ICON Namelist Overview

April 22, 2025

Contents

1. ICON Namelists **2**
1.1. Scripts, Namelist files and Programs . . . 2
1.2. Namelist parameters . . . 3

2. Namelist parameters defining the atmospheric model **3**
2.1. aes_bubble_nml . . . 3
2.2. aes_cop_nml . . . 4
2.3. aes_cov_nml . . . 5
2.4. aes_phy_nml . . . 5
2.5. aes_rad_nml . . . 6
2.6. aes_vdf_nml . . . 10
2.7. aes_wmo_nml . . . 11
2.8. assimilation_nml . . . 11
2.9. ccycle_nml . . . 15
2.10. cloud_mig_nml . . . 16
2.11. coupling_mode_nml . . . 16
2.12. diffusion_nml . . . 17
2.13. dynamics_nml . . . 19
2.14. ensemble_pert_nml . . . 20
2.15. gribout_nml . . . 25
2.15.1. Notes on the GRIB library backward compatiblity adjustment: . . . 28
2.16. grid_nml . . . 29
2.17. gridref_nml . . . 32
2.18. initicon_nml . . . 34
2.19. interpol_nml . . . 42
2.20. io_nml . . . 44
2.20.1. Restart read/write mode: . . . 49

2.20.2. Some notes on the output of optional diagnostics: 50
2.21. les_nml (parameters for LES turbulence scheme; valid for inwp_turb=5) 53
2.22. limarea_nml (Scope: l_limited_area=.TRUE. in grid_nml) 55
2.23. lnd_nml 57
2.24. ls_forcing_nml (parameters for large-scale forcing; valid for torus geometry; is_plane_torus=.TRUE.) 63
2.25. master_nml 64
2.26. master_model_nml (repeated for each model) 64
2.27. master_time_control_nml 65
2.28. meteogram_output_nml 66
2.29. nonhydrostatic_nml 67
2.30. nudging_nml 71
2.31. nwp_phy_nml 76
2.31.1. Notes on use of stochastic convection schemes 83
2.32. nwp_tuning_nml 84
2.33. twomom_mcrph_nml 89
2.34. output_nml (relevant if run_nml/output='nml') 98
2.35. parallel_nml 107
2.36. radiation_nml (relevant if run_nml:iforcing=3 (NWP)) 110
2.37. run_nml 118
2.38. scm_nml (relevant if l_scm_mode) 121
2.39. sleve_nml (relevant if nonhydrostatic_nml:ivctype=2) 122
2.40. sppt_nml 124
2.41. synsat_nml 124
2.42. synradar_nml 125
2.43. time_nml 127
2.44. transport_nml (used if run_nml/ltransport=.TRUE.) 129
2.45. turb_vdiff_nml 131
2.46. turbdiff_nml 133
2.47. upatmo_nml 138

3. Ocean-specific namelist parameters 145
3.1. ocean_physics_nml 145
3.2. sea_ice_nml (relevant if run_nml/iforcing=2 (ECHAM)) 145

4. Ocean waves specific namelist parameters 146
4.1. energy_propagation_nml (used if run_nml/ltransport=.TRUE.) 146
4.2. initwave_nml 146
4.3. wave_nml 147

5. Namelist parameters for testcases (NAMELIST_ICON) 149
5.1. nh_testcase_nml (Scope: ltestcase=.TRUE. in run_nml) 149

6. External data 156
6.1. extpar_nml (Scope: itopo=1 in run_nml) . . . 156

7. Serialization 158

8. External packages 160
8.1. Community Interface (ComIn) . . . 160

9. Information on vertical level distribution 161

10. Compile flag for mixed precision 161

A. Arithmetic expression evaluation 162
A.1. Examples for arithmetic expressions . . . 162
A.2. Expression syntax . . . 162
A.2.1. List of functions . . . 162
A.2.2. List of operators . . . 163
A.2.3. List of available constants . . . 163
A.3. Usage with Fortran . . . 163
A.3.1. Fortran examples . . . 163
A.3.2. Error handling . . . 164
A.4. Remarks . . . 164

B. Changes incompatible with former versions of the model code 164

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/.

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_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 controling the experiment, and the properties of dynamics, transport, physics etc.

2.1. aes_bubble_nml

The following namelist controls the parameter setting for the testcase 'aes_bubble'. In the framework of this testcase, particular initial conditions can be set by the parameters described in the table of the namelist variables hereafter:

Parameter	Type	Default	Unit	Description	Scope
aes_bubble_config% psfc	R	101325.0	Pa	Initial value of surface pressure.	
aes_bubble_config% t_am	R	180.	K	Absolute minimum of atmospheric temperature in initial state.	
aes_bubble_config% t0	R	303.5	K	Temperature at bottom of atmosphere in initial state.	
aes_bubble_config% gamma0	R	0.009	K/m	Lapse rate in lowest atmospheric part in initial temperature profile.	
aes_bubble_config% z0	R	3000.	m	Below z0, the lapse rate gamma0 is applied in the initial temperature profile, above z0, the lapse rate is gamma1.	
aes_bubble_config% gamma1	R	0.00001	K/m	Lapse rate above z0 in the initial temperature profile. However, temperature cannot fall below t_am in the initial temperature profile.	
aes_bubble_config% t_perturb	R	3.	K	Maximum temperature perturbation in center of Gaussians in initial state.	

Parameter	Type	Default	Unit	Description	Scope
aes_bubble_config% relhum_bg	R	0.7		Background relative humidity in initial state.	
aes_bubble_config% relhum_mx	R	0.95		Maximum relative humidity in initial state.	
aes_bubble_config% hw_x	R	12500.	m	Half width in x-direction in meters of the bubble in initial state.	
aes_bubble_config% hw_z	R	500.	m	Half width in z-direction in meters of the bubble in initial state.	
aes_bubble_config% x_center	R	0.	m	Placement of maximum of Gaussian relative to the origin in x-direction (if Gaussian is applied into x-direction only, lgaussxy=.FALSE.) or relative to the origin in x- and y-direction (if Gaussian is applied into x- and y- direction, lgaussxy=.TRUE.) in initial state.	
aes_bubble_config% lgaussxy	L	.FALSE.	K	.TRUE., if half width calculated for x-direction and x_center is applied also to y direction in initial state.	

2.2. aes_cop_nml

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

Parameter	Type	Default	Unit	Description	Scope
aes_cop_config(jg)% cn1lnd	R	20.	1e6/m3	cloud droplet number concentration over land	
aes_cop_config(jg)% cn2lnd	R	180.	1e6/m3	cloud droplet number concentration over land	
aes_cop_config(jg)% cn1sea	R	20.	1e6/m3	cloud droplet number concentration over sea	
aes_cop_config(jg)% cn2sea	R	80.	1e6/m3	cloud droplet number concentration over sea	
aes_cop_config(jg)% cinhomi	R	0.8		ice cloud inhomogeneity factor	
aes_cop_config(jg)% cinhoms	R	0.8		snow cloud inhomogeneity factor,	
aes_cop_config(jg)% cinhoml	R	0.4		liquid cloud inhomogeneity factor,	
aes_cop_config(jg)% cthomi	R	tmelt-35.	K	maximum temperature for homogeneous freezing	
aes_cop_config(jg)% csecfrl	R	1.5E-5	kg/kg	minimum in-cloud water mass mixing ratio in mixed phase clouds	

2.3. aes_cov_nml

The parameterization of cloud cover for the AES physics is configured by a data structure *aes_cov_config(jg=1:ndom)%<param>*, which is a 1-dimensional array extending over all domains. The structure contains the following control parameters:

Parameter	Type	Default	Unit	Description	Scope
aes_cov_config(jg)% cqx	R	1.0e-8	kg/kg	critical mass fraction of cloud water + cloud ice in air, if exceeded cloud cover in cell = 1, otherwise = 0	

2.4. aes_phy_nml

The AES physics is configured by a data structure *aes_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. Further logical switches control how the atmospheric boundary conditions for the AES physics are determined. Time control parameters are available for the atmospheric processes tabulated below.

prc	*parameterized process*
rad	LW and SW radiation
vdf	vertical diffusion
mig	graupel microphysics
two	two moment microphysics
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
aes_phy_config(jg)% dt_prc	C	""		This is the time interval in ISO 8601-2004 format at which the forcing by the process *prc* is computed.	run_nml/iforcing = 2

Parameter	Type	Default	Unit	Description	Scope
aes_phy_config(jg)% sd_prc	C	""		Defines the start date/time in ISO 8601-2004 format of the interval *[sd_prc,ed_prc[*, in which the forcing by the process *prc* is computed in intervals *dt_prc*.	run_nml/iforcing = 2 and *dt_prc* > 0.000s
aes_phy_config(jg)% ed_prc	C	""		Defines the end date/time in ISO 8601-2004 format of the interval *[sd_prc,ed_prc[*, in which the forcing by the process *prc* is computed in intervals *dt_prc*.	run_nml/iforcing = 2 and *dt_prc* > 0.000s
aes_phy_config(jg)% fc_prc	I	1		Forcing control for process *prc*. 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/iforcing = 2 and *dt_prc* > 0.000s
aes_phy_config(jg)% ljsb	L	.FALSE.		.TRUE. for using the JSBACH land surface model	run_nml/iforcing = 2
aes_phy_config(jg)% llake	L	.FALSE.		.TRUE. for using lakes in JSBACH	run_nml/iforcing = 2
aes_phy_config(jg)% lamip	L	.FALSE.		.TRUE. for AMIP boundary conditions	run_nml/iforcing = 2
aes_phy_config(jg)% l2moment	L	.FALSE.		.TRUE. for the 2-moment microphysics scheme	run_nml/iforcing = 2
aes_phy_config(jg)% lmlo	L	.FALSE.		.TRUE. for mixed layer ocean	run_nml/iforcing = 2
aes_phy_config(jg)% lice	L	.FALSE.		.TRUE. for sea-ice temperature calculation	run_nml/iforcing = 2
aes_phy_config(jg)% lsstice	L	.FALSE.		.TRUE. for inst. 6hourly sst and ice (prelim)	run_nml/iforcing = 2
aes_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/iforcing = 2
aes_phy_config(jg)% iqneg_p2d	I	0		If negative tracer mass fractions are found in the physics to dynamics interface, then: 1,3: they are reported; 2,3: they are replaced with zero	run_nml/iforcing = 2
aes_phy_config(jg)% zmaxcloudy	R	33000.	m	maximum height (m) for cloud related computations	

2.5. aes_rad_nml

The input from AES physics to the rte_rrtmgp scheme is configured by a data structure *aes_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
aes_rad_config(jg)% isolrad	I	0		Selects the spectral solar irradiation (SSI) at 1 AU distance from the sun 0: SRTM default solar spectrum, TSI = 1368.222 Wm2. 1: Time dependent solar sprectrum from file 2: Average 1844–1856 of transient CMIP5 solar, TSI = 1360.875 W/m2 3: Average 1979–1988 of transient CMIP5 solar spectrum, TSI = 1361.371 W/m2 4: Solar flux for RCE simulations with diurnal cycle, TSI = 1069.315 W/m2 5: Solar flux for RCE simulations without diurnal cycle, TSI = 433.3371 W/m2 6: Average 1850-1873 of transient CMIP6 solar, TSI = 1360.744 W/m2 7: Solar flux for RCEmip analytical simulations without diurnal cycle, TSI = 551.58 W/m2	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% fsolrad	R	1		Scaling factor for solar irradiance	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% l_orbvsop87	L	.TRUE.		.TRUE. for the realistic VSOP87 Earth orbit .FALSE. for the Kepler orbit	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% cecc	R	0.016715		eccentricity of the Kepler orbit	aes_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .FALSE.
aes_rad_config(jg)% cobld	R	23.44100	deg	obliquity of the Earth rotation axis on the Kepler orbit	aes_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .FALSE.
aes_rad_config(jg)% clonp	R	282.7000	deg	longitude of perihelion with respect to vernal equinox on the Kepler orbit	aes_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .FALSE.
aes_rad_config(jg)% lyr_perp	L	.FALSE.		.FALSE. for transient VSOP87 Earth orbit .TRUE.: VSOP87 Earth orbit of year yr_perp is perpertuated	aes_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .TRUE.
aes_rad_config(jg)% yr_perp	L	-99999		year of vsop87 orbit to be perpetuated for lyr_perp = .TRUE.	aes_phy_config(jg)% dt_rad > 0.000s and l_orbvsop87 = .TRUE.
aes_rad_config(jg)% ldiur	L	.TRUE.		.TRUE. for diurnal cycle in solar irradiation .FALSE. for zonally averaged solar irradiation	aes_phy_config(jg)% dt_rad > 0.000s

Parameter	Type	Default	Unit	Description	Scope
aes_rad_config(jg)% l_sph_symm_irr	L	.FALSE.		.TRUE. for globally averaged irradiation; .FALSE. for lat (lon) dependent irradiation	
aes_rad_config(jg)% icosmu0	I	3		PROVISIONAL - ONLY BEST METHODS WILL BE KEPT ("0" or "3") 0: no adjustment, the original cosmu0 is used 3: 0.5*SIN(dmu0)*(1+(pi/2-mu0)/dmu0), dmu0=pi*dt_rad/86400s Has small effects on the MA temp. and wind and the land surface temp.	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% irad_h2o	I	1		Selects source for concentration of water vapor, cloud water and cloud ice 0: No H2O(gas,liq,ice) in radiation 1: H2O(gas,liq,ice) mass mixing ratios from tracer fields	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% irad_co2	I	2		Selects source for concentration of CO2 0: No CO2 in radiation 1: CO2 mass mixing ratio from tracer field 2: CO2 volume mixing ratio set by 'vmr _co2' 3: CO2 volume mixing ratio from ghg scenario file	aes_phy_config(jg)% dt_rad > 0.000s and CO2 tracer is defined
aes_rad_config(jg)% irad_ch4	I	2		Selects source for concentration of CH4 0: No CH4 in radiation 2: CH4 volume mixing ratio set by 'vmr _ch4' 3: CH4 vertically constant volume mixing ratio from ghg scenario file 12: CH4 tanh-profile with surface volume mixing ratio set by 'vmr _ch4' 13: CH4 tanh-profile with surface volume mixing ratio from ghg scenario file	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% irad_n2o	I	2		Selects source for concentration of N2O 0: No N2O in radiation 2: N2O volume mixing ratio set by 'vmr _n2o' 3: N2O vertically constant volume mixing ratio from ghg scenario file 12: N2O tanh-profile with surface volume mixing ratio set by 'vmr _n2o' 13: N2O tanh-profile with surface volume mixing ratio from ghg scenario file	aes_phy_config(jg)% dt_rad > 0.000s

Parameter	Type	Default	Unit	Description	Scope
aes_rad_config(jg)% irad_cfc11	I	2		Selects source for concentration of CFC11 0: No CFC11 in radiation 2: CFC11 volume mixing ratio set by 'vmr _cfc11' 3: CFC11 volume mixing ration from ghg scenario file	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% irad_cfc12	I	2		Selects source for concentration of CFC12 0: No CFC12 in radiation 2: CFC12 volume mixing ratio set by 'vmr _cfc12' 3: CFC12 volume mixing ration from ghg scenario file	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% irad_o3	I	0		Selects source for concentration of O3 0: No O3 in radiation 1: O3 mass mixing ratio from tracer field 4: O3 constant-in-time 3-dim. volume mixing ratio from file 5: O3 transient 3-dim. volume mixing ratio from file 6: O3 clim. annual cycle 3-dim. volume mixing ratio from file aes_phy_config(jg)% dt_rad > 0.000s	
aes_rad_config(jg)% irad_o2	I	2		Selects source for concentration of O2 0: No O2 in radiation 2: O2 volume mixing ratio set by 'vmr _o2'	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% vmr_co2	R	348.0e-06	m3/m3	Volume mixing ratio of CO2	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% vmr_ch4	R	1650.0e-09	m3/m3	Volume mixing ratio of CH4	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% vmr_n2o	R	306.0e-09	m3/m3	Volume mixing ratio of N2O	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% vmr_o2	R	0.20946	m3/m3	Volume mixing ratio of O2	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% vmr_cfc11	R	214.5e-12	m3/m3	Volume mixing ratio of CFC11	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% vmr_cfc12	R	371.1e-12	m3/m3	Volume mixing ratio of CFC11	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% frad_h2o	R	1.0		Scaling factor for concentration of water vapor, cloud water and cloud ice	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% frad_co2	R	1.0		Scaling factor for concentration of CO2	aes_phy_config(jg)% dt_rad > 0.000s

Parameter	Type	Default	Unit	Description	Scope
aes_rad_config(jg)% frad_ch4	R	1.0		Scaling factor for concentration of CH4	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% frad_n2o	R	1.0		Scaling factor for concentration of N2O	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% frad_o3	R	1.0		Scaling factor for concentration of O3	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% frad_o2	R	1.0		Scaling factor for concentration of O2	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% frad_cfc11	R	1.0		Scaling factor for concentration of CFC11	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% frad_cfc12	R	1.0		Scaling factor for concentration of CFC12	aes_phy_config(jg)% dt_rad > 0.000s
aes_rad_config(jg)% irad_aero	I	2		Selects source of aerosol types 0: No aerosol in radiation 13: total tropospheric 'Kinne' aerosols, time dependent from file 19: tropospheric natural 'Kinne' aerosols from pre–industrial period (the 1850–file has to be linked for all simulated years under a name that does not contain the year) + parameterized time dependent anthropogenic 'simple plumes'	aes_phy_config(jg)% dt_rad > 0.000s

2.6. aes_vdf_nml

Parameter	Type	Default	Unit	Description	Scope
aes_vdf_config(jg)% lsfc_mom_flux	L	.TRUE.		switch on/off surface momentum flux	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% lsfc_heat_flux	L	.TRUE.		switch on/off surface heat flux	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% pr0	R	1.0		neutral limit Prandtl number, can be varied from about 0.6 to 1.0	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% f_tau0	R	0.17		neutral non-dimensional stress factor	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% c_f	R	0.185		mixing length: coriolis term tuning parameter	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% c_n	R	2.0		mixing length: stability term tuning parameter	aes_phy_config(jg)% dt_vdf > 0.000s

Parameter	Type	Default	Unit	Description	Scope
aes_vdf_config(jg)% wmc	R	0.5		ratio of typical horizontal velocity to wstar at free convection	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% fsl	R	0.4		fraction of first-level height at which surface fluxes are nominally evaluated, tuning param for sfc stress	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% fbl	R	3.0		1/fbl: fraction of BL height at which lmix hat its max	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% lmix_max	R	150.	m	maximum mixing length	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% z0m_min	R	0.000015	m	minimum roughness length	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% z0m_ice	R	0.001	m	roughness length for sea ice surfaces	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% z0m_oce	R	0.001	m	roughness length for sea water surfaces	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% turb	I	1		1: TTE scheme, 2: 3D Smagorinsky	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% smag_constant	R	0.23			aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% max_turb_scale	R	300.		max turbulence length scale	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% turb_prandtl	R	0.33333333333		turbulent prandtl number	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% km_min	R	0.001		min mass weighted turbulent viscosity	aes_phy_config(jg)% dt_vdf > 0.000s
aes_vdf_config(jg)% min_sfc_wind	R	1.		min sfc wind in free convection limit	aes_phy_config(jg)% dt_vdf > 0.000s

2.7. aes_wmo_nml

The diagnostics of the tropopause pressure, following the WMO definition is configured by a data structure *aes_ wmo_ config(jg =1:ndom)%<param>*, which is a 1-dimensional array extending over all domains:

Parameter	Type	Default	Unit	Description	Scope
aes_wmo_config(jg)% zmaxwmo	R	38000.	m	maximum height for tropopause search	
aes_wmo_config(jg)% zminwmo	R	5000.	m	minimum height for tropopause search	

2.8. 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	run_nml:ldass_lhn = .true.
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.	
lhn_logscale	L	.TRUE.		Apply all scaling factors as logarithmic values	fac_lhn_down, fac_lhn_up, fac_lhn_artif
lhn_updt_rule	I(max_dom)	0		Rule for humidity update by LHN: 0: LHN updates qv (standard). 1: LHN updates qi if qi>0 and T<0; qv update otherwise.	
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.	
rqrsgmax	R	1.0		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_refbias	L	.FALSE.		Apply a bias correction between so called reference precipitation (lhn_qrs = .TRUE.) and modelled precipitation at ground. This option is recommended when both quantities shows a systematic bias which cannot be adjusted by changing rqrsgmax.	lhn_qrs = .TRUE.

