam_vdf_nml, which additionally includes tuning parameters for the total turbulent energy scheme, and extends the control to multiple domains.

Change: echam_conv_nml / echam_cnv_nml
Date of Change: 2017-11-29
Revision: icon-aes:icon-aes-cfgnml 099c40f88dbaae6c7cc79ea878e5862847ef7e27

- The namelists echam_conv_nml is replaced by the namelist echam_cnv_nml, which extends the control to multiple domains.

Change: echam_cloud_nml / echam_cld_nml
Date of Change: 2017-12-04
Revision: icon-aes:icon-aes-cfgnml afacc102a87b03f78ff47ad0b7af8f348bacef6f

- The namelists echam_cloud_nml is replaced by the namelist echam_cld_nml, which extends the control to multiple domains.

Change: psrad_orbit_nml / radiation_nml / echam_rad_nml
Date of Change: 2017-12-12
Revision: icon-aes:icon-aes-cfgnml 8da087238b81183c337a3b1ae81d2b2e3dafdba8

- For controlling the input of ECHAM physics to the PSrad scheme, the namelists `psrad_orbit_nml` and `radiation_nml` are replaced by the namelist `echam_rad_nml`, which extends the control to multiple domains. For controlling the input of NWP physics to the RRTMG radiation, the `radiation_nml` namelist remains valid. The `psrad_orbit_nml` namelist, which is not used for RRTMG radiation, is deleted.