ICON Namelist Overview

May 11, 2011

Contents

1 ICON Namelists

1.1 Scripts, Namelist files and Programs

Run scripts starting the programs for the grid generation and 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/.

Table 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_OCEAN_GRID	Gen. ocean grid	create_ocean_grid.run	grid_command
NAMELIST_TORUS_GRID	Gen. torus grid	create_torus_grid.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 for grid generation

2.1 Namelist parameters defining the atmosphere grid

2.1.1 graph_ini (NAMELIST_GRAPH)

Parameter	Type	Default	Unit	Description	Scope
nroot	I	2		root subdivision of initial edges	
grid_levels	I	4		number of edge bisections following the root subdivision	
lplane	L	.FALSE.		switch for generating a double periodic planar grid. The root level consists of 8 triangles.	

Defined and used in: src/grid_generator/mo_io_graph.f90

2.1.2 grid_ini (NAMELIST_GRID)

Parameter	Type	Default	Unit	Description	Scope
nroot	I	2		root subdivision of initial edges	
grid_levels	I	4		number of edge bisections following the root subdivision	
lplane	L	.FALSE.		switch for generating planar grid. The root level consists of 8 triangles.	

Defined and used in: src/grid_generator/mo_io_grid.f90

2.1.3 grid_options (NAMELIST_GRID)

x_rot_angle	R	0.0	deg	Rotation of the icosahedron about the x-axis (connecting the origin and [0°E, 0°N])	

Parameter	Type	Default	Unit	Description	Scope
y_rot_angle	R	0.0	deg	Rotation of the icosahedron about the y-axis (connecting the origin and [90°E, 0°N), done after the rotation about the x-axis.	
z_rot_angle	R	0.0	deg	rotation of the icosahedron about the z-axis (connecting the origin and [0°E, 90°N), done after the rotation about the y-axis.	
itype_optimize	I	4		Grid optimization type 0: no optimization 1: Heikes Randall 2: equal area 3: c-grid small circle 4: spring dynamics	
l_c_grid	L	.FALSE.		C-grid constraint on last level	
maxlev_optim	I	100		Maximum grid level where the optimization is applied	i_type_optimize = 1 or 4
beta_spring	R	0.90		tuning factor for target grid length	i_type_optimize = 4

Defined and used in: src/grid_generator/mo_io_grid.f90

2.1.4 plane_options (NAMELIST_GRID)

tria_arc_km	R	10.0	km	length of triangle edge on plane	lplane=.TRUE.

The number of grid points is generated by root level section and further bisections. The double periodic root level consists of 8 triangles. The spatial coordinates are $-1 <= x <= 1$, and $-\sqrt{3}/2 <= y <= \sqrt{3}/2$. Currently the planar option can only be used as an f-plane.
Defined and used in: src/grid_generator/mo_io_grid.f90

2.1.5 gridref_ini (NAMELIST_GRIDREF)

Parameter	Type	Default	Unit	Description	Scope
grid_root	I	2		root subdivision of initial edges	
start_lev	I	4		number of edge bisections following the root subdivision	
n_dom	I	2		number of logical model domains, including the global one	
n_phys_dom	I	n_dom		number of physical model domains, may be larger than n_dom (in this case, domain merging is applied)	
parent_id	I(n_phys_dom-1)	i		ID of parent domain (first entry refers to first nested domain; needs to be specified only in case of more than one nested domain per grid level)	
logical_id	I(n_phys_dom-1)	i+1		logical grid ID of parent domain (first entry refers to first nested domain; needs to be specified only in case of domain merging, i.e. n_dom < n_phys_dom)	
l_plot	L	.FALSE.		produces GMT plots showing the locations of the nested domains	
l_circ	L	.TRUE.		Create circular (.T.) or rectangular (.F.) refined domains	
l_rotate	L	.FALSE.		Rotates center point into the equator in case of l_circ = .FALSE.	lcirc=.FALSE.
write_hierarchy	I	1		0: Output only computational grids 1: Output in addition parent grid of global model domain (required for computing physics on a reduced grid) 2: Output all grids back to level 0 (required for hierarchical search algorithms)	

Parameter	Type	Default	Unit	Description	Scope
bdy_indexing_depth	I	max_rlcell (=8)		Number of cell rows along the lateral boundary of a model domain for which the refin_ctrl fields contain the distance from the lateral boundary; needs to be enlarged when lateral boundary nudging is required for one-way nesting	
radius	R(n_dom-1)	30.	deg	radius of nested domain (first entry refers to first nested domain; needs to be specified for each nested domain separately)	lcirc=.TRUE.
hwidth_lon	R(n_dom-1)	20.	deg	zonal half-width of refined domain (first entry refers to first nested domain; needs to be specified for each nested domain separately)	lcirc=.FALSE.
hwidth_lat	R(n_dom-1)	20.	deg	meridional half-width of refined domain (first entry refers to first nested domain; needs to be specified for each nested domain separately)	lcirc=.FALSE.
center_lon	R(n_dom-1)	90.	deg	center longitude of refined domain (first entry refers to first nested domain; needs to be specified for each nested domain separately)	
center_lat	R(n_dom-1)	30.	deg	center latitude of refined domain (first entry refers to first nested domain; needs to be specified for each nested domain separately)	