Parameter	Type	Default	Unit	Description	Scope
ref_bias0	R	1.0		In case of lhn_refbias = .TRUE. the bias correction starts with this factor. So far, there is no cycling of the factor foreseen, but could be implemented, when it seems to be beneficial.	lhn_refbias = .TRUE.
dtrefbias	R	1800.0	s	Relaxation time, which defines how fast the bias correction is done.	lhn_refbias = .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_no_ttend	L	.FALSE.		Only apply moisture increments. Temperature increments will only be used for calculation of moisture increments	lhn_hum_adj=.TRUE.
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.	K/s	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	grid points	Number of horizontal grid point, where smoothing is applied.	lhn_relax = .TRUE.
lhn_wweight	L	.FALSE.		Reduction of the LHN temperature increment in case of strong advection, messured by horizontal wind in 950, 850 and 700 hPa. The reduction is done linearly down to cero.	
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		Value of the ratio of radar to model precipitation rate, from which an artificial temperature profile is applied	lhn_artif=.TRUE.
fac_lhn_artif_tune	R	1.0		Tuning factor to optimize the effectiveness of the artificial profile.	lhn_artif=.TRUE.

Parameter	Type	Default	Unit	Description	Scope
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	K	Maximal temperature of Gaussian shaped function used a artificial temperature profile.	lhn_artif, lhn_artif_only
zlev_artif_max	R	1000.0	m	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:ldass_lhn = .true.
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.	
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 satelite cloud information	
blacklist_file(:)	C (n_dom)	'radarblacklist.nc'		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_black=.TRUE.
lhn_bright	L	.FALSE.		Apply a model intern bright band detection to avoid strong overestimation due to uncertain radar observations.	

Parameter	Type	Default	Unit	Description	Scope
height_file(:)	C (n_dom)	'radarheight.nc'		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).	lhn_bright=.TRUE.
nradar	I (n_dom)	20		Maximal number of radar height layers contained within height_file	lhn_bright=.TRUE.
lhn_spqual	L	.FALSE.		Use quality index to infer the horizontal spatial weight of the LHN increments. The quality index is read in as RAD_QUAL variable (besides the RAD_PRECIP variable) from the LHN input file.	
dace_coupling	L	.FALSE.		Invoke DACE for model equivalents of observations	Requires initicon_nml%iterate_iau=.T. if initicon_nml%init_mode == MODE_IAU (5)
dace_time_ctrl	I(3)	0		Steering parameters for DACE time control: start,end,step	
dace_debug	I	0		Debugging level for DACE interface	
dace_output_file	C	""		Filename for redirection of DACE stdout	
dace_namelist_file	C	'namelist'		Filename of the file containing the dace namelist	

Defined and used in: `src/namelists/mo_assimilation_nml.f90`

2.9. ccycle_nml

The coupling of the carbon cycle between the atmosphere and land and ocean is configured by the data structure *ccycle_config(jg=1:ndom)%<param>*, which is a 1-dimensional array extending over all domains.

Parameter	Type	Default	Unit	Description	Scope
ccycle_config(jg)% iccycle	I	0		controls the carbon cycle mode: 0: no C-cycle 1: C-cycle with interactive atmospheric CO_2 concentration 2: C-cycle with prescribed atmospheric CO_2 concentration	aes_phy_config(jg)% dt_vdf > 0.000s and aes_phy_config(jg)% ljsb = .TRUE. (and atmosphere is coupled to ocean with biogeochemistry)

Parameter	Type	Default	Unit	Description	Scope
ccycle_config(jg)% ico2conc	I	2		controls the CO_2 concentration provided to land/JSBACH and - if coupled to the ocean - to the ocean/HAMOCC 2: constant concentration as defined by ccycle_config(jg)% vmr_co2 4: transient concentration scenario from file bc_greenhouse_gases.nc	ccycle_config(jg)% iccycle = 2
ccycle_config(jg)% vmr_co2	R	284.32	ppmv	constant CO_2 volume mixing ratio of 1850 (CMIP6)	ccycle_config(jg)% ico2conc = 2

2.10. cloud_mig_nml

The parameterization of cloud microphysics 'graupel' for the AES physics is configured by a data structure *cloud_mig_config(jg=1:ndom)%<param>*, which is a 1-dimensional array extending over all domains. There are no namelist parameters available for this parameterization.

2.11. coupling_mode_nml

Parameter	Type	Default	Unit	Description	Scope
coupled_to_ocean	L	.FALSE.		.TRUE.: Required for coupled ocean-atmosphere or ocean-wave similations. Indicates the coupling of the model component at hand (e.g. atmo, wave) to the ocean model. Yac coupling routines have to be called.	
coupled_to_waves	L	.FALSE.		.TRUE.: Required for coupled wave-atmosphere or wave-ocean similations. Indicates the coupling of the model component at hand (e.g. atmo, ocean) to the wave model. Yac coupling routines have to be called.	
coupled_to_atmo	L	.FALSE.		.TRUE.: Required for coupled atmosphere-ocean or atmosphere-wave similations. Indicates the coupling of the model at hand (e.g. ocean, wave) to the atmosphere model. Yac coupling routines have to be called.	

Parameter	Type	Default	Unit	Description	Scope
coupled_to_aero	L	.FALSE.		.TRUE.: Activates the coupling of aes atmosphere to Kinne aerosole input files. Kinne aerosol input is taken from python processes rather than direct reading from pre-processed input files. rte-rrtmgp radiation supported only. In this case yac coupling routines are called.	
coupled_to_o3	L	.FALSE.		.TRUE.: Activates the coupling of aes atmosphere to o3 input files. O3 input is taken from python processes rather than direct reading from pre-processed input files. In this case yac coupling routines are called.	
coupled_to_river	L	.FALSE.		.TRUE.: Required for coupled atmosphere-river-ocean similations. Indicates the coupling of the model at hand (e.g. atmo, ocean) to the river model. Yac coupling routines have to be called.	
use_sens_heat_flux_hack	L	.FALSE.		.TRUE.: ??	
suppress_sens_heat_flux_hack_over_Ice	L	.FALSE.		.TRUE.: ??	
coupled_to_output	L	.FALSE.		enables the output coupling - All suitable variables in the varlist are defined in the coupler for coupling with external output components.	

Defined and used in: `src/namelists/mo_coupling_nml.f90`

2.12. diffusion_nml

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	
lhdiff_w	L	.TRUE.		Diffusion on the vertical wind field	
lhdiff_q	L	.FALSE.		Diffusion on QV and QC (water vapor and cloud water)	

Parameter	Type	Default	Unit	Description	Scope
hdiff_order	I	5		Order of ∇ operator for diffusion: -1: no diffusion 2: ∇^2 diffusion 3: (removed) 4: ∇^4 diffusion 5: Smagorinsky ∇^2 diffusion combined with ∇^4 background diffusion as specified via hdiff_efdt_ratio. Set hdiff_efdt_ratio<=0 for switching off background diffusion.	
lsmag_3d	L(max_dom)	.FALSE.		.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=5; itype_vn_diffu=1
lhdiff_smag_w	L(max_dom)	.FALSE.		.TRUE.: Use additional Smagorinsky diffusion for w (recommended at mesh sizes finer than 500 m if the LES turbulence scheme is not used)	hdiff_order=5; lhdiff_w=.true.
itype_vn_diffu	I	1		Reconstruction method used for Smagorinsky diffusion: 1: u/v reconstruction at vertices only 2: u/v reconstruction at cells and vertices	hdiff_order=5
itype_t_diffu	I	2		Discretization of temperature diffusion: 1: $K_h \nabla^2 T$ 2: $\nabla \cdot (K_h \nabla T)$	hdiff_order=5
hdiff_efdt_ratio	R	36.0		ratio of e-folding time to time step (or 2* time step when using a 3 time level time stepping scheme) (values above 30 are recommended when using hdiff_order=5)	
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	hdiff_order=4
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_faci	R	0.015		Scaling factor for Smagorinsky diffusion at height $hdiff\_smag\_z$ and below. $hdiff\_smag\_fac \geq 0$.	

Parameter	Type	Default	Unit	Description	Scope
hdiff_smag_fac2	R	$2 \cdot 10^{-6} \cdot (1600 + 25000 + \sqrt{(1600 \cdot (1600 + 50000)))} \approx 0.071$		Scaling factor for Smagorinsky diffusion at height $hdiff\_smag\_z2$. $hdiff\_smag\_fac2 \geq 0$. Between $hdiff\_smag\_z$ and $hdiff\_smag\_z2$ the scaling factor changes linearly from $hdiff\_smag\_fac$ to $hdiff\_smag\_fac2$.	
hdiff_smag_fac3	R	0.		Scaling factor for Smagorinsky diffusion at height $hdiff\_smag\_z3$. $hdiff\_smag\_fac3 \geq 0$. The three points $(hdiff\_smag\_z2, hdiff\_smag\_fac2)$, $(hdiff\_smag\_z3, hdiff\_smag\_fac3)$, and $(hdiff\_smag\_z4, hdiff\_smag\_fac4)$ determine the quadratic function for the scaling factor between $hdiff\_smag\_z2$ and $hdiff\_smag\_z4$.	
hdiff_smag_fac4	R	1.0		Scaling factor for Smagorinsky diffusion at height $hdiff\_smag\_z4$ and higher. $hdiff\_smag\_fac4 \geq 0$.	
hdiff_smag_z	R	32500.	m	Height up to which $hdiff\_smag\_fac$ is used, and where the linear profile up to height $hdiff\_smag\_z2$ starts.	
hdiff_smag_z2	R	$1600 + 50000 + \sqrt{(1600 \cdot (1600 + 50000))} \approx 60686$	m	Height with scaling factor $hdiff\_smag\_fac2$ where the linear profile starting at $hdiff\_smag\_z$ ends, and where the quadratic profile up to $hdiff\_smag\_z4$ starts. $hdiff\_smag\_z < hdiff\_smag\_z2 < hdiff\_smag\_z4$.	
hdiff_smag_z3	R	50000.	m	Height with scaling factor $hdiff\_smag\_fac3$. Needed to determine the quadratic function between $hdiff\_smag\_z2$ and $hdiff\_smag\_z4$. $hdiff\_smag\_z3 \neq hdiff\_smag\_z2 \wedge hdiff\_smag\_z3 \neq hdiff\_smag\_z4$.	
hdiff_smag_z4	R	90000.	m	Height from which $hdiff\_smag\_fac4$ is used. $hdiff\_smag\_z4 > hdiff\_smag\_z2$.	

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

2.13. dynamics_nml

This namelist is relevant if run_nml:ldynamics=.TRUE.

Parameter	Type	Default	Unit	Description	Scope
divavg_cntrwgt	R	0.5		Weight of central cell for divergence averaging	
lcoriolis	L	.TRUE.		Coriolis force	
ldeepatmo	L	.FALSE.		Switch for deep-atmosphere modification. [Note: only the reversible part of the dynamics (largely coincident with what is commonly referred to as “the dynamical core”) is modified for the deep atmosphere. Irreversible dynamics of any kind (largely coincident with what is commonly referred to as “the physics”) are not explicitly modified. Neither are artificial numerical measures for stabilizing, smoothing and the like modified explicitly.]	iforcing = 0, 2, 3 is_plane_torus = .FALSE.
lmoist_thdyn	L	.TRUE.		Include moisture-dependence of atmospheric heat capacities in thermodynamic equation (automatically reset to .FALSE. in dry model configurations)	

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

2.14. 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	

Parameter	Type	Default	Unit	Description	Scope
timedep_pert	I	0		Time-dependence of ensemble physics 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 2: Time-dependent perturbations varying sinusoidally within their range	Note: LHN perturbations always follow option 2 if the time dependence is not turned off.
fac_rng_spinup	I	1		Factor for number of spinup calls for random number generator	
range_gkwake	R	1.5		Variability range (multiplicative) 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	
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_ccn_Ncn0	R	1.		For 2-moment mircophysics: multiplicative change of CCN concentration for Segal & Khain cloud activation parameterization. The base value `ccn_Ncn0` may be explicitly specified in namelist `/twomom_mcrph_nml/`, otherwise the respective base value from the aerosol scenario `ccn_type` is automatically taken. Not time dependent, regardless of `timedep_pert`.	inwp_gscp = 4,5,7
range_in_fact	R	1.		For 2-moment mircophysics: multiplicative change of IN concentration for ice nucleation parameterization. Not time dependent, regardless of `timedep_pert`.	inwp_gscp = 4,5,7

Parameter	Type	Default	Unit	Description	Scope
range_avel_i	R	1.		For 2-moment mircophysics: multiplicative change of cloud ice fall speed. The base value `avel_i` may be explicitly specified in namelist `/twomom_mcrph_nml/`, otherwise the default `avel_i` is taken. Not time dependent, regardless of `timedep_pert`.	inwp_gscp = 4,5,7
range_avel_g	R	1.		For 2-moment mircophysics: multiplicative change of graupel fall speed. The base value `avel_g` may be explicitly specified in namelist `/twomom_mcrph_nml/`, otherwise the default `avel_g` is taken. Not time dependent, regardless of `timedep_pert`.	inwp_gscp = 4,5,7
range_cap_ice	R	1.		For 2-moment mircophysics: multiplicative change of capacitance of cloud ice for depositional growth. The base value `cap_ice` may be explicitly specified in namelist `/twomom_mcrph_nml/`, otherwise the default `cap_ice` is taken.	inwp_gscp = 4,5,7
range_cap_snow	R	1.		For 2-moment mircophysics: multiplicative change of capacitance of snow for depositional growth. The base value `cap_snow` may be explicitly specified in namelist `/twomom_mcrph_nml/`, otherwise the default `cap_snow` is taken.	inwp_gscp = 4,5,7
range_entrorg	R	0.2e-3	1/m	Variability range (additive) for entrainment parameter in convection scheme	inwp_convection = 1
range_entrorg_mult	R	1		Asymmetric-multiplicative variation for entrainment parameter in convection scheme, combined with a quadratic reduction of the convective adjustment time scale for positive perturbations. Should be used alternatively to the additive perturbation described above, i.e. setting a factor above 1 shall be combined with range_entrorg = 0.	inwp_convection = 1
range_rdepths	R	5.e3	Pa	Variability range for maximum allowed shallow convection depth	inwp_convection = 1

Parameter	Type	Default	Unit	Description	Scope
range_rmfdeps	R	1		Multiplicative variation of the rmfdeps parameter, i.e. the fraction of the updraft mass flux that is used as a start value for the downdraft calculation at the level of free sinking	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_texc	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_thicklayfac	R	0.0025		Variability range for thick-layer correction factor for sub-grid scale liquid cloud distribution	inwp_cldcover = 1
range_fac_ccqc	R	4		Factor for latent-heat correction in CLC-QC relationship in cloud cover scheme	inwp_cldcover = 1
range_tkhmin	R	0.2	m^2s^{-1}	Variability range for minimum vertical diffusion for heat/moisture	inwp_turb = 1
range_tkmmin	R	0.2	m^2s^{-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	1		Variability range for stability correction	inwp_turb = 1
range_c_diff	R	2.0		Range for multiplicative change of length scale factor for vertical diffusion	inwp_turb = 1
range_q_crit	R	1		Variability range for critical value for normalized supersaturation in turbulent cloud scheme	inwp_turb = 1

Parameter	Type	Default	Unit	Description	Scope
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	8.0		Variability range (additive) 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
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
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_lhn_coef	R	0.0		Scaling factor for latent heat nudging increments	latent heat nudging; i.e. ldass_lhn = .true.
range_lhn_artif_fac	R	0.0		Scaling factor for artificial heating profile in latent heat nudging	latent heat nudging; i.e. ldass_lhn = .true.
range_lhn_down	R	0.0		Lower limit for reduction of pre-existing latent heating in LHN	latent heat nudging; i.e. ldass_lhn = .true.
range_lhn_up	R	0.0		Upper limit for increase of pre-existing latent heating in LHN	latent heat nudging; i.e. ldass_lhn = .true.
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.	

Parameter	Type	Default	Unit	Description	Scope
shift_boxliq_asy	R	0.		Option to shift ensemble mean of tune_box_liq_asy w.r.t. the deterministic value.	
shift_ratsea	R	0.		Option to shift ensemble mean of rat_sea w.r.t. the deterministic value.	

Defined and used in: `src/namelists/mo_ensemble_pert_nml.f90`

2.15. 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" * "modcomp:deterministic" * "modcomp:ensemble" They correspond to: typeOfGeneratingProcess = 2/4/2/4 localDefinitionNumber = 254/253/230/230 typOfProcessedData = 1/5/1/5 typeOfEnsembleForecast = 192/192 (Note: "modcomp:..." require ecCodes version >= 2.31.0)	filetype=2
tablesVersion	I	15		Main switch for Table version	filetype=2
backgroundProcess	I	0		Background process - GRIB2 code table backgroundProcess.table	filetype=2

Parameter	Type	Default	Unit	Description	Scope
generatingCenter	I	-1		Output generating center. If this key is not set, center information is taken from the grid file ∗ 78: DWD ∗ 98: MPIMET $^{+}$ ∗ 98: ECMWF ($^{+}$ The official WMO code for the MIPMET is 252.)	filetype=2
generatingSubcenter	I	-1		Output generating Subcenter. If this key is not set, subcenter information is taken from the grid file ∗ 255: DWD ∗ 232: MPIMET ∗ 0 : ECMWF	filetype=2
generatingProcessIdentifier	I(n_dom)	1		generating Process Identifier - GRIB2 code table generatingProcessIdentifier.table	filetype=2
numberOfForecastsInEnsemble	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
productionStatusOfProcessedData	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

Parameter	Type	Default	Unit	Description	Scope
localDefinitionNumber	I	-1		local Definition Number: ∗ 254: Deterministic system ∗ 253: Ensemble system: ∗ 230: Model composition - GRIB2 code table grib2LocalSectionNumber.78.table. Note that in case of 230 ("Model composition") `preset = ''modcomp:deterministic/ensemble''` has to be used to choose between deterministic or ensemble systems! In addition, 230 requires ecCodes version >= 2.31.0.	filetype=2 generatingCenter=78/80/215
localNumberOfExperiment	I	1		local Number of Experiment	filetype=2 generatingCenter=78/80/215
localTypeOfEnsembleForecast	I	-1		Local definiton for ensemble products (only set if value changed from default)	filetype=2 generatingCenter=78/80/215
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
grib_lib_compat	C	"current"		Type of GRIB library backward compatibility adjustment: ∗ "current": No adjustment ∗ "eccodes:2.31.0": Please see Section 2.15.1 below.	filetype=2 ecCodes version >= 2.32.0

Parameter	Type	Default	Unit	Description	Scope
model_components	C(3)	" "," "," "		Model components. Currently, the following options are available: ∗ "icon-nwp" ∗ "icon-nwp", "art-nwp" ∗ "icon-nwp", "ocean-nwp" ∗ "icon-nwp", "art-nwp", "ocean-nwp" Note that it is the responsibility of the user, to choose a setting that is in line with the actual model composition!	filetype=2 generatingCenter=78/80/215 localDefinitionNumber=230 ecCodes version >= 2.31.0

2.15.1. Notes on the GRIB library backward compatiblity adjustment:

Why do we need the namelist parameter `grib_lib_compat`*?*
The I/O library CDI uses the ecCodes library for GRIB file handling. Sometimes, version updates of ecCodes come along with a change in behavior that results in some GRIB metadata having different values in GRIB output files (which cannot be avoided without additional measures). This is undesirable, at least in operational NWP. In order to allow for maintaining continuity, we try to "overwrite" such new behavior with an explicit reproduction of the prior behavior, if possible. The following Table 16 describes the options of the associated namelist parameter `grib_lib_compat` in more detail.

Table 16: Options for parameter: `grib_lib_compat`

`grib_lib_compat`	**Description**	**Applies to GRIB lib version**	**Notes**
"current"	No adjustment applied.		

Table 16: Options for parameter: `grib_lib_compat`

`grib_lib_compat`	**Description**	**Applies to GRIB lib version**	**Notes**
"eccodes:2.31.0"	CDI uses the ecCodes sample GRIB file "GRIB2.tmpl" as a starting file for ecCodes. The `SecondFixedSurface` GRIB keys are assigned the following values in GRIB2.tmpl: ∗ `typeOfSecondFixedSurface = 255 (MISSING)` ∗ `scaleFactorOfSecondFixedSurface = 255 (MISSING)` ∗ `scaledValueOfSecondFixedSurface = 2,147,483,647 (MISSING)` Now, if `typeOfSecondFixedSurface` is set to a value >= 10, 102 say ("Specific altitude above mean sea level"), but `scaleFactorOfSecondFixedSurface` and `scaledValueOfSecondFixedSurface` are not explicitly set, the result is as follows: <u>(1) *For ecCodes version < 2.32.0*</u> ∗ `typeOfSecondFixedSurface = 102` ∗ `scaleFactorOfSecondFixedSurface = 0` ∗ `scaledValueOfSecondFixedSurface = 0` <u>(2) *For ecCodes version >= 2.32.0*</u> ∗ `typeOfSecondFixedSurface = 102` ∗ `scaleFactorOfSecondFixedSurface = MISSING` ∗ `scaledValueOfSecondFixedSurface = MISSING` With `grib_lib_compat =` "eccodes:2.31.0", we try to reproduce behavior (1) even for ecCodes version >= 2.32.0.	ecCodes version >= 2.32.0	If the used ecCodes version is < 2.32.0, "eccodes:2.31.0" will be overwritten with "current".

Defined and used in: `src/namelists/mo_gribout_nml.f90`

2.16. 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.	

Parameter	Type	Default	Unit	Description	Scope
l_scm_mode	L	.FALSE.		Single Column Model (SCM) mode. Can be extended to equivalent LES and CRM setups by setting ldynamics=.TRUE. .	is_plane_torus=.TRUE.
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.	
lrescale_timestep	L	.FALSE.		if .TRUE. then the timestep will be multiplied by `grid_rescale_factor`.	
lrescale_ang_vel	L	.FALSE.		if .TRUE. then the angular velocity will be divided by `grid_rescale_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
ifeedback_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` / `experimentStartDate`). (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` / `experimentStartDate`). (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 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
lredgrid_phys	L(n_dom)	.FALSE.		If set to .true. radiation is calculated on a reduced grid (= one grid level higher) Needs to be set for each model domain separately; for the global domain, the file containing the reduced grid must be specified in the variable "radiation_grid_filename"	
nexlevs_rrg_vnest	I	8		Maximum number of extra (additional) model layers used for calculating radiation if vertical nesting is combined with a reduced radiation grid. For these layers, temperature and pressure are copied from the parent domain (thus, the difference in the number of model levels constitutes another upper limit). Higher values improve the consistency of radiative flux divergences near the top of a vertically nested domain.	lredgrid_phys = .TRUE., lvert_nest = .TRUE., latm_above_top = .TRUE.
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`.	

Parameter	Type	Default	Unit	Description	Scope
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 *and* the grid files are rather old s.t. they do not contain a `uuidOfParHGrid` global 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.	
vct_filename	C			Filename of ASCII file containing the 1D vertical coordinate tables `vct_a`, `vct_b`. See Sect. 9 for further information on the format. If empty, `vct_a`, `vct_b` are 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`

2.17. 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	n_dom>1

Parameter	Type	Default	Unit	Description	Scope
				2: gradient-based interpolation	
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: (removed) 2: RBF interpolation 3: (removed) 4: combination gradient-based / RBF 5: (removed) 6: same as 4, 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
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		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

Parameter	Type	Default	Unit	Description	Scope
l_density_nudging	L	.FALSE.		.TRUE.: Apply density nudging near lateral nest boundary if grf_intmethod_e ∈ {2,4}	n_dom>1 .AND. lfeedback = .TRUE.
fbk_relax_timescale	R	10800		Relaxation time scale for feedback	n_dom>1 .AND. lfeedback = .TRUE. .AND. ifeedback_type = 2

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

2.18. 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 hor. interpolated DWD initialized analysis data with subsequent vertical remapping	
dt_ana	R	10800	s	Time interval of assimilation cycle.	icpl_da_sfcevap>= 2
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

Parameter	Type	Default	Unit	Description	Scope
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 `ini_datetime_string` or `experimentStartDate`. `dt_shift` 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 dt_iau in first cycle. This 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 dt_shift <0
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 concentration 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 concentration are read and processed.	init_mode=5

Parameter	Type	Default	Unit	Description	Scope
qrsgana_mode	I	0		1: analysis increments for rain, snow and graupel mass concentrations are read and processed. In case of the 2-moment microphysics (inwp_gscp=4,5,6), also hail mass concentration increments are processed.	init_mode=5
qnxana_2mom_mode	I	0		Only effective in case of 2-moment microphysics (inwp_gscp=4,5,6). Affects the analysis increments of the the number concentrations of those hydrometeors in IAU which have been selected by the settings of qcana_mode, qiana_mode and qrsgana_mode: 0: analysis increments are not taken from analysis files but diagnosed based on the mass concentrations (from fg) and mass increments. 1: analysis increments are taken from the analysis files. If missing for a specific hydrometeor type, they are diagnosed similar to option 0 as a fallback.	init_mode=5, inwp_gscp=4,5,6

Parameter	Type	Default	Unit	Description	Scope
icpl_da_sfcevap	I	0		Coupling between data assimilation and model parameters controlling surface evaporation (bare soil and plants). Choosing values > 0 requires itype_vegetation_cycle=2 (in extpar_nml) : 0: off 1: use time-filtered T2M bias provided by the soil moisture analysis 2: use in addition a time-filtered RH increment at the lowest model level (requires assimilation of RH2M) 3: as option 2, but use a time-filtered temperature increment at the lowest model level instead of the T2M bias provided by the SMA (requires assimilation of T2M and RH2M) 4: as option 3, but uses the minimum evaporation resistance (default set by cr_bsmin) instead of c_soil for adaptive tuning of bare-soil evaporation 5: as option 4, but additional adjustment of hydraulic diffusivity (capillary transport) and asymmetry factor for stomata resistance	init_mode=5
smi_relax_timescale	R	20.	days	Relaxation time scale for ICON-internal soil moisture adjustment, referring to a filtered RH increment of 1%. Setting the time scale to zero turns off the soil moisture adjustment.	icpl_da_sfcevap $\geq$ 2
itype_sma	I	1		Type of soil moisture analysis used 1: use external soil moisture analysis provided by the data assimilation 2: use ICON-internal SMA based on adaptive parameter tuning	init_mode=5; icpl_da_sfcevap$\geq$3

Parameter	Type	Default	Unit	Description	Scope
icpl_da_snowalb	I	0		Coupling between temperature bias inferred from data assimilation and snow albedo 0: off 1: on; requires assimilation of T2M and cycling of a time-filtered temperature increment at the lowest model level 2: as option 1, but additional adaptation of sea-ice albedo 3: as option 2, but additional adaptation of snow-cover fraction diagnosis init_mode=5; icpl_da_sfcevap≥3	
icpl_da_landalb	I	0		Coupling between temperature/humidity bias inferred from data assimilation and albedo of snow-free land 0: off 1: on; requires assimilation of T2M and RH2M and cycling of the full set of filtered assimilation increments coming along with icpl_da_sfcevap≥5 init_mode=5; icpl_da_sfcevap≥5	
icpl_da_seaice	I	0		Coupling between temperature bias inferred from data assimilation and seaice scheme 0: off 1: add filtered T increment to initial seaice temperature 2: as above, and additional adaptive tuning of bottom heat flux if lbottom_hflux = true	init_mode=5; icpl_da_sfcevap≥3
icpl_da_skinc	I	0		Coupling between bias of diurnal temperature amplitude inferred from data assimilation and skin conductivity 0: off 1: on; requires assimilation of T2M and cycling of a time-filtered weighted (with cosine of local time) temperature increment at the lowest model level 2: as option 1, but additional adaptation of soil heat conductivity and heat capacity	init_mode=5

Parameter	Type	Default	Unit	Description	Scope
icpl_da_sfcfric	I	0		Coupling between data assimilation and model parameters controlling surface friction (roughness length and SSO blocking tendency at lowest level). 0: off 1: on; requires assimilation of 10m-winds and cycling a time-filtered assimilation increment of absolute wind speed at the lowest model level; moreover, it is strongly recommended to use extpar data with full SSO information (generated in Feb. 2022 or later). Coupling is masked in large parts of Russia where the assimilation of 10m winds is blacklisted in the operational settings of 2022 2: on without masking over Russia, to be combined with 10m wind assimilation without blacklisting	init_mode=5
scalfac_da_sfcfric	R	2.5		Scaling factor for adaptive surface friction (see eqns. 3 and 4 in `https://doi.org/10.1002/qj.4535`)	icpl_da_sfcfric > 0
icpl_da_tkhmin	I	0		Coupling between data assimilation and near-surface reduction profile for minimum vertical diffusion of heat 0: off 1: on	init_mode=5, icpl_da_sfcevap > 2 and icpl_da_skinc > 0
adjust_tso_tsnow	L	.FALSE.		If .TRUE., apply T increments for lowest model level also to snow and upper soil layers (full to upper 3 cm, half to 3-9 cm layer). Requires assimilation of T2M to be meaningful	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 *uuidOfHGrid* and *validity time*.	init_mode=1,3,4,5,6,7
l_coarse2fine_mode	L(n_dom)	.FALSE.		If true, apply corrections for coarse-to-fine mesh interpolation to wind and temperature	

Parameter	Type	Default	Unit	Description	Scope
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
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=1,5,6,7
ltile_coldstart	L	.FALSE.		If true, tiled surface fields are initialized with tile-averaged fields from a previous run with tiles. A neighbor search is applied to subgrid-scale ocean points for SST and sea-ice fraction.	init_mode=1,5,6,7
lcouple_ocean_coldstart	L	.TRUE.		If true, initialize newly defined land points from ICON-O with default T and Q profiles.	is_coupled_mode=T
lvert_remap_fg	L	.FALSE.		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 "`<path>ifs2icon_R<nroot>B<jlev>_DOM <idom>.nc`". May contain the keywords `<path>` which will be substituted by `model_base_dir`, as well as `nroot`, `nroot0`, `jlev`, and `idom` defining the current patch.	init_mode=2

Parameter	Type	Default	Unit	Description	Scope
dwdfg_filename	C			Filename of DWD first-guess input file, default "`<path>dwdFG_R<nroot>B<jlev>_DOM<idom>.nc`". May contain the keywords `<path>` which will be substituted by `model_base_dir`, as well as `nroot`, `nroot0`, `jlev`, and `idom` defining the current patch.	init_mode=1,3,5,6,7
dwdana_filename	C			Filename of DWD analysis input file, default "`<path>dwdana_R<nroot>B<jlev>_DOM<idom>.nc`". May contain the keywords `<path>` which will be substituted by `model_base_dir`, as well as `nroot`, `nroot0`, `jlev`, and `idom` defining the current patch.	init_mode=1,3,5,6
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".	
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.	init_mode=1,5,6,7
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.	init_mode=1,5,6

Parameter	Type	Default	Unit	Description	Scope
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; l_limited_area = .FALSE.
fire2d_filename	C	'gfas2d_emi_ <species>_ <gridfile>_ <yyyymmdd>.nc'		Wildfire emission data sets for the <species> bc, oc and so2. Possible keywords: <species>, <gridfile>, <nroot>, <nroot0>, <jlev>, <idom>, <yyyymmdd>	Requires: iprog_aero = 3

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

2.19. interpol_nml

Parameter	Type	Default	Unit	Description	Scope
l_intp_c2l	L	.TRUE.		DEPRECATED	
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.	
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 of high order least-squares reconstruction for tracer transport 1: linear 2: quadratic 3: cubic	ihadv_tracer > 2
llsq_lin_consv	L	.FALSE.		conservative (T) or non-conservative (F) least-squares reconstruction for 2nd order (linear) transport	

Parameter	Type	Default	Unit	Description	Scope
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 $0.06 - 0.075$. The range of validity is $[0 - 0.2]$. **Please note** that the user value is internally multiplied by 5.	
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 `output_nml::rbf_scale`, p. 104).	
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: 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	
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/namelists/mo_interpol_nml.f90`

2.20. 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	`run_nml:output = "totint"`
dt_checkpoint	R	0	s	Time interval for writing restart files. Note that if the value of dt_checkpoint resulting from model default or user's specification is longer than time_nml:dt_restart, it will be reset (by the model) to dt_restart so that at least one restart file is generated during the restart cycle.	`output /= "none"` (`run_nml`)
inextra_2d	I	0		Number of extra 2D Fields for diagnostic/debugging output.	
inextra_3d	I	0		Number of extra 3D Fields for diagnostic/debugging output.	
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	iforcing=3

Parameter	Type	Default	Unit	Description	Scope
itype_hzerocl	I	1		Specifies setting of hzerocl if no freezing level is found. 1: Height of orography, 2: -999.0_wp (undef), 3: extrapolated value below ground (assuming -6.5 K/km).	
itype_pres_msl	I	1		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)	
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(n_dom)	3600.	s	Interval over which wind gusts are maximized	iforcing=3
ff10m_interval	R(n_dom)	600.	s	Interval over which 10-m winds are averaged (and used as basis for the gust diagnosis)	itune_gust_diag=4
celltracks_interval	R(n_dom)	3600.	s	Interval over which celltrack variables are maximized (lpi_max, uh_max, vorw_ctmax, w_ctmax, tcond_max, tcond10_max, dbz_ctmax, tot_pr_max)	iforcing=3
dt_celltracks	R(n_dom)	120.	s	Interval at which celltrack variables except lpi (uh, vorw, w_ct, tcond, tcond10) are calculated to determine uh_max, vorw_ctmax, w_ctmax, tcond_max, tcond10_max and dbz_ctmax	iforcing=3
dt_lpi	R(n_dom)	180.	s	Interval at which lpi is calculated for determining lpi_max	iforcing=3
dt_hailcast	R(n_dom)	180.	s	Interval at which hailcast is called for determining dhail_mx, dhail_sd, dhail_av	iforcing=3