Defined and used in: src/grid_generator/mo_gridrefinement.f90

2.2 Namelist parameters defining the ocean grid (NAMELIST_OCEAN_GRID)

The ocean grids are created by the script `run/create_ocen_grid.run`

2.2.1 grid_geometry_conditions

no_of_conditions	I	0		Number of geometric conditions	
patch_shape	I(no_of_conditions)	0		1=rectangle; 2=circle	
patch_center_x	R(no_of_conditions)	0.0	degrees	longitude of patch center	
patch_center_y	R(no_of_conditions)	0.0	degrees	latitude of patch center	
rectangle_xradious	R(no_of_conditions)	0.0	degrees	half meridional extension of a rectangular patch	patch_shape=1
rectangle_yradious	R(no_of_conditions)	0.0	degrees	half zonal extension of a rectangular patch	patch_shape=1
circle_radious	R(no_of_conditions)	0.0	degrees	radius of a circular patch	patch_shape=2

Defined in `mo_grid_conditions.f90`

2.2.2 create_ocean_grid

only_get_sea_land_mask	L	.false.		.true.:returns the whole grid with a sea-land mask; .false.:returns only the ocean grid	
smooth_ocean_boundary	L	.true.		.true.:smooths the ocean boundaries so no triabgle has two boundary edges; .false.:no smoothing	
input_file	C			name of the input grid file	
elevation_file	C			name of the file containing cell elevation values for the input_file	no_of_conditions=0
elevation_field	C			name of the field containing the cell elevation values	no_of_conditions=0
min_sea_depth	R	0.0	m (negative)	if cell elevation < min_sea_depth then the cell is consider sea	
set_sea_depth	R	0.0	m (negative)	if not 0, then sea cells are of set_sea_depth elevation	
set_min_sea_depth	R	0.0	m (negative)	if not 0, then sea cells have a maximum of set_min_sea_depth elevation	
edge_elev_interp_method	I	2		compute edge elevation from cells using: linear interpolation=1; min value = 2	
output_refined_ocean_file	C			name of the output refined ocean grid file	

Defined in `mo_create_ocean_grid.f90`

2.3 Namelist parameters defining the torus grid (NAMELIST_TORUS_GRID)

2.3.1 torus_grid_parameters

y_no_of_rows	I		4	number of triangle rows of the torus grid	
x_no_of_columns	I		8	number of triangle columns of the torus grid	
edge_length	R	m	1000.0	the triangle edge length	
x_center	R	m	0.0	the x coordinate of the torus center	
y_center	R	m	0.0	the y coordinate of the torus center	
out_file_name	C			the torus grid file name	
unfolded_torus_file_name	C			the unfolded torus grid file name (for plotting)	
ascii_filename	C			the unfolded torus grid ascci file name (for plotting)	

Defined in `mo_create_torus_grid.f90`. See the run script `run/create_torus_grid.run`.

3 Namelist parameters defining the ICON model (NAMELIST_ICON)

3.1 grid_ctl

Parameter	Type	Default	Unit	Description	Scope
nroot	I	2		root subdivision of initial edges	
start_lev	I	4		coarsest bisection level	
n_dom	I	1		number of model domains, 1 = global domain only	
parent_id	I(n_dom-1)	i		ID of parent domain (first entry refers to first nested domain; needs to be specified only in case of more than one nested domain per grid level) MUST be the same as in gridref_ini	n_dom>1

Parameter	Type	Default	Unit	Description	Scope
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
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 or 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
lplane	L	.FALSE.		planar option	
corio_lat	R	0.0	deg	Center of the f-plane is located at this geographical latitude	lplane=.TRUE.
lpatch0	L	.FALSE.		If set to .true. an additional patch one level below the root patch is allocated and read so that physics calculations on a coarser grid are possible	
lredgrid_phys	L	.FALSE.		If set to .true. is calculated on a reduced grid (= one grid level higher); requires lpatch0=.TRUE.	

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

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

3.2 parallel_ctl

Parameter	Type	Default	Unit	Description	Scope
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 2: use METIS	
p_test_run	L	.FALSE.		.TRUE. means verification run for MPI parallelization (PE 0 processes full domain)	
l_test_openmp	L	.FALSE.		if .TRUE. is combined with p_test_run=.TRUE. and OpenMP paralllelization, the test PE gets only 1 thread in order to verify the OpenMP paralllelization	
l_log_checks	L	.FALSE.		if .TRUE. messages are generated during each synchonization step (use for debugging only)	
l_fast_sum	L	.FALSE.		if .TRUE., use fast (not processor-configuration-invariant) global summation	
iorder_sendrecv	I	1		Sequence of send/receive calls: 1 = irecv/send; 2 = isend/recv; 3 = isend/irecv	

Parameter	Type	Default	Unit	Description	Scope
itype_comm	I	1		1: use local memory for exchange buffers 2: use global memory for exchange buffers 3: asynchronous halo communication for dynamical core (NH tria only)	
num_io_procs	I	0		Number of I/O processors (running exclusively for doing I/O)	
pio_type	I	1		Type of parallel I/O. Only used if number of I/O processors greater number of domains. Experimental!	

Defined and used in: src/shared/mo_parallel_ctl.f90

3.3 run_nml

Parameter	Type	Default	Unit	Description	Scope
nproma	I	1		chunk length	
nlev	I	31		Number of full levels (atm. or ocean)	
num_nlev	I(n_dom)	31		Number of full levels (atm.) for each domain	lvert_nest=.TRUE.
nshift	I(n_dom)	0		vertical half level of parent domain which coincides with upper boundary of the current domain	lvert_nest=.TRUE.
lvert_nest	L	.FALSE.		If set to .true. vertical nesting is switched on (i.e. variable number of vertical levels)	
ntracer	I	0		number of tracers	
dtime	R	600.0	s	time step Std. time step for ICOHAM R2B04 = 600s	
calendar	I	1		Calendar type:	

Parameter	Type	Default	Unit	Description	Scope
				0=Julian/Gregorian 1=proleptic Gregorian 2=30day/month,360day/year	
				The initial date and time of this run is preset to 2008-09-01 / 00:00:00 UT	
ini_year	I	2008			
ini_month	I	9			
ini_day	I	1			
ini_hour	I	0	hr		
ini_minute	I	0	min		
ini_second	R	0.0	s		
				The end date and time of this run is preset to 2008-09-01 / 01:40:00 UT, which is 10 steps of 600 s after the initial date and time	
end_year	I	2008			
end_month	I	9			
end_day	I	1			
end_hour	I	1	hr		
end_minute	I	40	min		
end_second	R	0.0	s		
				Length of the run If "nsteps" is positive, then nsteps*dtime is used to compute the end date and time of the run. Else if any of the "run_..." variables is positive, then these are used to compute the end date and time and, using dtime, also "nsteps". Else the initial date and time and the end date and time and the time step are used to compute "nsteps".	
nsteps	I	0		number of time steps of this run.	

Parameter	Type	Default	Unit	Description	Scope
run_day run_hour run_minute run_second	I I I R	0 0 0 0.0		number of whole days of the run number of additional hours of the run number of additional minutes of the run number of additional seconds of the run	
iequations	I	1		Equations and prognostic variables. Use positive indices for the atmosphere and negative indices for the ocean. 0: shallow water model 1: hydrostatic atmosphere, T 2: hydrostatic atm., θ·dp 3: non-hydrostatic atmosphere -1: hydrostatic ocean	
i_cell_type	I	3		Cell type 3: triangular cells 4: quadrilateral cells (to be done) 6: pentagonal/hexagonal cells	
ldynamics	L	.TRUE.		Read namelist 'dynamics_ctl' and compute adiabatic dynamic tendencies	
ltransport	L	.FALSE.		Read namelist 'transport_ctl' and compute tracer tendencies by transport	

Parameter	Type	Default	Unit	Description	Scope
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)	
ltestcase	L	.TRUE.		Idealized testcase runs	
lcorio	L	.TRUE.		Coriolis force	
itopo	I	0		topography (not yet implemented)	
msg_level	I	10		controls how much printout is written during runtime. For values less than 5, only the time step is written.	
inextra_2d	I	0		Number of 2D Fields for diagnostic/debugging output.	iequations = 3 (to be done for 1, 2)
inextra_3d	I	0		Number of 3D Fields for diagnostic/debugging output.	iequations = 3 (to be done for 1, 2)
ltimer	L	.TRUE.		TRUE: Timer for monitoring thr runtime of specific routines is on (FALSE = off)	
ldump_states	L	.FALSE.		Dump patch/interpolation/grid refinement state of every patch (after subdivision in case of a parallel run) to a Netcdf file and exit program.	
lrestore_states	L	.FALSE.		Restore patch/interpolation/grid refinement states from NetCDF dump files instead of calculating them.	

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

3.4 dynamics_ctl (used if run_nml/ldynamics=.TRUE.)

Parameter	Type	Default	Unit	Description	Scope
idiv_method	I	1		Method for divergence computation: 1: Standard Gaussian integral. Hydrostatic atm. model: for unaveraged normal components, Non-hydrostatic atm. model: for averaged normal components 2: bilinear averaging of divergence	i_cell_type=3
divavg_cntrwgt	R	0.5		Weight of central cell for divergence averaging	idiv_method= 2
lsi_3d	L	.FALSE.		3D GMRES solver or decomposistion into 2D problems	lshallow_water=.FALSE. and itime_scheme=4
itime_scheme	I	4		Time integration scheme: 1: pure advection (no dynamics) 2: 2 time level semi implicit (not yet implemented) 3: 3 time level explicit 4: 3 time level with semi implicit correction 5: standard 4th-order Runge-Kutta method (4-stage) 6: SSPRK(5,4) scheme (5-stage) 3: same as default, but computation of velocity tendencies in corrector step only 4: Matsuno scheme 6: same as default, but usage of velocity tendencies at (nnow+nnew)/2	 ieqations=1 or 2 iequations=1 or 2 iequations=1 or 2 iequations=1 or 2 iequations=1 or 2 iequations=3 and i_cell_type=3 iequations=3 and i_cell_type=3 iequations=3 and i_cell_type=3