Parameter	Type	Default	Unit	Description	Scope
wdur_min_hailcast	R(n_dom)	900.	s	Minimal updraft persistence per column for hailcast to be activated	iforcing=3
dt_radar_dbz	R(n_dom)	120.	s	Interval at which radar reflectivity is calculated for determining dbz_ctmax	iforcing=3
force_calc_optvar	I(n_dom)	0		Allows to force the computation of optional diagnostics in domains where no output is written, e.g. to have valid fields for nest boundary interpolation. By default, the computations are triggered by the output namelists of a given model domain. Setting, for instance, the second entry to 3 means that the output namelists of domain 3 are used to trigger the optional diagnostics in domain 2 as well. **Caution: If the output fields written in a nested domain are a subset of the fields written in the parent domain, using this option will cause a failure!**	iforcing=3
precip_interval	C(n_dom)	"P01Y"		Interval over which precipitation variables are accumulated (rain_gsp, snow_gsp, graupel_gsp, ice_gsp, hail_gsp, prec_gsp, rain_con, snow_con, prec_con, tot_prec, prec_con_rate_avg, prec_gsp_rate_avg, tot_prec_rate_avg)	iforcing=3
totprec_d_interval	C(n_dom)	"PT01H"		Interval over which the special precipitation variable tot_prec_d is accumulated, which can be output alongside or alternatively to tot_prec and enables a different accumulation time for this field than precip_interval.	iforcing=3
maxt_interval	C(n_dom)	"PT06H"		Interval over which max/min 2-m temperatures are calculated	iforcing=3
runoff_interval	C(n_dom)	"P01Y"		Interval over which surface and soil water runoff are accumulated	iforcing=3
sunshine_interval	C(n_dom)	"P01Y"		Interval over which sunshine duration is accumulated	iforcing=3
itype_dursun	I	0		Type of sunshine. 0 for WMO standard and for sunshine duration counted if $>120 W/m^2$. In the case of type 1 (this is the MeteoSwiss definition) the sunshine duration is counted only if $>200 W/m^2$	iforcing=3

Parameter	Type	Default	Unit	Description	Scope
wshear_uv_heights	R(max_wshear) max_wshear=10	1000.0, 3000.0, 6000.0		List of height levels (m AGL) for which the vertical windshear output variables "wshear_u" and "wshear_v" are to be output.	iforcing=3
srh_heights	R(max_srh) max_srh=10	1000.0, 3000.0		List of height levels (m AGL) for which the storm relative helicity "srh" is to be output. "srh" is a vertical integral from the ground to a certain height. The listed height levels denote different upper bounds for this integration.	iforcing=3
echotop_meta	TYPE(n_dom)			Derived type to define properties of radar reflectivity echotops for each domain. Two types of echotops are available: minimum pressure ('echotop') and maximum height ('echotopinm') during a given time interval where a given reflectivity threshold is exeeded. Takes effect if 'echotop' and/or 'echotopinm' is/are present in the `ml_varlist` of any domain-specific namelist `output_nml`. The derived type contains the echotop properties which are listed to the left, along with their defaults and units: **time_interval:** time interval [s] over which echotops are calculated **dbzthresh:** list of reflectivity thresholds [dBZ] for which echotops shall be computed You have to specify properties for each domain separately, e.g. echotop_meta(1)%time_interval=3600.0 echotop_meta(1)%dbzthresh=19.0,25.0,35.0,46.0 echotop_meta(2)%time_interval=1800.0 echotop_meta(2)%dbzthresh=27.0,36.0	iforcing=3
This type contains:					
echotop_meta(1:n_dom)%time_interval	R(1)	3600.0	s		
echotop_meta(1:n_dom)%dbzthresh	R(max_echotop) max_echotop=10	(/18.0,25.0,35.0/)	dBZ		

Parameter	Type	Default	Unit	Description	Scope
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.	`output_nml` namelists
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).	
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 (*inverse to the definition of* 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

Parameter	Type	Default	Unit	Description	Scope
lnetcdf_flt64_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"`
checkpoint_on_demand	L	F		`.TRUE.` allows checkpointing (followed by stopping) during runtime triggered by a file named 'stop_icon' in the working directory. In addition, a file named 'ready_for_checkpoint' is generated in the working directory once the model is ready for checkpointing, i.e. after the end of the setup phase, or, if applicable, the end of the IAU phase.	Combination with `restart_write_mode` = `"joint procs multifile"` is strongly recommended
lmask_boundary	L(n_dom)	F		Set to `.TRUE.`, if interpolation zone should be masked in triangular output.	

2.20.1. 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”`.

2.20.2. Some notes on the output of optional diagnostics:

■ *How can I switch on the output of one of the available diagnostics?*

Let us assume that you would like to output *potential vorticity* (see table of available diagnostics below) on model levels. Simply add the following element to the desired output namelist (see 2.34) in your run script:

```
&output_nml
...
ml_varlist = ..., 'pv'
...
/
```

Please note that the output of some diagnostics is restricted to the NWP mode (`iforcing = inwp = 3`, see column “Scope” in the table 22 below).

■ *Which optional diagnostics are currently available for output?*

Here is a table of the available diagnostics and some additional information on them.

Table 22: Optional diagnostics (last update Aug. 2020)

Short name*	**Long name**	**Unit**	**Scope**	**Shape**	**Specifications in io_nml**	**Place of computation in source code****
rh	relative humidity	%	iforcing = inwp = 3	3d	itype_rh	[1]
pv	potential vorticity	K m2 kg-1 s-1	iforcing = inwp	3d	-	[2]
sdi2	supercell detection index (SDI2)	s-1	iforcing = inwp	2d	-	[2]
lpi	lightning potential index (LPI)	J kg-1	iforcing = inwp	2d	-	[2]

Table 22: Optional diagnostics (last update Aug. 2020)

Short name*	Long name	Unit	Scope	Shape	Specifications in io_nml	Place of computation in source code**
lpi_max	lightning potential index, maximum during prescribed time interval	J kg-1	iforcing = inwp	2d	celltracks_interval dt_lpi	[2]
ceiling	ceiling height	m	iforcing = inwp	2d	-	[2]
vis	near-surface horizontal visibility	m	iforcing = inwp	2d	-	[2]
hbas_sc	cloud base above msl, shallow convection	m	iforcing = inwp	2d	-	[2]
htop_sc	cloud top above msl, shallow convection	m	iforcing = inwp	2d	-	[2]
twater	total column-integrated water	kg m-2	iforcing = inwp	2d	-	[2]
q_sedim	specific content of precipitation particles	kg kg-1	iforcing = inwp	2d	-	[2]
tcond_max	total column-integrated condensate, maximum during prescribed time interval	kg m-2	iforcing = inwp	2d	celltracks_interval dt_celltracks	[2]
tcond10_max	total column-integrated condensate above z(T=-10 degC), maximum during prescribed time interval	kg m-2	iforcing = inwp	2d	celltracks_interval dt_celltracks	[2]
uh_max	updraft helicity, maximum during prescribed time interval	m2 s-2	iforcing = inwp	2d	celltracks_interval dt_celltracks	[2]
vorw_ctmax	maximum rotation amplitude during prescribed time interval	s-1	iforcing = inwp	2d	celltracks_interval dt_celltracks	[2]
w_ctmax	maximum updraft track during prescribed time interval	m s-1	iforcing = inwp	2d	celltracks_interval dt_celltracks	[2]
dbz	radar reflectivity	dBZ	iforcing = inwp	3d	-	[2]
dbz_cmax	column maximum reflectivity	dBZ	iforcing = inwp	2d	-	[2]
dbz_850	reflectivity in approx. 850 hPa	dBZ	iforcing = inwp	2d	-	[2]
dbz_ctmax	column and time maximum reflectivity during prescribed time interval	dBZ	iforcing = inwp	2d	celltracks_interval dt_radar_dbz	[2]
echotop	minimum pressure of exceeding radar reflectivity threshold during prescribed time interval	Pa	iforcing = inwp	3d	celltracks_interval echotop_meta	[2]
echotopinm	maximum height of exceeding radar reflectivity threshold during prescribed time interval	m	iforcing = inwp	3d	celltracks_interval echotop_meta	[2]
pres_msl	mean sea level pressure	Pa	-	2d	itype_pres_msl	[3]
omega	vertical (pressure) velocity	Pa s-1	-	3d	-	[2]

Table 22: Optional diagnostics (last update Aug. 2020)

Short name*	**Long name**	**Unit**	**Scope**	**Shape**	**Specifications in io_nml**	**Place of computation in source code****
tot_prec_d	total accumulated precipitation during a different time interval compared to tot_prec	kg m-2	iforcing = inwp	2d	totprec_d_interval	[1], [4], [5]
tot_pr_max	maximum total precipitation rate during prescribed time interval	kg m-2 s-1	iforcing = inwp	2d	celltracks_interval	[4]
lapse_rate	temperature gradient between 500 and 850 hPa	K m-1	iforcing = inwp	2d	-	[2]
mconv	low level horizontal moisture convergence averaged over 0-1000 m AGL layer based on specific humidity, $\frac{1}{1\,\text{km}}\int_0^{1\,\text{km AGL}} \nabla_h \cdot (q_v \vec{v}_h)\, dz$	s-1	iforcing = inwp	2d	-	[2]
wshear_u	difference of U component between certain heights (“wshear_uv_heights”) AGL and the lowest model level	m s-1	iforcing = inwp	3d	wshear_uv_heights	[2]
wshear_v	difference of V component between certain heights (“wshear_uv_heights”) AGL and the lowest model level	m s-1	iforcing = inwp	3d	wshear_uv_heights	[2]
srh	storm relative helicity considering storm motion estimate of Bunkers et al. (2000) for right-movers. srh is a vertical intergal up to a certain height AGL and may be output for different upper bounds (“srh_heights”).	m2 s-2	iforcing = inwp	3d	srh_heights	[2]
cape_mu	approximate value of the most unstable CAPE considering a test parcel from the height level with largest equivalent potential temperature between the ground and 3000 m AGL	J kg-1	iforcing = inwp	2d	-	[2]
cin_mu	approximate value of the most unstable CIN consistent to cape_mu	J kg-1	iforcing = inwp	2d	-	[2]

* To be used in output_nml.
** The keys, [1], [2], etc., are itemized under the following point.

■ *Where can I find more about the computation of the diagnostics in the source code?*

As for the ICON model component of the non-hydrostatic atmosphere:

Each optional diagnostic has its own switch in the source code of ICON which is set to `.TRUE.` if the diagnostic is found in one of the `output_nml` in your run script. This configuration can be found in the module:
`/src/configure_model/mo_io_config`.

Further information on the metadata of the diagnostics can be found in their allocation area. For the diagnostics that are meant for the NWP mode of ICON (`iforcing = inwp = 3`, see column "Scope" in table 22 above), the allocation takes place in:
`/src/atm_phy_nwp/mo_nwp_phy_state`.
Optional diagnostics with unrestricted scope are allocated in:
`/src/atm_dyn_iconam/mo_nonhydro_state`.

The job control of the computation and output of most of the optional diagnostics is organized by the post-processing scheduler:
`/src/atm_dyn_iconam/mo_pp_scheduler`,
`/src/atm_dyn_iconam/mo_pp_tasks`,
and integrated into the main time loop in:
`/src/atm_dyn_iconam/mo_nh_stepping`.
The job control of a small portion of the diagnostics is organized in:
`/src/atm_phy_nwp/mo_nwp_diagnosis`.

Finally, the computation of the individual diagnostics can be found in the following modules (the assignment of the keys, [1], [2], etc., to the respective diagnostic is found in the column "Place of computation in source code" of table 22 above):

[1] `/src/atm_phy_nwp/mo_util_phys`
[2] `/src/atm_phy_nwp/mo_opt_nwp_diagnostics`
[3] `/src/atm_phy_nwp/mo_nh_diagnose_pmsl`
[4] `/src/atm_phy_nwp/mo_nwp_gscp_interface`
[5] `/src/atm_phy_nwp/mo_nwp_diagnosis`

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

2.21. les_nml (parameters for LES turbulence scheme; valid for inwp_turb=5)

Parameter	Type	Default	Unit	Description	Scope
sst	R	300	K	sea surface temperature for idealized LES simulations	isrfc_type=5,4
shflx	R	0.1	Km/s	Kinematic sensible heat flux at surface	isrfc_type = 2
lhflx	R	0	m/s	Kinematic latent heat flux at surface	isrfc_type = 2

Parameter	Type	Default	Unit	Description	Scope
isrfc_type	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 qv_s case with prescribed roughness length for semi-idealized setups	
ufric	R	-999	m/s	friction velocity for idealized LES simulations; if <0 then it is automatically diagnosed	
psfc	R	-999	Pa	surface pressure for idealized LES simulations; if <0 then it uses the surface pressure from dynamics	
min_sfc_wind	R	1.0	m/s	Minimum surface wind for surface layer useful in the limit of free convection	
is_dry_cbl	L	.FALSE.		switch for dry convective boundary layer simulations	
smag_constant	R	0.23		Smagorinsky constant	
km_min	R	0.0		Minimum turbulent viscosity	
smag_coeff_type	I	1		choose type of coefficient setting: 1 = Smagorinsky model (default) 2 = set coeff. externally by Km_ext, Kh_ext (for testing purposes, e.g. Straka et al. (1993))	
Km_ext	R	75.0	m^2/s	externally set constant kinematic viscosity	smag_coeff_type=2
Kh_ext	R	75.0	m^2/s	externally set constant diffusion coeff.	smag_coeff_type=2
max_turb_scale	R	300.0		Asymtotic maximum turblence length scale (useful for coarse grid LES and when grid is vertically stretched)	
turb_prandtl	R	0.333333		turbulent Prandtl number	
bflux	R	0.0007	m^2/s^3	buoyancy flux for idealized LES simulations (Stevens 2007)	isrfc_type=3
tran_coeff	R	0.02	m/s	transfer coefficient near surface for idealized LES simulation (Stevens 2007)	isrfc_type=3

Parameter	Type	Default	Unit	Description	Scope
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/namelists/mo_les_nml.f90`

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

Parameter	Type	Default	Unit	Description	Scope
itype_latbc	I	0		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)	
dtime_latbc	R	-1.0	s	Time difference between two consecutive boundary data. (Upper bound for asynchronous read-in: 1 day = 86400 s.)	itype_latbc ≥ 1
init_latbc_from_fg	L	.FALSE.		If .TRUE., take lateral boundary conditions for initial time from first guess (or analysis) field	itype_latbc ≥ 1
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 ≥ 1

Parameter	Type	Default	Unit	Description	Scope
fac_latbc_presbiascor	R	0.		Scaling factor for pressure bias correction at lateral boundaries. Requires running in data assimilation cycle. Recommended value for activating the option is 1.	itype_latbc $\geq$ 1, init_mode=5
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 = 1
latbc_path	C	'./'		Absolute path to boundary data.	itype_latbc = 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 = 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
latbc_contains_qcqi	L	.TRUE.		Set to .FALSE. if there is no qc, qi in latbc data.	
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/namelists/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.

2.23. lnd_nml

Parameter	Type	Default	Unit	Description	Scope
nlev_snow	I	2		number of snow layers	lmulti_snow=.true.
ntiles	I	1		number of tiles	
zml_soil	R(:)	0.005, 0.02, 0.06, 0.18, 0.54, 1.62, 4.86, 14.58	m	soil full layer depths	init_mode = 2, 3
czbot_w_so	R	2.5	m	thickness of the hydrological active soil layer	
lsnowtile	L	.FALSE.		.TRUE.: consider snow-covered and snow-free tiles separately	ntiles>1
frlnd_thrhld	R	0.05		fraction threshold for creating a land grid point	ntiles>1
frlake_thrhld	R	0.05		fraction threshold for creating a lake grid point	ntiles>1
frsea_thrhld	R	0.05		fraction threshold for creating a sea grid point	ntiles>1
frlndtile_thrhld	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	
lmelt_var	L	.TRUE.		.TRUE. freezing temperature dependent on water content	
lana_rho_snow	L	.TRUE.		.TRUE. take rho_snow-values from analysis file	init_mode=1
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 = .FALSE.

Parameter	Type	Default	Unit	Description	Scope
max_toplaydepth	R	0.25	m	maximum depth of uppermost snow layer	lmulti_snow=.TRUE. or l2lay_rho_snow=.TRUE.
idiag_snowfrac	I	1		Type of snow-fraction diagnosis: 1 = based on SWE only 2 = more advanced method used operationally 20 = same as "2", but with artificial reduction of snow fraction in case of melting snow (should be used only in combination with 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 avaliable from the DWD assimilation cycle)	lsnowtile=.TRUE.
itype_lndtbl	I	3		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_root	I	2		type of root density distribution 1 = constant 2 = exponential	

Parameter	Type	Default	Unit	Description	Scope
itype_evsl	I	2		type of bare soil evaporation parameterization 2 = BATS scheme, Dickinson (1984) 3 = ISBA scheme, Noilhan and Planton (1989) 4 = Resistance-based formulation by Schulz and Vogel (2020) 5 = same as 4, but uses the minimum evaporation resistance (default set by cr_bsmin) instead of c_soil for tuning; the namelist parameter c_soil is ignored in this case, and a value of 2 is used internally	
itype_trvg	I	2		type of vegetation 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_canopy	I	1		Type of canopy parameterization with respect to surface energy balance 1 = Surface energy balance equation solved at the ground surface, canopy energetically not represented 2 = Skin temperature formulation by Schulz and Vogel (2020), based on Viterbo and Beljaars (1995)	
cskinc	R	-1.0	$Wm^{-2}K^{-1}$	Skin conductivity For cskinc $<$ 0, an external parameter field SKC is read and used For cskinc $>$ 0, this globally constant value is used in the whole model domain Reasonable range: 10.0 – 1000.0	itype_canopy = 2
tau_skin	R	3600.	s	Relaxation time scale for the computation of the skin temperature	itype_canopy = 2
lterra_urb	L	.FALSE.		If .TRUE., activate urban model TERRA_URB by Wouters et al. (2016, 2017) (see Schulz et al. 2023)	

Parameter	Type	Default	Unit	Description	Scope
lurbalb	L	.TRUE.		If .TRUE., use urban albedo and emissivity (Wouters et al. 2016)	lterra_urb = .TRUE.
itype_ahf	I	2		If ≥1, use urban anthropogenic heat flux (Wouters et al. 2016) 1: constant value given by the first entry in tuning_nml:tune_urbahf 2: variable value depending on climatological T2M as specified in tuning_nml:tune_urbahf 3: as option 2, but using a time-filtered predicted T2M rather than climatology	lterra_urb = .TRUE.
itype_kbmo	I	2		Type of bluff-body thermal roughness length parameterisation 1 = Standard SAI-based turbtran (Raschendorfer 2001) 2 = Brutsaert-Kanda parameterisation for bluff-body elements (kB-1) (Kanda et al. 2007) 3 = Zilitinkevich (1970)	lterra_urb = .TRUE.
itype_eisa	I	3		Type of evaporation from impervious surface area 1 = Evaporation like bare soil (see Schulz and Vogel 2020) 2 = No evaporation 3 = PDF-based puddle evaporation (Wouters et al. 2015)	lterra_urb = .TRUE.
itype_heatcond	I	2		type of soil thermal conductivity 1 = constant soil thermal conductivity 2 = moisture dependent soil thermal conductivity (see Schulz et al. 2016) 3 = variant of option 2 with reduced near-surface thermal 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	
cwimax_ml	R	$1.e-6$	m	scaling parameter for maximum interception storage (almost switched off); use $5.e-4$ to activate interception storage	itype_interception = 1

Parameter	Type	Default	Unit	Description	Scope
c_soil	R	1.		surface area density of the (evaporative) soil surface allowed range: 0 – 2	itype_evsl = 2,3,4
c_soil_urb	R	1.		surface area density of the (evaporative) soil surface, urban areas allowed range: 0 – 2	itype_evsl = 2,3,4
cr_bsmin	R	110.	s/m	minimum bare soil evaporation resistance (see Schulz and Vogel 2020) Note: c_soil and c_soil_urb are ignored in this case	itype_evsl = 5 or icpl_da_sfcevap = 4
rsmin_fac	R	1.		scaling factor for rsmin.	
itype_hydbound	I	1		type of hydraulic lower boundary condition 1 = none 3 = ground water as lower boundary of soil column	
lstomata	L	.TRUE.		If .TRUE., use map of minimum stomatal resistance If .FALSE., use constant value of 150 s/m.	
l2tls	L	.TRUE.		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	lseaice=.TRUE.
lbottom_hflux	L	.FALSE.		If .TRUE., use parameterization for bottom heat flux in seaice scheme	lseaice=.TRUE.
llake	L	.TRUE.		.TRUE. for use of lake model	

Parameter	Type	Default	Unit	Description	Scope
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. (This mode also applies to coupled atmo/ocean simulations.) 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) 6: SST and sea ice fraction are updated with user-defined interval	iforcing=3
itype_oskin_warm	I	0		SST skin formulation: warm layer component 0: off 1: activated	
itype_oskin_cold	I	0		SST skin formulation: cold skin component 0: off 1: activated	
hice_min	R	0.05	m	Minimum sea-ice thickness	lseaice=.TRUE.
hice_max	R	3.0	m	Maximum sea-ice thickness (for coupled runs assure consistency with seaice_limit)	lseaice=.TRUE.
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 In case sstice_mode=6, SST data for all time steps in the current simulation should be prepared in one single file, variable should be named SST in this file.	sstice_mode=3,4,5,6

Parameter	Type	Default	Unit	Description	Scope
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 In case sstice_mode=6, sea ice data for all time steps in the current simulation should be prepared in one single file, variable should be named SIC in this file.	sstice_mode=3,4,5,6
lcuda_graph_lnd	L	.FALSE.		Activate cuda graphs for the land scheme. Automatically set to .FALSE. if not compiled with the ICON_USE_CUDA_GRAPH cpp key.	ICON_USE_CUDA_GRAPH activated

Defined and used in: `src/namelists/mo_lnd_nwp_nml.f90`

2.24. ls_forcing_nml (parameters for large-scale forcing; valid for torus geometry; is_plane_torus=.TRUE.)

Parameter	Type	Default	Unit	Description	Scope
is_ls_forcing	L	.TRUE.		switch for enabling LS forcing	
is_subsidence_moment	L	.FALSE.		switch for enabling LS vertical advection due to subsidence for momentum equations	
is_subsidence_heat	L	.FALSE.		switch for enabling LS vertical advection due to subsidence for thermal equations	
is_advection	L	.FALSE.		switch for enabling LS horizontal advection	
is_advection_uv	L	.TRUE.		switch for enabling LS horizontal advection for u and v	is_advection=.TRUE.
is_advection_tq	L	.TRUE.		switch for enabling LS horizontal advection for temperature and moisture	is_advection=.TRUE.
is_nudging	L	.FALSE.		switch for enabling LS Newtonian relaxation (nudging)	
is_nudging_uv	L	.TRUE.		switch for enabling LS Newtonian relaxation (nudging) for horizontal winds only	is_nudging=.TRUE.
is_nudging_tq	L	.TRUE.		switch for enabling LS Newtonian relaxation (nudging) for temperature and specific humidity only	is_nudging=.TRUE.
nudge_start_height	R	1000.	m	height where nudging starts	is_nudging=.TRUE.
nudge_full_height	R	2000.	m	height where nudging reaches full strength	is_nudging=.TRUE.
dt_relax	R	3600.	s	relaxation time scale for the nudging	is_nudging=.TRUE.

Parameter	Type	Default	Unit	Description	Scope
is_geowind	L	.FALSE.		switch for enabling geostrophic wind	
is_rad_forcing	L	.FALSE.		switch for enabling radiative forcing	inwp_rad=.FALSE.
is_sim_rad	L	.FALSE.		switch for enabling a simplified radiation scheme	inwp_rad=.FALSE.
is_theta	L	.FALSE.		switch to indicate that the prescribed radiative forcing is for potential temperature	is_rad_forcing=.TRUE.

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

2.25. 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_namelists	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.	

2.26. master_model_nml (repeated for each model)

Parameter	Type	Default	Unit	Description	Scope
model_name	C			Character string for naming this component.	
model_namelist_ filename	C			File name containing the model namelists.	

Parameter	Type	Default	Unit	Description	Scope
model_type	I	-1		Identifies which component to run. 1=atmosphere 2=ocean 3=radiation 4=hamocc 5=jsbach 98=wave 99=dummy_model	
model_do_restart	C	'yes'/'no' based on lrestart (master_nml)		Allows to overwrite the main restart switch `lrestart (master_nml)` for a single model component. Available options: 'no' : no restart 'yes': classic restart 'for_init': initialize model from restart file (skips reading of restart attributes and ignores missing variables in restart file)	
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.	
model_rank_group_size	I	1		???	

2.27. master_time_control_nml

Parameter	Type	Default	Unit	Description	Scope
calendar	C	"proleptic gregorian"		Selects the calendar type to use: "proleptic gregorian" = proleptic Gregorian calendar "365 day year" = 365 day year without leap years "360 day year" = 360 day year with 30 day months	
experimentReferenceDate	C	" "	ISO8601 formatted string	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.	

Parameter	Type	Default	Unit	Description	Scope
experimentStartDate	C	" "	ISO8601 formatted string	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	This is the date an experiment is finished.	
forecastLeadTime	C	" "	ISO8601 formatted string	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	Time interval for writing checkpoints.	
restartTimeIntVal	C	" "	ISO8601 formatted string	Time interval for writing a restart file and interrupt the current running job.	

2.28. meteogram_output_nml

This namelist is relevant if run_nml: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)	"METEO GRAM_"		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)	

Parameter	Type	Default	Unit	Description	Scope
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/namelists/mo_mtgrm_nml.f90`

2.29. nonhydrostatic_nml

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 slops, otherwise no significant impact) 6: As 5, but velocity tendencies are also computed in both substeps (no apparent benefit, but more expensive)	
rayleigh_type	I	2		Type of Rayleigh damping 1: CLASSICAL (requires velocity reference state!) 2: Klemp (2008) type	
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	

Parameter	Type	Default	Unit	Description	Scope
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	ihadv_tracer=22, 32, 42 or 52
htop_aero_proc	R	22500.0	m	Height above which physical processes and advection of ART aerosol tracer variables are turned off; the default value is set to the same value as htop_moist_proc. This value is taken for all ART aerosol tracers, but not chemical tracers for which physical processes and advection are computed in all model levels per default; it may be overwritten for specific ART tracers (also chemical tracers) by the tag 'htop_proc' in the XML file when defining the individual ART tracers.	ART aerosol 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	
ivctype	I	2		Type of vertical coordinate: 1: Gal-Chen hybrid 2: SLEVE (uses sleve_nml)	
ndyn_substeps	I	5		number of dynamics substeps per fast-physics / transport step	
vcfl_threshold	R	1.05		Threshold for vertical advection CFL number at which the adaptive time step reduction (increase of ndyn_substeps w.r.t. the fixed fast-physics time step) is triggered.	

Parameter	Type	Default	Unit	Description	Scope
nlev_hcfl	I(max_dom)	0		Number of model levels (counted from the top) for which the horizontal CFL number is evaluated in addition and used for an adaptive dynamics time step reduction. In practice, doing this for the upper 10–15 levels is sufficient with a model top of 75 km.	
cfl_monitoring_freq	I	5		Monitoring frequency for CFL number (in units of fast-physics time steps of domain 1)	
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 at height $divdamp\_z$ and below. $divdamp\_fac \geq 0$.	
divdamp_fac2	R	0.004		Scaling factor for divergence damping at height $divdamp\_z2$. $divdamp\_fac2 \geq 0$. Between $divdamp\_z$ and $divdamp\_z2$ the scaling factor changes linearly from $divdamp\_fac$ to $divdamp\_fac2$.	
divdamp_fac3	R	0.004		Scaling factor for divergence damping at height $divdamp\_z3$. $divdamp\_fac3 \geq 0$. The three points $(divdamp\_z2, divdamp\_fac2)$, $(divdamp\_z3, divdamp\_fac3)$, and $(divdamp\_z4, divdamp\_fac4)$ determine the quadratic function for the scaling factor between $divdamp\_z2$ and $divdamp\_z4$.	
divdamp_fac4	R	0.004		Scaling factor for divergence damping at height $divdamp\_z4$ and higher. $divdamp\_fac4 \geq 0$.	
divdamp_z	R	32500.	m	Height up to which $divdamp\_fac$ is used, and where the linear profile up to height $divdamp\_z2$ starts.	
divdamp_z2	R	40000.	m	Height with scaling factor $divdamp\_fac2$ where the linear profile starting at $divdamp\_z$ ends, and where the quadratic profile up to $divdamp\_z4$ starts. $divdamp\_z < divdamp\_z2 < divdamp\_z4$.	

Parameter	Type	Default	Unit	Description	Scope
divdamp_z3	R	60000.	m	Height with scaling factor $divdamp\_fac3$. Needed to determine the quadratic function between $divdamp\_z2$ and $divdamp\_z4$. $divdamp\_z3 \neq divdamp\_z2 \wedge divdamp\_z3 \neq divdamp\_z4$.	
divdamp_z4	R	80000.	m	Height from which $divdamp\_fac4$ is used. $divdamp\_z4 > divdamp\_z2$.	
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)	
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	
divdamp_trans_start	R	12500.		Lower bound of transition zone between 2D and 3D divergence damping	divdamp_type = 32
divdamp_trans_end	R	17500.		Upper bound of transition zone between 2D and 3D divergence damping	divdamp_type = 32
iadv_rhotheta	I	2		Advection method for rho and rhotheta: 1: simple second-order upwind-biased scheme 2: 2nd order Miura horizontal	

Parameter	Type	Default	Unit	Description	Scope
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=5 .AND. lhdiff_temp = .true.
thslp_zdiffu	R	0.025		Slope threshold above which truly horizontal temperature diffusion is activated	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)	hdiff_order=5 .AND. lhdiff_temp=.true. .AND. l_zdiffu_t=.true.
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.	

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

2.30. nudging_nml

Parameters for the upper boundary nudging in the limited-area mode (grid_nml: l_limited_area = .TRUE.) or global nudging. 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(n_dom)	0		Nudging type: ∗ 0: none ∗ 1: upper boundary nudging ∗ 2: global nudging Please note: • nudge_type = 1 requires l_limited_area = .TRUE. • nudge_type = 1 is also applicable to nested domains. Nudging is performed against the same forcing data set for all domains. If nudging is enabled for one or more nested domains, it needs to be enabled for the base domain, as well. • nudge_type = 2 (global nudging) is applied in primary domain only • for global nudging the following settings in limarea_nml are mandatory: - itype_latbc = 1 (time-dependent driving data) - dtime_latbc = ... - latbc_path = "..." - latbc_boundary_grid = " " (no boundary grid: driving data have to be available on entire grid) - latbc_varnames_map_file = "..." (e.g., run/dict.latbc), if num_prefetch_proc = 1 (asynchronous read-in of driving data) • defaults and (additional) scopes for global nudging are marked by $(\cdot)_{\mathrm{glbndg}}$, if a parameter applies to both upper boundary and global nudging	run_nml:iforcing = 3 (NWP) ivctype = 2 (SLEVE)

Parameter	Type	Default	Unit	Description	Scope
max_nudge_coeff_vn	R	0.04 $(0.016)_{\mathrm{glbndg}}$	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: $v_n(t) = v_n^\star(t) + \text{nudge\_coeff\_vn}(z) * \text{ndyn\_substeps} * [\overline{v_n}(t) - v_n^\star(t)]$, where t and z denote time and height, respectively, $\overline{v_n}(t)$ is the target wind to nudge to, and $v_n^\star$ is the value before the nudging, the vertical profile of the coefficient for upper boundary nudging 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 $\text{max\_nudge\_coeff\_vn} \in [0, \sim 0.2]$, where the lower boundary is mandatory. **Please note** that the user value is internally multiplied by 5.	nudge_type > 0 (nudge_var $=$ "all" or "...,vn,...")$_{\mathrm{glbndg}}$
max_nudge_coeff_thermdyn	R	0.075 $(0.03)_{\mathrm{glbndg}}$	1	Max. nudging coefficient for the thermodynamic variables selected by limarea_nml: nudge_hydro_pres in case of upper boundary nudging and by thermdyn_type in case of global nudging. The range of validity is $\text{max\_nudge\_coeff\_thermdyn} \in [0, \sim 0.2]$, where the lower boundary is mandatory. **Please note** that the user value is internally multiplied by 5.	nudge_type > 0 (nudge_var $=$ "all" or "...,thermdyn,...")$_{\mathrm{glbndg}}$

Parameter	Type	Default	Unit	Description	Scope
max_nudge_coeff_qv	R	0.008	1	Max. nudging coefficient for water vapor. The range of validity is max_nudge_coeff_qv $\in [0, \sim 0.2]$, where the lower boundary is mandatory. (For global nudging only.) **Please note** that the user value is internally multiplied by 5.	nudge_type = 2 nudge_var = "all" or "...,qv,..."
nudge_start_height	R	12000 $(2000)_{\mathrm{glbndg}}$	m	Nudging is applied for: nudge_start_height $\leq z \leq$ top_height in case of upper boundary nudging and for: nudge_start_height $\leq z \leq$ nudge_end_height in case of global nudging, where z denotes the nominal height of the grid layer center, and top_height is the height of the model top (see sleve_nml). For upper boundary nudging the range of validity is nudge_start_height $\in [0, \mathrm{top\_height}]$, where both boundaries are mandatory. For global nudging a nudge_start_height in the range $[0, \mathrm{top\_height}]$ has to satisfy nudge_start_height < nudge_end_height. Values outside $[0, \mathrm{top\_height}]$ will be interpreted as nudge_start_height = 0.	nudge_type > 0
nudge_end_height	R	40000	m	Nudging is applied for: nudge_start_height $\leq z \leq$ nudge_end_height, where z denotes the nominal height of the grid layer center. A nudge_end_height in the range $[0, \mathrm{top\_height}]$ has to satisfy nudge_start_height < nudge_end_height. Values outside $[0, \mathrm{top\_height}]$ will be interpreted as nudge_start_height = top_height. (For global nudging only.)	nudge_type = 2

Parameter	Type	Default	Unit	Description	Scope
nudge_profile	I	4		Vertical profile of the nudging coefficient (nudging strength) between nudge_start_height and nudge_end_height: ∗ 1: squared scaled vertical distance from nudge_start_height (this is the profile used for upper boundary nudging) ∗ 2: constant profile ∗ 3: hyperbolic tangent profile ∗ 4: trapezoidal profile The profile values range from 0 to 1. A multiplication with max_nudge_coeff_vn/thermdyn/qv and ndyn_substeps yields the final value of the nudging coefficient. (For global nudging only.)	nudge_type = 2
nudge_scale_height	R	3000	m	Scale height of nudging profile. (For global nudging only.)	nudge_type = 2 nudge_profile = 3 or 4
nudge_var	C	"all"		Select the variables that shall be nudged: ∗ "vn": horizontal wind ∗ "thermdyn": thermodynamic variables ∗ "qv": water vapor ∗ comma-separated list: e.g., "vn,thermdyn" ∗ "all": all available variables (i.e. equivalent to "vn,thermdyn,qv") Please note that the nudging of water vapor requires ltransport = .TRUE. (For global nudging only.)	nudge_type = 2
thermdyn_type	I	1		Set of variables used to compute the thermodynamic nudging increments: ∗ 1: hydrostatic set (pressure and temperature) ∗ 2: non-hydrostatic set (density and virtual potential temperature)	nudge_type = 2 nudge_var = "all" or "...,thermdyn,..."