Parameter	Type	Default	Unit	Description	Scope
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= 3 or 4
asselin_coeff	R	0.1		Asselin filter coefficient	itime_scheme= 3 or 4
si_2tls	R	0.6		weight of time step n+1. Valid range: [0,1]	itime_scheme=2
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=2
si_cmin	R	30.0	m/s	semi implicit correction is done for eigenmodes with speeds larger than si_cmin	itime_scheme=4 and lsi_3d=.FALSE.
si_coeff	R	1.0		weight of the semi implicit correction	itime_scheme=4
si_offctr	R	0.7		future implict weight for si correction	itime_scheme=4
si_rtol	R	1.0e-3		relative tolerance for GMRES solver	itime_scheme=4
ldry_dycore	L	.TRUE.		Assume dry atmosphere	iequations∈{1,2}
lref_temp	L	.FALSE.		Set a background temperature profile as base state	iequations∈{1,2}

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

3.5 nonhydrostatic_ctl (used if run_nml/iequations=3)

Parameter	Type	Default	Unit	Description	Scope
rayleigh_coeff	R(n_dom)	0.05		Rayleigh damping coefficient (Klemp, Dudhia, Hassiotis: MWR136, pp.3987-4004)	i_cell_type=3
damp_height	R(n_dom)	17500.0	m	Height in which Rayleigh damping starts	

Parameter	Type	Default	Unit	Description	Scope
htop_moist_proc	R	200000.0	m	Height above which moist physics and advection of cloud and precipitation variables are turned off	
k2_updamp_coeff	R	2.0e6		enhanced 2nd order diffusion coefficient in upper damping layer	i_cell_type=6, hdiff_order=3 (Smagorinski)
vwind_offctr	R	0.05		Off-centering in vertical wind solver	i_cell_type=3
ivctype	I	1		Type of vertical coordinate: 1: Gal-Chen hybrid 2: SLEVE (uses sleve_ctl)	
iadv_rcf	I	1		reduced calling frequency (rcf) for transport 1: no rcf (every dynamics-step) 2: transport every 2. step 4: . . .	
l_nest_rcf	L	.TRUE.		Synchronize interpolation/feedback calls with advection (transport) time steps. l_nest_rcf is automatically reset to .FALSE. if iadv_rcf=1	i_cell_type=3
l_masscorr_nest	L	.FALSE.		Apply mass conservation correction also in nested domain	i_cell_type=3
iadv_rhotheta	I	2		Advection method for rho and rhotheta: 1: centred differences horiz. + vert. 2: 2nd order Miura horizontal 3: 3rd order Miura horizontal (not recommended)	i_cell_type=3

Parameter	Type	Default	Unit	Description	Scope
igradp_method	I	1		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	i_cell_type=3
l_zdiffu_t	L	.FALSE.		.TRUE.: Compute Smagorinsky temperature diffusion truly horizontally over steep slopes	i_cell_type=3 .AND. hdiff_order=5 .AND. lhdiff_temp = .true.
thslp_zdiffu	R	0.025		Slope threshold above which truly horizontal temperature diffusion is activated	i_cell_type=3 .AND. hdiff_order=5 .AND. lhdiff_temp=.true. .AND. l_zdiffu_t=.true.
thhgtd_zdiffu	R	200	m	Threshold of height difference between neighboring grid points above which truly horizontal temperature diffusion is activated (alternative criterion to thslp_zdiffu)	i_cell_type=3 .AND. hdiff_order=5 .AND. lhdiff_temp=.true. .AND. l_zdiffu_t=.true.
exner_expol	R	0.5		Temporal extrapolation (fraction of dt) of Exner function for computation of horizontal pressure gradient	i_cell_type=3
l_open_ubc	L	.FALSE.		.TRUE.: Use open upper boundary condition (rather than w=0) to better conserve sea-level pressure in the presence of diabatic heating	i_cell_type=3
ltheta_up_hori	L	.FALSE.		upstream biased horizontal advection for theta (see also upstr_beta)	i_cell_type=6

Parameter	Type	Default	Unit	Description	Scope
upstr_beta	R	1.0		Selection of order for horiz. theta advection: 3rd order=1.0, 4th order=0.0	i_cell_type=6
gmres_rtol_nh	R	1.0e-6		relative tolerance for convergence in gmres solver	i_cell_type=6
l_impl_vert_adv	L	.TRUE.		implicit vertical advection for horizontal velocity and potential temperature (Implicit vertical advection for ρ and w is automatically included in the new 5band matrix solver for divergent modes. For theta, one of Daniels schemes is envisaged for the future.)	i_cell_type=6

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

3.6 sleve_ctl (used if nonhydrostatic_ctl/ivctype=2)

Parameter	Type	Default	Unit	Description	Scope
min_lay_thckn	R	50	m	Layer thickness of lowermost layer	
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	

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

3.7 diffusion_ctl

Parameter	Type	Default	Unit	Description	Scope
lhdiff_temp	L	.TRUE.		Diffusion on the temperature field	
lhdiff_vn	L	.TRUE.		Diffusion on the horizontal wind field	
hdiff_order	I	4 2		Order of ∇ operator for diffusion: -1: no diffusion 2: ∇^2 diffusion 3: Smagorinsky ∇^2 diffusion for the hexagonal model (includes frictional heating if lhdiff_temp=.TRUE.) 4: ∇^4 diffusion 5: Smagorinsky ∇^2 diffusion combined with ∇^4 background diffusion as specified via hdiff_efdt_ratio defaults: 2 for hexagonal model, 4 for triangular model 24 or 42: $\nabla 2$ diffusion from model top to a certain level (cf. k2_pres_max and k2_klev_max below); ∇^4 for the lower levels.	24 and 42 currently allowed only in the hydrostatic atm model (run_nml/iequation = 1 or 2).
k2_pres_max	R	-99.	Pa	Pressure level above which ∇^2 diffusion is applied.	hdiff_order = 24 or 42, and run_nml/iequation = 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 run_nml/iequation = 1 or 2.

Parameter	Type	Default	Unit	Description	Scope
hdiff_efdt_ratio	R	1.0		ratio of e-folding time to time step (or 2* time step when using a 3 time level time stepping scheme) (only for triangles currently)	
hdiff_tv_ratio	R	1.0		Ratio of diffusion coefficients for temperature and normal wind: $T : v_n$	
hdiff_multfac	R	1.0		Multiplication factor of normalized diffusion coefficient for nested domains	n_dom>1
hdiff_smag_fac	R	0.15		Scaling factor for Smagorinsky diffusion	for triangles only with iequations=3, for hexagons with hdiff_order=3

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

3.8 transport_ctl (used if run_nml/ltransport=.TRUE.)

Parameter	Type	Default	Unit	Description	Scope
ihadv_tracer	I(ntracer)	2 4		Horiz. transport scheme: 0: no horiz. transport 1: upwind (1st order) 2: miura (2nd order, lin. reconstr.) 3: miura3 (quadr. or cubic reconstr.) 4: up3 (3rd or 4th order upstream)	 if i_cell_type=3 lsq_high_ord $\in$ [2,3] if i_cell_type=6
ivadv_tracer	I(ntracer)	3		Vert. transport scheme: 0: no vert. transport 1: upwind (1st order) 2: muscl_cfl (2nd order, handles CFL > 1) 20: muscl (2nd order) 3: ppm_cfl (3^{rd} order, handles CFL > 1)	

Parameter	Type	Default	Unit	Description	Scope
				30: ppm (3rd order)	
lvadv_tracer	L	.TRUE.		calculate vertical tracer advection	
lstrang	L	.FALSE.		splitting into fractional steps - second order Strang splitting (.TRUE.) - first order Godunov splitting (.FALSE.)	
ctracer_list	C	”		list of tracer names	
itype_hlimit	I(ntracer)	3 4		Type of limiter for horizontal transport: 0: no limiter 1: semi-monotonous slope limiter 2: monotonous slope limiter 3: monotonous flux limiter 4: positive definite flux limiter	 ihadv_tracer='miura' ihadv_tracer='miura' ihadv_tracer='miura[3]' ihadv_tracer='miura[3]', 'iup3[4]'
itype_vlimit	I(ntracer)	1		Type of limiter for vertical transport: 0: no limiter 1: semi-monotone slope limiter 2: monotonous slope limiter 4: positive definite flux limiter	
iord_backtraj	I	1		order of backward trajectory calculation: 1: first order 2: second order (iterative; currently 1 iteration hardcoded)	 ihadv_tracer='miura'
igrad_c_miura	I	1		Method for gradient reconstruction at cell center for 2nd order miura 1: Least-squares (linear, non-consv) 2: Green-Gauss 3: gradient reconstruction (RBF) at cell center on the basis of normal gradients at edges	 ihadv_tracer=2

Parameter	Type	Default	Unit	Description	Scope
lclip_tracer	L	.FALSE.		Clipping negative values	
upstr_beta_adv	R	1.0		parameter to select 3rd order (=1) or 4th order (=0) advection, or something inbetween (0..1)	ihadv_tracer=iup3
ivcfl_max	I	5		determines stability range of vertical PPM-scheme in terms of the maximum allowable CFL-number	ivadv_tracer=3

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

3.9 nwp_phy_ctl

Parameter	Type	Default	Unit	Description	Scope
inwp_gscp	I	0		cloud microphysics and precipitation 0: none 1: hydci (COSMO-EU microphysics)	run_nml/iforcing = inwp
inwp_convection	I	0		convection 0: none 1: Tiedtke/Bechtold convection	run_nml/iforcing = inwp
inwp_cldcover	I	1		cloud cover scheme for radiation 0: no clouds (only QV) 1: grid-scale clouds and QV 2: clouds from COSMO turbulence scheme 3: clouds from COSMO SGS cloud scheme	run_nml/iforcing = inwp
inwp_radiation	I	0		radiation 0: none 1: RRTM radiation 2: Ritter-Geleyn radiation	run_nml/iforcing = inwp