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

2.31. 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: cloud ice scheme (based on COSMO-EU microphysics, 2-cat ice: cloud ice, snow) 2: graupel scheme (based on COSMO-DE microphysics, 3-cat ice: cloud ice, snow, graupel) 3: Two-moment cloud ice scheme 4,5,6,7: Two-moment microphysics of ICON-RUC, further configuration possible in namelist `/twomom_mcrph_nml/` 8: Spectral Bin Microphysics (SBM) by A. Khain 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
lvariable_rain_n0	L	.FALSE.		variable intercept parameter of raindrop size distribution: the multiplicative factor rain_n0_factor is used for drizzle (small q_r) while the default value is approached for heavy rain (large q_r)	inwp_gscp=2
mu_snow	R	0.0		shape parameter in gamma distribution for snow	inwp_gscp>0

Parameter	Type	Default	Unit	Description	Scope
icpl_aero_gscp	I	0		0: off 1: simple coupling between autoconversion and Tegen aerosol climatology; requires irad_aero=6, 7 or 9 3: use cloud-droplet number climatology from external parameter file. External parameter files containing cloud-droplet number climatology can be generated with extpar code from version rc_5.14 More advanced options are in preparation	currently only for inwp_gscp = 1
lscale_cdnc	L	.FALSE.		.TRUE.: scaling of external (MODIS) cdnc with a time varying 2-dimensional factor derived from the simple plume scheme. It is advisable to turn the scaling off at least for the years 2000-2020.	climate projections using icpl_aero_gscp = 3 irad_aero = 18,19
icpl_aero_ice	I	0		0: ice crystal concentration defined by temperature using Cooper (1987) formula 1: ice nucleating particles concentration defined by temperature and dust concentration using DeMott (2015) formula 2: ice nucleating particles parameterization of Ullrich et al. (2017) 3: prognostic mineral dust used as ice nucleating particles (dust number only) 4: prognostic mineral dust used as ice nucleating particles (dust number and size)	inwp_gscp= 1, 2 inwp_gscp=1, 2 and irad_aero=6, 7 or 8 inwp_gscp=3 inwp_gscp=3 and ART inwp_gscp=3 and ART
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; cannot be combined with lgrayzone_deepconv
lgrayzone_deepconv	L (max_dom)	.FALSE.		.TRUE.: activates shallow and deep convection but not mid-level convection, together with some tuning measures targeted at grayzone (convection-permitting) model resolutions	inwp_convection = 1; cannot be combined with lshallowconv_only

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
lstoch_expl	L (max_dom)	.FALSE.		.TRUE.: activate explicit stochastic shallow convection scheme EXPERIMENTAL! will not produce clean restart to be used in conjunction with lrestune_off=.T. and lmflimiter_off=.T.	inwp_convection = 1
lstoch_sde	L (max_dom)	.FALSE.		.TRUE.: activate stochastic differential equation (SDE) shallow convection scheme to be used in conjunction with lrestune_off=.T. and lmflimiter_off=.T.	inwp_convection = 1
lstoch_deep	L (max_dom)	.FALSE.		.TRUE.: activate stochastic differential equation (SDE) deep convection scheme	inwp_convection = 1
lrestune_off	L (max_dom)	.FALSE.		.TRUE.: switches off resolution-dependent tuning of shallow convection parameters	inwp_convection = 1
lmflimiter_off	L (max_dom)	.FALSE.		.TRUE.: disables mass flux limiter by setting it to high values that are rarely reached by shallow convection	inwp_convection = 1
lvvcouple	L (max_dom)	.FALSE.		.TRUE.: use vertical velocity at 650hPa as criterion to couple shallow convection with resolved deep convection	inwp_convection = 1
lvv_shallow_deep	L (max_dom)	.FALSE.		.TRUE.: use vertical velocity at 650hPa to distinguish between shallow and deep convection within convection routines (instead of cloud depth)	inwp_convection = 1
lstoch_spinup	L (max_dom)	.FALSE.		.TRUE.: spin up cloud ensemble to equilibrium in stochastic shallow convection schemes, only takes effect when lstoch_expl=T or lstoch_sde=T	inwp_convection = 1

Parameter	Type	Default	Unit	Description	Scope
nclds	I (max_ dom)	5000		maximum possible number of shallow clouds per grid box in explicit stochastic cloud ensemble. only takes effect when lstoch_expl=T	inwp_convection = 1
icpl_aero_conv	I	0		0: off 1: simple coupling between autoconversion and Tegen aerosol climatology; requires irad_aero=6 or 7	
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. Requires fields emi_bc, emi_oc and emi_so2 in the extpar dataset 3: as option 2, but including wildfire emissions for bc, oc and so2. Emission data set can be specified with fire2d_filename (&initicon_nml)	irad_aero=6.
icpl_o3_tp	I	1		0: off 1: simple coupling between the ozone mixing ratio and the thermal tropopause, restricted to the extratropics 2: improved variant of 1 that avoids excessive additional ozone for low tropopauses	irad_o3 = 7 or 9
itype_dissip_heat	I	1		Options for calculating dissipative heating in NWP interface 0: off; setting turbdiff_nml:ltmpcor = .TRUE. forces reset to 0 because dissipative heating is then calculated in turbdiff 1: dissipative heating from SSO, GWD and Rayleigh friction 2: as 1, plus dissipative heating from momentum dissipation by turbulence	turbdiff_nml:ltmpcor = .FALSE.

Parameter	Type	Default	Unit	Description	Scope
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
lsgs_cond	L (max_dom)	.TRUE.		Apply subgrid-scale condensational heating related to the non-convective part of diagnosed cloud water	inwp_cldcover = 1
lsbm_warm_full	L (max_dom)	.TRUE.		TRUE: warm-phase Spectral Bin Microphysics (SBM). False: 2-moment scheme is applied (as for inwp_gscp = 4) and SBM is running in a "Piggy Backing" mode for diagnostic purposes	inwp_gscp = 8
inwp_radiation	I (max_dom)	1		radiation 0: none 1: RRTM radiation 2: (removed) 3: (removed) 4: ecRad radiation	run_nml:iforcing = inwp
inwp_satad	I	1		saturation adjustment 0: none 1: saturation adjustment at constant density	run_nml:iforcing = inwp
inwp_turb	I (max_dom)	1		vertical diffusion and transfer 0: none 1: COSMO diffusion and transfer 2: GME turbulence scheme 5: Classical Smagorinsky diffusion 6: VDIFF turbulence scheme (requires inwp_surface = 2)	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

Parameter	Type	Default	Unit	Description	Scope
inwp_surface	I (max_dom)	1		surface scheme 0: none 1: TERRA 2: JSBACH (requires inwp_turb = 6)	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
itype_satpres_coeffs	I	1		Set of coefficients used for computing the saturation vapor pressure: 1 = Coefficients inherited from the COSMO model (inaccurate at temperatures below −50°C) 2 = Coefficients used in the IFS (and the parameterizations taken over from the IFS) inwp_satad > 0	
dt_conv	R (max_dom)	600.	s	time interval of convection call. by default, each subdomain has the same value	run_nml:iforcing = inwp
dt_ccov	R (max_dom)	dt_conv	s	time interval of cloud-cover call. by default, dt_ccov equals dt_conv for each domain	run_nml:iforcing = inwp
dt_rad	R (max_dom)	1800.	s	time interval of radiation call by default, each subdomain has the same value	run_nml:iforcing = inwp
dt_sso	R (max_dom)	1200.	s	time interval of sso call by default, each subdomain has the same value	run_nml:iforcing = inwp

Parameter	Type	Default	Unit	Description	Scope
dt_gwd	R (max_dom)	1200.	s	time interval of gwd call by default, each subdomain has the same value	run_nml:iforcing = inwp
lrtm_filename	C(:)	"rrtmg_ lw.nc"		NetCDF file containing longwave absorption coefficients and other data for RRTMG_LW k-distribution model.	
cldopt_filename	C(:)	"ECHAM6_CldOptProps.nc"		NetCDF file with RRTM Cloud Optical Properties for ECHAM6.	
icalc_reff	I (max_dom)	0		Parameterization set for diagnostic calculations of effective radius: 0 = No calculation 1,2,4,5,6,7 = Consistent with microphysics given by icalc_reff (naming same convention as inwp_gscp) 100 = Consistent with current microphysics (it sets icalc_reff = inwp_gscp) 101 = Reff given by RRTM parameterization	run_nml:iforcing = inwp
icpl_rad_reff	I (max_dom)	0		Coupling of the effective radius with radiation: 0 = No coupling. The calculation of the effective radius happens at the radiation interface. 1 = Radiation uses the effective radius defined by icalc_reff. All hydrometeors are combined in a frozen and a liquid phase. 2 = Radiation uses the effective radius defined by icalc_reff for all hydrometeors independently (given by ecrad_iset_genhyd).	run_nml:iforcing = inwp inwp_radiation = 1 or 4 icalc_reff > 0
ithermo_water	I (max_dom)	0		Latent Heat Function 0 = Temperature-dependent latent heat in saturation adjustment but constant in microphysics: 1 = Temperature-dependent latent heat in saturation adjustment and microphysics	run_nml:iforcing = inwp inwp_gscp = 1,2,4,5,7

Parameter	Type	Default	Unit	Description	Scope
lupatmo_phy	L (max_ dom)	.FALSE.		Switch for upper-atmosphere physics. Examples of usage for multi-domain applications: • set lupatmo_phy = .TRUE. to switch on upatmo physics for all domains • set lupatmo_phy = .TRUE., .TRUE., .FALSE. to switch on upatmo physics for dom 1 and 2, but switch them off for dom 3 • please note that “skipping” domains is currently not possible, i.e. lupatmo_phy = .TRUE., .FALSE., .TRUE. is transformed into lupatmo_phy = .TRUE., .FALSE., .FALSE. See upatmo_nml for configuration of the upper-atmosphere physics parameterizations.	run_nml:iforcing = inwp init_mode < 4 inwp_turb > 0 inwp_radiation > 0
lcuda_graph_turb_tran	L	.FALSE.		Activate cuda graphs for turbulent transfers. Automatically set to .FALSE. if not compiled with the ICON_USE_CUDA_GRAPH cpp key.	ICON_USE_CUDA_GRAPH activated

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

2.31.1. Notes on use of stochastic convection schemes

There are currently three stochastic convection schemes available, two versions for shallow convection and one for deep convection. Conceptually, these schemes attempt to represent that for grid box sizes smaller than the size of a typical cloud ensemble, the clouds actually populating the grid box will not be fully representative of that cloud ensemble. The two stochastic shallow schemes (lstoch_expl, lstoch_sde) are therefore aimed at resolutions of a few kilometers (typically used for LAM simulations, where deep convection is resolved) and will in fact be automatically switched off for resolutions greater than 20km. The scheme converges to the standard Tiedtke-Bechtold mass flux scheme at resolutions sufficiently coarse, such that there is no additional gain from using the stochastic schemes. They should therefore be run with lshalloconv_only=T. A combination with the grayzone tuning (lgrayzone_deepconv) is technically possible, but not recommended as the grayzone tuning interferes with the intended behaviour of the stochastic scheme.
The stochastic deep convection scheme (lstoch_deep) is intended for resolutions where the deep convection parameterization is still active, but again, grid size is not large enough to contain a fully representative cloud ensemble (e.g. global runs with resolution on the order of 10s of kilometers). Thus the deep and shallow stochastic schemes are not intended to be used together, as the resolutions they are designed for are (mostly) mutually exclusive.
The shallow schemes should be run without resolution-dependent tuning of the convection parameters (lrestune_off=T) and with disabled mass flux limiters (lmflimiter_off=T). The mass flux limiters are in fact not fully disabled but set to values high enough to be rarely reached during shallow cloud simulations. The

deep stochastic scheme cannot be run without mass flux limiters or simulatons will become unstable.

2.32. 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_gkdrag_enh	R (max_dom)	0.075		enhanced value of gravity wave drag constant at low latitudes (needs to be actively set to a larger value than gkdrag to be effective)	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
tune_grcrit_enh	R (max_dom)	0.25		enhanced value of critical Richardson number at low latitudes (needs to be actively set to a larger value than grcrit to be effective)	run_nml:iforcing = inwp
tune_minsso	R (max_dom)	10.	m	minimum SSO standard deviation for which SSO scheme is applied	run_nml:iforcing = inwp
tune_minsso_gwd	R (max_dom)	0.	m	minimum SSO standard deviation for which wave drag component if SSO scheme is applied (effective only if larger than minsso; the default of zero means that the parameter needs to be actively set)	run_nml:iforcing = inwp
tune_blockred	R (max_dom)	100.		multiple of SSO standard deviation above which blocking tendency is reduced	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
tune_gcstar	R	1.0		constant in saturation wave spectrum	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

Parameter	Type	Default	Unit	Description	Scope
tune_icesedi_exp	R	0.33		Exponent for density correction of cloud ice sedimentation	run_nml:iforcing = inwp
tune_zcsg	R	0.5		Efficiency for snow-to-graupel conversion by riming	inwp_gscp=2
tune_sbmccn	R	1.0		Scaling factor (0,1] for initial aerosol concentration profile, used for comparison between simulations with two-moment and warm SBM microphysics	run_nml:iforcing = inwp
Convection scheme					
tune_entrorg	R	1.95e-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_capethresh	R	7000.0	J/kg	CAPE threshold above which the convective adjustment time scale and entrainment rate are reduced for numerical stability	run_nml:iforcing = inwp
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
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_texc	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

Parameter	Type	Default	Unit	Description	Scope
tune_grzdc_offset	R	0.0		Scaling factor for offset in CAPE closure for grayzone deep convection. Positive values reduce the activity of the convection scheme and suppress convective drizzle (recommendation: 0.1–0.2)	run_nml:iforcing = inwp; lgrayzone_deepconv = .TRUE.
tune_box_liq	R	0.05		Box width scale for liquid cloud diagnostic in cloud cover scheme	run_nml:iforcing = inwp; inwp_cldcover = 1
tune_box_ice	R	0.05		Box width scale for ice cloud diagnostic in cloud cover scheme	run_nml:iforcing = inwp; inwp_cldcover = 1
tune_thicklayfac	R	0.005	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
tune_box_liq_sfc_fac	R (max_dom)	1.0		Tuning factor for box_liq reduction near the surface	run_nml:iforcing = inwp; inwp_cldcover = 1
allow_overcast	R	1.0		Tuning factor for the dependence of liquid cloud cover on relative humidity. This is an unphysical ad-hoc parameter to improve the cloud cover in the Mediterranean	run_nml:iforcing = inwp; inwp_cldcover = 1
tune_sgsclifac	R	0.0		Scaling factor for parameterization of subgrid-scale (turbulence-induced) cloud ice (values > 0 not recommended for global configurations with RRTM radiation)	run_nml:iforcing = inwp; inwp_cldcover = 1
icpl_turb_clc	I	1		Mode of coupling between turbulence and cloud cover 1: strong dependency of box width on rcld with upper and lower limit 2: weak dependency of box width on rcld with additive term and upper limit	run_nml:iforcing = inwp; inwp_cldcover = 1
lcalib_clcov	L	.TRUE.		Apply calibration of layer-wise cloud cover diagnostics over land in order to improve scores against SYNOP reports	run_nml:iforcing = inwp
max_calibfac_clcl	R	4.0		Maximum allowed calibration factor for low clouds (CLCL)	run_nml:iforcing = inwp
tune_eiscrit	R	1000.0	K	Critical estimated inversion strength above which to switch off shallow convection (recommendation to activate: 7K)	run_nml:iforcing = inwp; inwp_convection = 1

Parameter	Type	Default	Unit	Description	Scope
tune_sc_eis	R	1000.0	K	Critical estimated inversion strength above which to use modified SGS cloud diagnostic for stratocumulus (recommendation to activate: 7K)	run_nml:iforcing = inwp; inwp_clcover = 1
tune_sc_invmin	R	200.0	m	Minimum inversion height above which to apply modified SGS cloud diagnostic for stratocumulus	run_nml:iforcing = inwp; inwp_clcover = 1
tune_sc_invmax	R	1500.0	m	Maximum inversion height below which to apply modified SGS cloud diagnostic for stratocumulus	run_nml:iforcing = inwp; inwp_clcover = 1
tune_supsat_limfac	R	0.0		Limiting factor for parameterized supersaturation in updrafts during the saturation adjustment (0.0 for thermodynamic equilibrium)	run_nml:iforcing = inwp; inwp_satad = 1

Misc					
tune_gust_factor	R	8.0		Multiplicative factor for friction velocity in gust parameterization	run_nml:iforcing = inwp
tune_gustsso_lim	R	100.0	m s^{-1}	Basic gust speed at which the SSO correction starts to be reduced (recommendation to activate: 20 m s^{-1}	run_nml:iforcing = inwp
itune_gust_diag	I	1		Method of SSO blocking correction used in the gust diagnostics 1: Use level above "SSO envelope top" for gust enhancement over mountains 2: Use "SSO envelope top" level for gust enhancement over mountains, combined with an adjusted nonlinearity factor (recommended for global configurations with MERIT/REMA orography) 3: Variant of option 1, recommended for ICON-D2 with subgrid-scale condensation (do not use with ntiles=1) 4: As 3, but using time-averaged 10-m wind speeds as input with additional limitation to resolved bouldary-layer wind speeds	run_nml:iforcing = inwp; related switches are tune_gustlim_agl and tune_gustlim_fac
tune_gustlim_agl	R(max_dom)	1500.0	m	Height range above ground, within which the maximum resolved wind speed is determined for gust limitation	itune_gust_diag = 4

Parameter	Type	Default	Unit	Description	Scope
tune_gustlim_fac	R(max_dom)	0.0		Tuning factor for gust limitation. The default value of zero means that the limitation is turned off. Otherwise, the difference between the 10-m wind speed and the maximum speed found below tune_gustlim_agl times tune_gustlim_fac is used to limit the excess gust speed	itune_gust_diag = 4
itune_vis_diag	I	1		Tuning variant of visibility diagnostics 1: First operational implementation 2: Optimized day-night conversion factor	run_nml:iforcing = inwp
itune_albedo	I	0		MODIS albedo tuning 0: None 1: dimmed sahara 2: dimmed sahara + brightened Antarctic (by 5%)	run_nml:iforcing = inwp albedo_type=2
tune_albedo_wso	R(2)	0.0, 0.0	-	Add a correction to MODIS albedo over [dry,wet] soil for soil types 3-6. Valid range: [-0.03, 0.03]. Supposed to be negative for wet soil.	run_nml:iforcing = inwp nwp_phy_nml:inwp_surface = 1 itype_albedo = 2 direct_albedo = 3,4
itune_slopecorr	I	0		Tuning measures for high-resolution configurations with mesh sizes around or below 1 km 0: None 1: Slope-dependent reduction of laminar transfer resistance and near-surface minimum vertical diffusion	run_nml:iforcing = inwp inwp_turb=1
itune_o3	I	2		Ozone tuning 0: no tuning 1: old tuning for RRTM radiation 2: standard tuning for EcRad with RRTM gas optics 3: updated variant of 2 for combination with icpl_o3_tp = 2 4: provisional tuning for EcRad with EcCKD gas optics	run_nml:iforcing = inwp irad_o3=7, 79 or 97
tune_difrad_3dcont	R	0.5		Tuning factor for 3D contribution to diagnosed diffuse radiation (no impact on prognostic results!)	inwp_radiation = 1 or 4

Parameter	Type	Default	Unit	Description	Scope
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
tune_dursun_scaling	R	1.0		Tuning factor for direct solar irradiance in sunshine duration diagnostic to account for the delta-Eddington scaling in ecRad and other possible biases (e.g. liquid/ice water path)	
tune_urbahf	R(4)	0., 2., 2., 50.	W m^{-2}	Tuning factors for specifying the anthropogenic heat flux (AHF) depending on climatological T2M. first value: constant base value independent of temperature second value: gradient per K below T2M = 15°C for heating third value: gradient per K above T2M = 20°C for cooling fourth value: upper limit for AHF	lterra_urb=.TRUE.
tune_urbisa	R(2)	0.6, 1.0		Lower and upper bound for variable ISA parameterization (fraction of impervious surface area on urban tiles) depending on smoothed urban fraction	lterra_urb=.TRUE.
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/namelists/mo_nwp_tuning_nml.f90`

2.33. twomom_mcrph_nml

This namelist offers the possibility to adapt some configuration parameters of the two-moment cloud microphysical parameterisation by A. Seifert and K.D. Beheng. It is only effective if this scheme is used, i.e., if **inwp_gscp=4, 5, 6, or 7** in namelist `/nwp_phy_nml/`.