Parameter	Type	Default	Unit	Description	Scope
inwp_satad	I	1		saturation adjustment 0: none 1:	run_nml/iforcing = inwp
inwp_turb	I	0		vertical diffusion and transfer 0: none 1: COSMO diffusion and transfer 2: ECHAM diffusion	run_nml/iforcing = inwp
inwp_sso	I	0		subgrid scale orographic drag 0: none 1:	run_nml/iforcing = inwp
inwp_surface	I	0		surface scheme 0: none 1:	run_nml/iforcing = inwp
dt_conv	R (max_dom)	600.	seconds	time interval of convection call currently each subdomain has the same value	run_nml/iforcing = inwp
dt_rad	R (max_dom)	1800.	seconds	time interval of radiation call currently each subdomain has the same value	run_nml/iforcing = inwp
dt_sso	R (max_dom)	3600.	seconds	time interval of sso call currently each subdomain has the same value	run_nml/iforcing = inwp
dt_ccov	R (max_dom)	dt_conv	seconds	time interval of cloud cover call currently each subdomain has the same value	run_nml/iforcing = inwp currently is not used
dt_gscp	R (max_dom)	iadv_rcf * dtime	seconds	time interval of gscp call each subdomain it is halved	run_nml/iforcing = inwp not recomended to change
dt_satad	R (max_dom)	iadv_rcf * dtime	seconds	time interval of satad call each subdomain it is halved	run_nml/iforcing = inwp not recomended to change

Parameter	Type	Default	Unit	Description	Scope
dt_turb	R (max_dom)	dt_gscp	seconds	time interval of turb call each subdomain it is halved	run_nml/iforcing = inwp not recomended to change
dt_radheat	R (max_dom)	dt_satad	seconds	time interval of radheat call each subdomain it is halved	run_nml/iforcing = inwp not recomended to change

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

3.10 radiation_nml

Parameter	Type	Default	Unit	Description	Scope
ldiur	L	.TRUE.		switch for solar irradiation: .TRUE.:diurnal cycle, .FALSE.:zonally averaged irradiation	
nmonth	I	0		0: Earth circles on orbit 1-12: Earth orbit position fixed for specified month	
yr_perp	L	-99999		year used for lyr_perp = .TRUE.	
lyr_perp	L	.FALSE.		.FALSE.: transient Earth orbit following VSOP87 .TRUE.: Earth orbit of year yr_perp of the VSOP87 orbit is perpertuated	
dt_rad	R	7200.	second	time interval of full radiation computation	run_nml/iforcing = iecham

Parameter	Type	Default	Unit	Description	Scope
izenith	I	3 4 (for iforcing = inwp)		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(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)	
irad_h2o irad_co2 irad_ch4 irad_n2o irad_o3 irad_o2 irad_cfc11 irad_cfc12 irad_aero	I	1 2 3 3 3 2 2 2 2		Switches for the concentration of radiative agents 0: 0. 1: prognostic variable 2: global constant 3: externally specified irad_aero = 5: aerosol climatology for run_nml/iforcing = 3 (NWP) when inwp_radiation = 2 irad_o3 = 6: ozone climatology with T5 geographical distribution and Fourier series for seasonal cycle for run_nml/iforcing = 3 (NWP)	Note: until further notice, please use irad_h2o = 1 irad_co2 = 2 and 0 for all the other agents for run_nml/iforcing = 2 (ECHAM).

Parameter	Type	Default	Unit	Description	Scope
vmr_co2 vmr_ch4 vmr_n2o vmr_o2 vmr_cfc11 vmr_cfc12	R	353.9e-6 1693.6e-9 309.5e-9 0.20946 252.8e-12 466.2e-12		Volume mixing ratio of the radiative agents	

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

3.11 nwp_lnd_nml

Parameter	Type	Default	Unit	Description	Scope
nlev_soil	I	7		number of soil layers	
nlev_snow	I	1		number of snow layers for `lmulti_snow=.true.`	
nsfc_subs	I	1		number of tiles	
nztlev	I	2		used time integration scheme	
lmulti_snow	L	.FALSE.		.TRUE. for use of multi-layer snow model	
lseaice	L	.FALSE.		.TRUE. for use of sea-ice model	
llake	L	.FALSE.		.TRUE. for use of lake model	

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

3.12 echam_phy_nml

Parameter	Type	Default	Unit	Description	Scope
lrad	L	.TRUE.		Switch on radiation.	iforcing = 2
lvdiff	L	.TRUE.		Switch on turbulent mixing (i.e. vertical diffusion).	iforcing = 2