The below set of parameters is a first reasonable choice to start with something. There might be coming more parameters in the future.

Please note: at the moment we do not support the option to have different configuration parameters on different domains. We did not really test this up to now, but it cannot be ruled out for the future. There are for sure parameters which could be optimized for different resolutions. Therefore, at the moment this possibility is not provided explicitly to the user via the below namelist parameters (they are scalars), but it is prepared internally in the ICON code.

Parameter	Type	Default	Unit	Description	Scope
i2mom_solver	I	1	-	Type of numerical time integration scheme for the two-moment scheme: 0: explicit Euler-forward 1: semi-implicit solver similar to that of the standard one-moment schemes	iforcing=3, inwp_gscp=4
ccn_type	I	Depends on inwp_gscp: for 4,7: 7 for 5: 8	-	Choice of the aerosol scenario for cloud nucleation (CCN): 6: “low CCN” (“maritime”) 7: “intermediate CCN” 8: “high CCN” (“continental”) 9: “very high CCN” (“polluted”) If applied together with the ART aerosol physics **inwp_gscp=6, this parameter has no effect.**	iforcing=3, inwp_gscp=4,5,7
ccn_Ncn0	R	-999.99	m^{-3}	CN concentration near ground. A value of < -900 indicates that the hardcoded value associated with the ccn_type will be used. If applied together with the ART aerosol physics **inwp_gscp=6, this parameter has no effect.**	iforcing=2,3 inwp_gscp=4,5,7
ccn_wcb_min	R	0.1	m^{-3}	Minimum updraft speed for Segal&Khain cloud nucleation parameterization. If applied together with the ART aerosol physics **inwp_gscp=6, this parameter has no effect.**	iforcing=2,3 inwp_gscp=4,5,7
iicephase	I	1	-	Turning on/off mixed phase processes in the two-moment scheme: 0: warm phase processes only 1: mixed phase processes	iforcing=3, inwp_gscp=4,5,6,7
alpha_spacefilling	R	0.01	-	Parameter in conversion of snow or cloud ice to graupel by riming: degree of void filling by frozen supercooled droples within the ice particle skeleton, above which the particle is converted to the graupel class. Smaller values lead to faster conversion. 0.01 means very fast conversion to graupel. A value of 0.68 is the theoretical limit for densest sphere packing and leads to rather slow conversion.	iforcing=2,3 inwp_gscp=4,5,6,7
D_conv_ii	R	75.0e-6	m	diameter threshold for the onset of conversion to snow by ice selfcollection	iforcing=2,3 inwp_gscp=4,5,6,7

Parameter	Type	Default	Unit	Description	Scope
D_rainfrz_ig	R	0.50e-3	m	Spectral size threshold below which freezing rain drops are converted to cloud ice. Larger drops are converted to graupel or hail, depending on parameter D_rainfrz_gh.	iforcing=2,3 inwp_gscp=4,5,6,7
D_rainfrz_gh	R	1.25e-3	m	Spectral size threshold above which freezing rain drops are converted to hail. Smaller drops are converted to cloud ice or graupel, depending on parameter D_rainfrz_ig.	iforcing=2,3 inwp_gscp=4,5,6,7
luse_mu_Dm_rain	L	.FALSE.		To switch on the usage of the dynamical μ-D_M-relation for raindrops below cloud base.	iforcing=2,3 inwp_gscp=4,5,6,7
rain_cmu0	R	6.0	-	Parameter of the μ-D-relation in the rain size distribution for evaporation and sedimentation below cloud base: asymptotic μ-value for spectra with small mean diameter.	iforcing=2,3 inwp_gscp=4,5,6,7
rain_cmu1	R	30.0	-	Parameter of the μ-D-relation in the rain size distribution for evaporation and sedimentation below cloud base: asymptotic μ-value for spectra with large mean diameter.	iforcing=2,3 inwp_gscp=4,5,6,7
rain_cmu3	R	1.1e-3	m	Parameter of the μ-D-relation in the rain size distribution for evaporation and sedimentation below cloud base: equilibrium mean spectral diameter for breakup and selfcollection.	iforcing=2,3 inwp_gscp=4,5,6,7
rain_cmu4	R	1.0	-	Parameter of the μ-D-relation in the rain size distribution for evaporation and sedimentation below cloud base: base value of μ.	iforcing=2,3 inwp_gscp=4,5,6,7
in_fact	R	1.0		Factor to tune the IN concentration for heterogeneous ice nucleation	iforcing=2,3 inwp_gscp=4,5,6,7
nu_i	R	-999.99	-	Shape parameter ν for cloud ice in the PSD $f(x) = N_0 x^\nu \exp(-\lambda x^\mu)$ A value of $<$ -900 indicates that the background value $\nu = 0.0$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7

Parameter	Type	Default	Unit	Description	Scope
mu_i	R	-999.99	-	Shape parameter μ for cloud ice in the PSD $f(x) = N_0 x^\nu \exp(-\lambda x^\mu)$ A value of < -900 indicates that the background value $\mu = 1/3$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
ageo_i	R	-999.99		Prefactor of the assumed size-mass-relation for cloud ice $D = a_{geo} x^{b_{geo}}$ for x in kg and D in m. A value of < -900 indicates that the background value $a_{geo} = 0.835$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
bgeo_i	R	-999.99	-	Exponent of the assumed size-mass-relation for cloud ice $D = a_{geo} x^{b_{geo}}$ for x in kg and D in m. A value of < -900 indicates that the background value $b_{geo} = 0.39$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
avel_i	R	-999.99		Prefactor of the assumed fallspeed-mass-relation for cloud ice $v = a_{vel} x^{b_{vel}}$ for x in kg and v in m/s. A value of < -900 indicates that the background value $a_{vel} = 27.7$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
bvel_i	R	-999.99	-	Exponent of the assumed fallspeed-mass-relation for cloud ice $v = a_{vel} x^{b_{vel}}$ for x in kg and v in m/s. A value of < -900 indicates that the background value $b_{vel} = 0.21579$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
nu_s	R	-999.99	-	Shape parameter ν for snow in the PSD $f(x) = N_0 x^\nu \exp(-\lambda x^\mu)$ A value of < -900 indicates that the background value $\nu = 0.0$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
mu_s	R	-999.99	-	Shape parameter μ for snow in the PSD $f(x) = N_0 x^\nu \exp(-\lambda x^\mu)$ A value of < -900 indicates that the background value $\mu = 0.5$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7

Parameter	Type	Default	Unit	Description	Scope
ageo_s	R	-999.99		Prefactor of the assumed size-mass-relation for snow $D = a_{geo} x^{b_{geo}}$ for x in kg and D in m. A value of $<$ -900 indicates that the background value $a_{geo} = 5.13$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
bgeo_s	R	-999.99	-	Exponent of the assumed size-mass-relation for snow $D = a_{geo} x^{b_{geo}}$ for x in kg and D in m. A value of $<$ -900 indicates that the background value $b_{geo} = 1/2$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
avel_s	R	-999.99		Prefactor of the assumed fallspeed-mass-relation for snow $v = a_{vel} x^{b_{vel}}$ for x in kg and v in m/s. A value of $<$ -900 indicates that the background value $a_{vel} = 400.0$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
bvel_s	R	-999.99	-	Exponent of the assumed fallspeed-mass-relation for snow $v = a_{vel} x^{b_{vel}}$ for x in kg and v in m/s. A value of $<$ -900 indicates that the background value $b_{vel} = 0.35$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
nu_r	R	-999.99	-	Shape parameter ν of the rain mass distribution inside clouds. Refers to the generalized gamma distribution with respect to mass x: $f(x) = N_0 x^{\nu} \exp(-\lambda x^{\mu})$ A value $<$ -900 indicates that the background value $\nu = 0.0$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
nu_g	R	-999.99	-	Shape parameter ν for graupel in the PSD $f(x) = N_0 x^{\nu} \exp(-\lambda x^{\mu})$ A value of $<$ -900 indicates that the background value $\nu = 1.0$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7

Parameter	Type	Default	Unit	Description	Scope
mu_g	R	-999.99	-	Shape parameter μ for graupel in the PSD $f(x) = N_0 x^\nu \exp(-\lambda x^\mu)$ A value of $<$ -900 indicates that the background value $\mu = 1/3$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
ageo_g	R	-999.99		Prefactor of the assumed size-mass-relation for graupel $D = a_{geo} x^{b_{geo}}$ for x in kg and D in m. A value of $<$ -900 indicates that the background value $a_{geo} = 0.124$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
bgeo_g	R	-999.99	-	Exponent of the assumed size-mass-relation for graupel $D = a_{geo} x^{b_{geo}}$ for x in kg and D in m. A value of $<$ -900 indicates that the background value $b_{geo} = 0.314$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
avel_g	R	-999.99		Prefactor of the assumed fallspeed-mass-relation for graupel $v = a_{vel} x^{b_{vel}}$ for x in kg and v in m/s. A value of $<$ -900 indicates that the background value $a_{vel} = 100.0$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
bvel_g	R	-999.99	-	Exponent of the assumed fallspeed-mass-relation for graupel $v = a_{vel} x^{b_{vel}}$ for x in kg and v in m/s. A value of $<$ -900 indicates that the background value $b_{vel} = 0.34$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
nu_h	R	-999.99		Shape parameter ν for hail in the PSD $f(x) = N_0 x^\nu \exp(-\lambda x^\mu)$ A value of $<$ -900 indicates that the background value $\nu = 1.0$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
mu_h	R	-999.99	-	Shape parameter μ for hail in the PSD $f(x) = N_0 x^\nu \exp(-\lambda x^\mu)$ A value of $<$ -900 indicates that the background value $\mu = 1/3$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7

Parameter	Type	Default	Unit	Description	Scope
ageo_h	R	-999.99		Prefactor of the assumed size-mass-relation for hail $D = a_{geo} x^{b_{geo}}$ for x in kg and D in m. A value of $<$ -900 indicates that the background value $a_{geo} = 0.1366$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
bgeo_h	R	-999.99	-	Exponent of the assumed size-mass-relation for hail $D = a_{geo} x^{b_{geo}}$ for x in kg and D in m. A value of $<$ -900 indicates that the background value $b_{geo} = 1/3$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
avel_h	R	-999.99		Prefactor of the assumed fallspeed-mass-relation for hail $v = a_{vel} x^{b_{vel}}$ for x in kg and v in m/s. A value of $<$ -900 indicates that the background value $a_{vel} = 39.3$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
bvel_h	R	-999.99	-	Exponent of the assumed fallspeed-mass-relation for hail $v = a_{vel} x^{b_{vel}}$ for x in kg and v in m/s. A value of $<$ -900 indicates that the background value $b_{vel} = 1/6$ from mo_2mom_mcrph_main.f90 is used.	iforcing=2,3 inwp_gscp=4,5,6,7
melt_h_tune_fac	R	1.0	-	Tuning factor for the hail melting rate. Values larger than 1.0 enhance the hail melting, smaller values slow down the melting.	iforcing=2,3 inwp_gscp=4,5,6,7
melt_g_tune_fac	R	1.0	-	Tuning factor for the graupel melting rate. Values larger than 1.0 enhance the graupel melting, smaller values slow down the melting.	iforcing=2,3 inwp_gscp=4,5,6,7
Tmax_gr_rime	R	270.16	K	Graupel formation by riming of snow and cloud ice is only active below this temperature threshold.	iforcing=2,3 inwp_gscp=4,5,6,7
lturb_enhc	L	.TRUE.		To switch on the turbulent enhancement of collision processes involving water droplets (autoconversion, accretion, rain selfcollection).	iforcing=2,3 inwp_gscp=4,5,6,7

Parameter	Type	Default	Unit	Description	Scope
lturb_len	R	300	m	Turbulent length scale used for lturb_enhc=.TRUE.	iforcing=2,3 inwp_gscp=4,5,6,7 lturb_enhc=.TRUE.
iice_stick	I	10	-	Optional choice of the sticking efficiency parameterization for ice-ice-collisions as function of temperature: 1: $e = \min(\exp(0.09(T-T_3)), 1.0)$ from Lin et al. (1983) 2: $e = \min(\exp(0.025(T-T_3)), 1.0)$, even larger as option 1 3: same as 2, but $e = 0.01$ for $T < -40°\mathrm{C}$ 4: piecewise constant sticking efficiency with maxima at $-7.5°\mathrm{C}$ and $-15°\mathrm{C}$ and much smaller values at warm temperatures compared to 1, 2 and 3 5: piecewise linear sticking efficiency with maximum at $-15°\mathrm{C}$ and smaller values elswhere compared to 5 6: option 5 times factor 0.5 7: constant $e = 0.1$ 8: piecewise linear, a mix of 4 and 5 9: option 5 times factor 0.75 10: $e = \min(10^{(0.035(T-T_3)-0.7)}, 0.2)$ from Cotton et al. (1986) with $T_3 = 273.16\,\mathrm{K}$	iforcing=2,3 inwp_gscp=4,5,6,7
isnow_stick	I	5	-	Optional choice of the sticking efficiency parameterization for snow-snow collisions as function of temperature. Same choices as for `iice_stick`.	iforcing=2,3 inwp_gscp=4,5,6,7
iparti_stick	I	5	-	Optional choice of the sticking. efficiency parameterization for other frozen category collisions as function of temperature. Same choices as for `iice_stick`. Does not apply for graupel selfcollection. There, the below `ecoll_gg`, `ecoll_gg_wet` and `Tcoll_gg_wet` apply.	iforcing=2,3 inwp_gscp=4,5,6,7
ecoll_gg	R	0.1	-	Collision efficiency for graupel autoconversion (dry graupel). Value between 0 and 1.	iforcing=2,3 inwp_gscp=4,5,6,7

Parameter	Type	Default	Unit	Description	Scope
ecoll_gg_wet	R	0.4	-	Collision efficiency for graupel autoconversion (wet graupel). Value between 0 and 1.	iforcing=2,3 inwp_gscp=4,5,6,7
Tcoll_gg_wet	R	270.16	K	Temperature limit above which graupel autoconversion is considered to be for wet surfaces.	iforcing=2,3 inwp_gscp=4,5,6,7
cap_ice	R	-999.99	-	Capacitance for clould ice depositional growth. A value $<$ -900 indicates the usage of the code-internal backgroud value 3.0.	iforcing=2,3 inwp_gscp=4,5,6,7
cap_snow	R	-999.99	-	Capacitance for snow depositional growth. A value $<$ -900 indicates the usage of the code-internal backgroud value 3.0.	iforcing=2,3 inwp_gscp=4,5,6,7
vsedi_max_s	R	-999.99	m/s	Maximum allowed spectral mean sedimentation velocity of snow at sea level. A value $<$ -900 indicates the usage of the code-internal backgroud value 1.2 m/s.	iforcing=2,3 inwp_gscp=4,5,6,7
itype_shedding_gh	I	0	-	Choice of shedding parameterization for graupel and hail during collision processes with water droplets: 0=no shedding, 1=simpe, 2=more physical. If applied together with the ART aerosol physics **inwp_gscp=6, only options 0 and 1 are currently supported.**	iforcing=2,3 inwp_gscp=4,5,6,7
D_shed_gh	R	0.009	m	Critical graupel/hail particle diameter for shedding during riming (wet growth) and melting. Shedding happens if: itype_shedding_gh = 1: $D_{meanmass}$ >D_shed_gh itype_shedding_gh = 2: in the spectral PSD-part where $D >$ max(D_{wetgr}, D_shed_gh) - that is for wet growth but not below a stable diameter, e.g., 9 mm after Rasmussen and Heymfield	itype_shedding_gh=1,2 iforcing=2,3 inwp_gscp=4,5,6,7
llim_gr_prod_rain_riming	L	.FALSE.	-	If .TRUE., limit the graupel production by rain riming of ice/snow by a bulk-density-based criterion on the mean-mass-particles of the collision partners.	iforcing=2,3 inwp_gscp=4,5,6,7
wgt_D_coll_limgrprod	R	0.5	-	Weight $\in [0, 1]$ for the collided mean-mass-particle's diameter D_{coll}: how much does the smaller collision partner contribute to the overall diameter?	llim_gr_prod_rain_riming=.TRUE. iforcing=2,3 inwp_gscp=4,5,6,7

Parameter	Type	Default	Unit	Description	Scope
wgt_rho_coll_limgrprod	R	0.5	-	Weight $\in [0, 1]$ for the limit of the collided mean-mass-particle's bulk density: how near should it be to the bulk density of graupel in order to convert it to graupel?	llim_gr_prod_rain_riming=.TRUE. iforcing=2,3 inwp_gscp=4,5,6,7

Defined and used in: `src/namelists/mo_2mom_mcrph_nml.f90`

Internally these namelist parameters are stored in the container `atm_phy_nwp_config(jg)%cfg_2mom` of type `t_cfg_2mom`.

The defaults are defined in the container `cfg_2mom_default` in `src/atm_phy_schemes/mo_2mom_mcrph_config_default.f90`

Adding new parameters can easily be done along the lines of one of the above existing parameters.

2.34. 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!	
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.	