Parameter	Type	Default	Unit	Description	Scope
lconv	L	.TRUE.		Switch on cumulus convection.	iforcing = 2
lcond	L	.TRUE.		Switch on large scale condensation.	iforcing = 2
lcover	L	.FALSE.		.TRUE. for prognostic cloud cover scheme, .FALSE. for diagnostic scheme.	iforcing = 2 Note: lcover = .TRUE. runs, but has not been evaluated (yet) in ICON.
llandsurf	L	.FALSE.		.TRUE. for surface exchanges	iforcing = 2 Not implemeted yet
lssodrag	L	.FALSE.		.TRUE. for subgrid scale orographic drag	iforcing = 2 Not implemeted yet
lagwdrag	L	.FALSE.		.TRUE. for atmospheric gravity wave drag	iforcing = 2 Not implemeted yet
lice	L	.FALSE.		.TRUE. for sea-ice temperature calculation	iforcing = 2 Not implemeted yet
lmeltpond	L	.FALSE.		.TRUE. for calculation of meltponds	iforcing = 2 Not implemeted yet
lmlo	L	.FALSE.		.TRUE. for mixed layer ocean	iforcing = 2 Not implemeted yet
lhd	L	.FALSE.		.TRUE. for hydrologic discharge model	iforcing = 2 Not implemeted yet
lmidatm	L	.FALSE.		.TRUE. for middle atmosphere model version	iforcing = 2 Not implemeted yet

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

3.13 echam_conv_ctl

Parameter	Type	Default	Unit	Description	Scope
lmfpen	L	.TRUE.		Switch on penetrative convection.	iforcing = 2 .AND. lconv = .TRUE.
lmfmid	L	.TRUE.		Switch on midlevel convection.	iforcing = 2 .AND. lconv = .TRUE.

Parameter	Type	Default	Unit	Description	Scope
lmfscv	L	.TRUE.		Switch on shallow convection.	iforcing = 2 .AND. lconv = .TRUE.
lmfdd	L	.TRUE.		Switch on cumulus downdraft.	iforcing = 2 .AND. lconv = .TRUE.
lmfdudv	L	.TRUE.		Switch on cumulus friction.	iforcing = 2 .AND. lconv = .TRUE.
iconv	I	1		Choice of cumulus convection scheme. 1: Nordeng scheme 2: Tiedtke scheme 3: hybrid scheme	iforcing = 2 .AND. lconv = .TRUE.
cmftau	R	10800.		Characteristic convective adjustment time scale.	iforcing = 2 .AND. lconv = .TRUE.
cmfctop	R	0.3		Fractional convective mass flux (valid range [0,1]) across the top of cloud	iforcing = 2 .AND. lconv = .TRUE.
cprcon	R	1.0e-4		Coefficient for determining conversion from cloud water to rain.	iforcing = 2 .AND. lconv = .TRUE.
cminbuoy	R	0.025		Minimum excess buoyancy.	iforcing = 2 .AND. lconv = .TRUE.
entrpen	R	1.0e-4		Entrainment rate for penetrative convection.	iforcing = 2 .AND. lconv = .TRUE.
dlev	R	3.e4	Pa	Critical thickness necessary for the onset of convective precipitation.	iforcing = 2 .AND. lconv = .TRUE.
nauto	I	1		autoconversion scheme: 1: Beheng (1994) 2: Khairoutdinov and Kogan (2000)	iforcing = 2 .AND. lconv = .TRUE.
lconvmassfix	L	.FALSE.		aerosol mass fixer in convection	iforcing = 2 .AND. lconv = .TRUE.

Defined and used in: src/atm_phy_echam/mo_echam_conv_parameters.f90

3.14 echam_vdiff_ctl

Parameter	Type	Default	Unit	Description	Scope
lsfc_mon_flux	L	.TRUE.		Switch on surface momentum flux.	lvdiff = .TRUE.
lsfc_heat_flux	L	.TRUE.		Switch on surface sensible and latent heat flux.	lvdiff = .TRUE.

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

3.15 io_ctl

Parameter	Type	Default	Unit	Description	Scope
out_expname	C	'IIIEEEETTTT'		Outfile basename	
lwrite_omega	L	.TRUE.		write out vertical velocity in pressure coords.	lshallow_water=.FALSE.
lwrite_pres	L	.TRUE.		write out full level pressure	lshallow_water=.FALSE.
lwrite_vorticity	L	.TRUE.		write out averaged vorticity at cells	
lwrite_divergence	L	.TRUE.		write out divergence at cells	
lwrite_z3	L	.TRUE.		write out geopotential on full levels	lshallow_water=.FALSE.
lwrite_tracer	L(ntracer)	.TRUE.		write out tracer at cells	
out_filetype	I	2		Type of output format: 1: GRIB1 (not yet implemented) 2: netCDF	
dt_data	R	21600.0	s	Output time increment	
dt_diag	R	dtime		diagnostic integral output timestep	
dt_file	R	2592000	s	Time increment of triggering new output file	
lkeep_in_sync	L	.FALSE.		Sync output stream with file on disk after each timestep	
lwrite_tend_phy	L	.TRUE. .FALSE. (Scope)		Physics induced tendencies.	.TRUE. if iforcing=iecham .FALSE. else
lwrite_radiation	L	.TRUE. .FALSE. (Scope)		Radiation related fields.	.TRUE. if iforcing=iecham or inwp .FALSE. else

Parameter	Type	Default	Unit	Description	Scope
lwrite_precip	L	.TRUE. .FALSE. (Scope)		Precipitation	.TRUE. if iforcing=iecham or inwp .FALSE. else
lwrite_cloud	L	.TRUE. .FALSE. (Scope)		Cloud variables	.TRUE. if iforcing=iecham or inwp .FALSE. else
lwrite_tke	L	.TRUE.		TKE	.FALSE. per Default .TRUE. if set (AND iforcing =nwp)
lwrite_surface	L	.FALSE.		surface variables	.TRUE. if set (AND iforcing =nwp) .FALSE. else
lwrite_extra	L	.FALSE.		debug fields	.TRUE. if inextra_2d /_3d > 0 .FALSE. else

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

3.16 interpol_ctl

Parameter	Type	Default	Unit	Description	Scope
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_v	I	1		Kernel type for reconstruction at vertices: 1: Gaussian	

Parameter	Type	Default	Unit	Description	Scope
				3: inverse multiquadric	
rbf_vec_scale_c	R(n_dom)	resolution-dependent		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	
nudge_max_coeff	R	0.02		Maximum relaxation coefficient for lateral boundary nudging	
nudge_efold_width	R	2.5		e-folding width (in units of cell rows) for lateral boundary nudging coefficient	
nudge_zone_width	I	8		Total width (in units of cell rows) for lateral boundary nudging zone	
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 2: quadratic 30: cubic (no 3^{rd} order cross deriv.) 3: cubic	
i_cori_method	I	3		Selector for tangential wind reconstruction method 1: Almut's method for tangential wind, but PV usage as in TRSK 2: method of Thuburn, Ringler, Skamarock and Klemp (TRSK) 3: Almut's method for tangential wind and PV usage	currently only for i_cell_type=6

Parameter	Type	Default	Unit	Description	Scope
l_corner_vort	L	.TRUE.		switch whether the rhombus averaged corner vorticity is averaged to the hexagon (.TRUE.) or the rhombi are directly averaged to the hexagon (.FALSE.)	i_cori_method=3

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

3.17 gridref_ctl

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	4		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	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}):	n_dom>1

Parameter	Type	Default	Unit	Description	Scope
				1: area-weighted averaging 2: bilinear interpolation	
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	0.5		RBF scale factor for grid refinement (edges)	n_dom>1
denom_diffu_t	R	135		Deniminator for lateral boundary diffusion of temperature	n_dom>1
denom_diffu_v	R	200		Deniminator for lateral boundary diffusion of velocity	n_dom>1

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

3.18 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.

3.18.1 testcase_ctl (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	 lshallow_water=.TRUE.

Parameter	Type	Default	Unit	Description	Scope
				'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 initalised 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 '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	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. lshallow_water=.FALSE. lshallow_water=.FALSE., ntracer = 2 lshallow_water=.FALSE.
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'

Parameter	Type	Default	Unit	Description	Scope
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 pertubation	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'
ihs_init_type	I	1		Choice of initial condition for the Held-Suarez test. 1: the zonal state defined in the JWs 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'
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	ctest_name='APE'

Parameter	Type	Default	Unit	Description	Scope
				'sst1': Control experiment 'sst2': Peaked experiment 'sst3': Flat experiment 'sst4': Control-5N experiment 'sst_qobs': Qobs SST distribution exp.	
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_hydro_testcases.f90

3.18.2 nh_testcase_ctl (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.	

Parameter	Type	Default	Unit	Description	Scope
				'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. '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. 'PA': Initializes the pure advection test case. '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). 'HS_jw': Initializes the Held-Suarez test case with Jablonowski Williamson initial conditions and zero topography. 'APE_nh': Initializes the APE experiments. At the moment with T=300K, ps=1013.25hPa, u=v=w=0).	
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 case	nh_test_name= 'mrw(2)_nh'

Parameter	Type	Default	Unit	Description	Scope
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.	degrees	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.	degrees	lat of mountain center in mrw(2) and mwbr_const	nh_test_name= 'mrw(2)_nh' and 'mwbr_const'
u0_mwbr_const	R	20.0	m/s	wind speed for mwbr_const case	nh_test_name= 'mwbr_const'
temp_i_mwbr_const	R	288.0	K	temp at isothermal lower layer for mwbr_const case	nh_test_name= 'mwbr_const'
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.
n_flat_level	I	2		level number for which the layer is still flat and not terrain-following	layer_thickness >0
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.

Parameter	Type	Default	Unit	Description	Scope
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.	nh_test_name='APE_nh'
linit_tracer_fv	L	.TRUE.		Finite volume initialization for tracer fields	ctest_name='PA'

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

4 Externally provided data

4.1 ext_par_ctl (Scope: itopo=1 in run_nml)

Parameter	Type	Default	Unit	Description	Scope
n_iter_smooth_topo	I	35		iterations of topography smoother	itopo = 1
fac_smooth_topo	R	0.015625		pre-factor of topography smoother	n_iter_smooth_topo > 0

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

4.2 Information on vertical level distribution

The hydrostatic and nonhydrostatic models need hybrid vertical level information to generate the terrain following coorindates. 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.