Parameter	Type	Default	Unit	Description	Scope
filetype	I	4		One of CDI's FILETYPE_XXX constants, or FILETYPE_NONE to prevent writing a file but adding the diagnostics (e.g. pressure level remap) to the variable list. Possible values: 2=FILETYPE_GRB2, 4=FILETYPE_NC2, 5=FILETYPE_NC4, 999=FILETYPE_NONE	
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, "n" or "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.	itype_pres_msl < 3 (for technical reasons?)
include_last	L	.TRUE.		Flag whether to include the last time step	
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	

Parameter	Type	Default	Unit	Description	Scope
taxis_tunit	I	2		Time unit of the TAXIS_RELATIVE time axis. 1 = TUNIT_SECOND 2 = TUNIT_MINUTE 5 = TUNIT_HOUR 9 = TUNIT_DAY For a complete list of possible values see cdilib.c	mode=1
output_bounds	R($k*$ 3)	None		Post-processing times: start, end, increment. The increment (output interval) must be larger than the advection time step (`dtime`) and should be an integer multiple of it. Multiple triples are possible in order to define multiple starts/ends/intervals. See namelist parameters `output_start`, `output_end`, `output_interval` 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 `output_bounds` for an alternative specification of output events.	

Parameter	Type	Default	Unit	Description	Scope
output_end	C(:)	" "		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(:)	" "		ISO8601 time stamp for repeating output intervals. The output interval must be larger than the advection time step (`dtime`) and should be an integer multiple of it. 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 and `acc` for accumulated 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.	

Parameter	Type	Default	Unit	Description	Scope
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.	
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 *ready file* 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>`, `<datetime2>` 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]`.	

Parameter	Type	Default	Unit	Description	Scope
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.	
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.	
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 $(i-1) *$ `output_interval`. The output interval is then replaced by $N *$ `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`.	

Parameter	Type	Default	Unit	Description	Scope
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.	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 $start + k * increment <= end$, 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 underlaying 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 mixed 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:atmo_timemean`	time mean variables from `prog_timemean`,`tracer_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 *relative* day-hour-minute-second string
dddhhmmss	substituted by *relative* three-digit day-hour-minute-second string
hhhmmss	substituted by *relative* 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

2.35. parallel_nml

Parameter	Type	Default	Unit	Description	Scope
nproma	I	1		Loop chunk length. Only one of (*nproma*, *nblocks_c*, *nblocks_e*) may be specified in the namelist (>0) at any time.	
nblocks_c	I	0		Number of looping chunks used for cells. For values >0, *nproma* is recomputed according to the specified *nblocks_c*.	
nblocks_e	I	0		Number of looping chunks used for edges. For values >0, *nproma* is recomputed according to the specified *nblocks_e*.	
nproma_sub	I	nproma		Chunk size of subblocks used for example by ecRad or rrtmgp, which is needed for the GPU port to reduce the memory footprint. May only specify one of (*nproma_sub*, *nblocks_sub*) in the namelist (>0) at any time.	
nblocks_sub	I	1		Number of looping chunks used for subblocking. For values $<=0$ this is ignored. For bigger values, this overwrites *nproma_sub*. For reduced-grid radiation, we suggest explicitly specifying *nproma_sub* instead of using *nblocks_sub*.	
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 $=1$

Parameter	Type	Default	Unit	Description	Scope
p_test_run	L	.FALSE.		.TRUE. means verification run for MPI parallelization (PE 0 processes full domain)	
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.	p_test_run = .TRUE.
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 synchonization 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	
default_comm-_pattern_type	I	1		Default implementation of mo_communication to be used: 1 = original 2 = YAXT	
num_io_procs	I	0		Number of I/O processors (running exclusively for doing I/O)	

Parameter	Type	Default	Unit	Description	Scope
num_io_procs_radar	I	0		Number of dedicated I/O processors for the efficient radar forward operator EMVORADO. Choosing more I/O processors than the total number of simulated radar stations of all domains is not advisable, because one station is handled by one I/O processor. However, less I/O processors can be chosen, in which case one processor handles several stations. I/O tasks actually include much more than plain output for each station and can be very time consuming. More details can be found in the EMVORADO User's Guide available from the COSMO web page (www.cosmo-model.org → Documentation → EMVORADO) or from the emvorado submodule `./externals/emvorado/DOC/TEX/emvorado_userguide.pdf`. If `num_io_procs_radar`=0, a subset of the worker processors (=number of radar stations) are doing the I/O tasks, which may slow down the model considerably.	luse_radarfwo(<idom>) =.TRUE., iforcing=3
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).	Mandatory for itype_latbc = 1
proc0_shift	I	0		Number of processors at the beginning of the rank list that are excluded from the domain decomposition. Setting this parameter to 1 serves for offloading I/O to the vector hosts of the NEC Aurora, but it works technically on other platforms as well.	
use_omp_input	L	.FALSE.		Setting this parameter to .TRUE. activates OpenMP sections in initicon that allow task parallelism for reading atmospheric input data, overlapping reading, sending, and statistics calculations.	

Parameter	Type	Default	Unit	Description	Scope
pio_type	I	1		Type of parallel I/O. 1: Classical async I/O proccessors 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`

2.36. radiation_nml (relevant if run_nml:iforcing=3 (NWP))

Parameter	Type	Default	Unit	Description	Scope
isolrad	I	1		Insolation scheme 0: Use original insolation (from SRTM in case inwp_radiation=1 or from ecRad in case inwp_radiation=4) 1: Use SSI values from Coddington et al. (2016) (inwp_radiation=1) or scale SSI values to Coddington et al. (2016) values (inwp_radiation=4) 2: SSI from an external file containing monthly mean time series (inwp_radiation=4)	
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(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) 5: Zenith angel for radiative convective equilibrium test: perpetual equinox with 340 W/m2 6: Zenith angle with prescribed cosine of solar zenith angle (see parameter cos_zenith_fixed)	
cos_zenith_fixed	R	0.5		Cosine of zenith angle for test cases including SCM	izenith=6

Parameter	Type	Default	Unit	Description	Scope
islope_rad(max_dom)	I	0		Slope correction for surface radiation: 0: None 1: Slope correction for direct solar radiation without shading effects 2: Slope and horizon / sky-view factor correction for direct solar radiation including shading (Remark: sky-view correction not yet activated) 3: Slope correction for direct solar radiation including shading, but no further consideration of sky-view factor effects.	2 and 3 require the additional field HORIZON to be present in the extpar data
albedo_type	I	1		Type of surface albedo 1: based on soil type specific tabulated values (dry soil) 2: MODIS albedo 3: fixed albedo for SCM and other testcases	iforcing=inwp
albedo_fixed	R	0.5		Fixed albedo value for SCM and various testcases	iforcing=inwp albedo_type=3
direct_albedo	I	4		Direct beam surface albedo over land and sea-ice. Options mainly differ in terms of their solar zenith angle (SZA) dependency. 1: Ritter-Geleyn (1992) 2: Zängl (pers. comm.): For 'rough surfaces' over land direct albedo is not allowed to exceed the corresponding broadband diffuse albedo. Ritter-Geleyn for ice. 3: Yang et al (2008) for snow-free land points. Ritter-Geleyn for ice and Zängl for snow. 4: Briegleb and Ramanathan (1992) for snow-free land points. Ritter-Geleyn for ice and Zängl for snow.	iforcing=inwp albedo_type=2
direct_albedo_water	I	2		Direct beam surface albedo over water (ocean or lake). Options mainly differ in terms of their solar zenith angle (SZA) dependency. 1: Ritter-Geleyn (1992) 2: Yang (2008), originally designed for land 3: Taylor et al (1996) for direct and 0.06 for diffuse albedo as in the IFS.	iforcing=inwp albedo_type=2

Parameter	Type	Default	Unit	Description	Scope
albedo_whitecap	I	0		Ocean albedo increase by foam from breaking waves (whitecaps). Not applied over lakes. 0: off 1: whitecap describtion by Seferian et al 2018	iforcing=inwp albedo_type=2
icld_overlap	I	2		Method for cloud overlap calculation in shortwave part of RRTM 1: maximum-random overlap 2: generalized overlap (Hogan, Illingworth, 2000) 3: maximum overlap 4: random overlap 5: exponential overlap	iforcing=inwp inwp_radiation=1 (1-4) inwp_radiation=4 (1,2,5)

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 = 5: 3-dim concentration, time dependent, monthly means from yearly files bc_ozone_<year>.nc or - with nesting - bc_ozone_DOM<jg>_<year>.nc irad_o3 = 6: ozone climatology with T5 geographical distribution and Fourier series for seasonal cycle for run_nml/iforcing = 3 (NWP) irad_o3 = 7: GEMS ozone climatology (from IFS) for run_nml/iforcing = 3 (NWP) irad_o3 = 9: MACC ozone climatology (from IFS) for run_nml/iforcing = 3 (NWP) irad_o3 = 79: Blending between GEMS and MACC ozone climatologies (from IFS) for run_nml/iforcing = 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/iforcing = 3 (NWP) irad_o3 = 11: Ozone from SCM input file	

Parameter	Type	Default	Unit	Description	Scope
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
irad_aero	I	2		Aerosols 0: none 2: global constant 3: externally specified. This creates additional 4D-fields for: optical depth long wave (od_lw), optical depth short wave (od_sw), single scattering albedo short wave (ssa_sw) and asymmetry parameter short wave (g_sw). These fields need to be filled externally (e.g. by a ComIn plugin). 6: Tegen aerosol climatology for run_nml/iforcing = 3 (NWP) .AND. itopo =1 7: CAMS 3D aerosol climatology, the filename can be specified via cams_aero_filename in &radiation_nml 8: CAMS 3D forecasted aerosol, the filename can be specified via cams_aero_filename in &radiation_nml 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 12: tropospheric 'Kinne' aerosols, constant in time 13: total tropospheric 'Kinne' aerosols, time dependent from file 14: volcanic stratospheric aerosols for CMIP6, time dependent from file 15: tropospheric 'Kinne' aerosols + volcanic stratospheric aerosols for CMIP6, time dependent, both from file. If the 1850–file of the 'Kinne' aerosols is given, only the natural background from Kinne aerosol is applied. 18: tropospheric natural 'Kinne' aerosols from pre-industry (the 1850–file has to be linked for all simulations!) + time dep. volcanic stratospheric aerosols for CMIP6, both from file + param. time dep. anthropogenic 'simple plumes' 19: tropospheric natural 'Kinne' aerosols from pre-industry (the 1850–file has to be linked for all simulations!) + param. time dep. anthropogenic 'simple plumes'	

Parameter	Type	Default	Unit	Description	Scope
lrad_aero_diag	L	.FALSE.		writes actual aerosol optical properties to output	
ecrad_data_path	C	"."		Path to the folder containing ecRad optical properties files.	inwp_radiation=4 (ecRad)
cams_aero_filename	C	"CAMS_aero_ R<nroot0>B<jlev>_ DOM<idom>.nc"		Path to the file containing CAMS 3D data Climatology data can be prepared using the script scripts/preprocessing/ make_camsclim_onICONgrid.sh	inwp_radiation=4 (ecRad) irad_aero=7 or 8 (CAMS 3D climatology or forecasted aerosols)
ecrad_isolver	I	0		Radiation solver 0: McICA (Pincus et al. 2003) 1: Tripleclouds (Shonk and Hogan 2008) 2: McICA for OpenACC 3: SPARTACUS (Hogan et al. 2016)	inwp_radiation=4 (ecRad)
ecrad_igas_model	I	0		Gas model and spectral bands 0: RRTMG (Iacono et al. 2008) 1: ecckd (Hogan and Matricardi 2020)	inwp_radiation=4 (ecRad)
ecrad_llw_cloud_scat	L	.FALSE.		Long-wave cloud scattering.	inwp_radiation=4 (ecRad)
ecrad_use_general_cloud_optics	L	.FALSE.		General cloud optics in ecrad. It allows for different optical properties of ice and liquid.	inwp_radiation=4 (ecRad)
ecrad_iliquid_scat	I	0		Optical properties for liquid cloud scattering. Different options depending on ecrad_use_general_cloud_optics (eugco) 0: SOCRATES (eugco = .FALSE.) \Mie-droplet (eugco = .TRUE.) 1: Slingo (1989) (eugco = .FALSE.)	inwp_radiation=4 (ecRad)
ecrad_iice_scat	I	0		Optical properties for ice cloud scattering. Different options depending on ecrad_use_general_cloud_optics (eugco) 0: Fu et al. (1996) (eugco = .FALSE.) \Rough from Muskatel et al. (2021) (eugco = .TRUE.) 1: Baran et al. (2016) (eugco=.FALSE.) 2: Yi (2013) (eugco=.FALSE.) 10: Smooth from Muskatel et al. (2021) (eugco=.TRUE.) 11: Baum (eugco=.TRUE.)	inwp_radiation=4 (ecRad)
ecrad_isnow_scat	I	-1		Optical properties for snow scattering. -1: No explicit snow in radiation. 0: Rough from Muskatel et al. (2021) 10: Smooth from Muskatel et al. (2021)	inwp_radiation=4 (ecRad) ecrad_use_general_cloud_optics = .TRUE.

Parameter	Type	Default	Unit	Description	Scope
ecrad_irain_scat	I	-1		Optical properties for rain scattering. -1: No explicit rain in radiation. 0: Mie-rain	inwp_radiation=4 (ecRad) ecrad_use_general_cloud_optics = .TRUE.
ecrad_igraupel_scat	I	-1		Optical properties for graupel scattering. -1: No explicit graupel in radiation. 0: Rough from Muskatel et al. (2021) 10: Smooth from Muskatel et al. (2021)	inwp_radiation=4 (ecRad) ecrad_use_general_cloud_optics = .TRUE.
decorr_pole	R	2000	m	Decorrelation length scale at poles	inwp_radiation=4 (ecRad)
decorr_equator	R	2000	m	Decorrelation length scale at equator	inwp_radiation=4 (ecRad)
ecrad_check_input	L	.FALSE.		Debug mode for ecRad input: - Checks several input fields for physical consistency - Increases the verbosity of ecRad (setup and runtime)	inwp_radiation=4 (ecRad)

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

2.37. 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.	
dtime	R	600.0	s	model time step For real case runs the maximum allowable time step can be estimated as $1.8 \cdot \text{ndyn\_substeps} \cdot \overline{\Delta x}\,\text{s}\,\text{km}^{-1}$, 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.	

Parameter	Type	Default	Unit	Description	Scope
modelTimeStep	C	”	ISO8601 format-ted string	model time step (**should be preferred over the concurrent namelist parameter dtime**) For real case runs the maximum allowable time step can be estimated as $1.8 \cdot \text{ndyn\_substeps} \cdot \overline{\Delta x}\, \mathrm{s\,km^{-1}}$, 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: AES 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)	

Parameter	Type	Default	Unit	Description	Scope
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.	
debug_check_level	I	0		Setting a value larger than 0 activates debug checks.	
output	C(:)	"nml", "totint"		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: • "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". If the output namelist parameter is not set explicitly, the default setting "nml","totint" is assumed.	
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	

Parameter	Type	Default	Unit	Description	Scope
check_uuid_gracefully	L	.FALSE.		If this flag is set to `.TRUE.` we give only warnings for non-matching UUIDs.	
luse_radarfwo	L(max_dom)	.FALSE.		For each domain, switch to activate the efficient volume scan radar forward operator EMVORADO. The EMVORADO code is provided as a submodule named `emvorado`, which is part of the ICON distribution. ICON itself contains only some ICON specific interface modules. `./configure` (respectively the call to a configure wrapper script) needs the option `--enable-emvorado`. EMVORADO needs its own namelist(s) for each radar-active model domain in a separate namelist input file `RADARSIM_PARAMS`. More details can be found in the EMVORADO User's Guide available from the COSMO web page (www.cosmo-model.org → Documentation → EMVORADO) or from the submodule `./externals/emvorado/DOC/TEX/emvorado_userguide.pdf`.	iforcing=3, ICON configure'd with `--enable-emvorado`
radarnmlfile	C			The name of the file containing the EMVORADO namelist. If this is empty or not set, the Default from radar_data_namelist.f90 is used. Only used if luse_radarfwo is .TRUE. .	

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

2.38. scm_nml (relevant if l_scm_mode)

Parameter	Type	Default	Unit	Description	Scope
i_scm_netcdf	I	1		reading SCM input data from 0: ASCII file 1: normal ICON netcdf file 2: DEPHY unified netcdf file	
lscm_icon_ini	L	.FALSE.		read initial conditions produced by ICON on the native grid	

Parameter	Type	Default	Unit	Description	Scope
lscm_random_noise	L	.FALSE.		initialize with random noise - for LEM runs by ICON on the native grid	
lscm_read_tke	L	.FALSE.		read init. tke from netcdf	
lscm_read_z0	L	.FALSE.		read z0 from netcdf	
scm_sfc_mom	I	0		prescribed surface boundary condition for momentum using 0: TERRA 2: friction velocity 4: drag coefficient 5: Louis surface layer scheme	
scm_sfc_qv	I	0		prescribed surface boundary condition for moisture using 0: TERRA 1: surface moisture (qv_s) 2: latent heat flux 3: saturation 4: draf coefficient 5: Louis surface layer scheme	
scm_sfc_temp	I	0		prescribed surface boundary condition for temperature using 0: TERRA 1: surface temperature (t_g) 2: sensible heat flux (shfl_s) 4: drag coefficient 5: Louis surface layer scheme	

Defined and used in: `src/namelists/mo_scm_nml.f90`

2.39. 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	

Parameter	Type	Default	Unit	Description	Scope
max_lay_thckn	R	25000	m	Maximum layer thickness below the height given by htop_thcknlimit (NWP recommendation: 400 m) *Use with caution! Too ambitious settings may result in numerically unstable layer configurations.*	
htop_thcknlimit	R	15000	m	Height below which the layer thickness does not exceed max_lay_thckn	
nshift_above_thcklay	I	0		Level shift above constant-thickness layer for further calculation of layer distribution. For strongly stretched grids with a deep constant-thickness layer, this parameter may be set to 1 in order to reduce the thickness jump right above the constant-thickness layer.	
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 3: second-order polynomial (see M. Baldauf COSMO-TR p. 33)	
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/namelists/mo_sleve_nml.f90`

2.40. sppt_nml

The Stochastic Perturbation of Physical Tendencies (SPPT) method is controlled by the following set of Namelist parameters. Note that SPPT is only available for the NWP physics package (`iforcing=3`)). In addition, SPPT is not supported on a global domain (hard exit) and is untested in limited area mode where the domain extends across the poles. Running the latter is currently not recommended.

Parameter	Type	Default	Unit	Description	Scope
lsppt	L	.FALSE.		TRUE: forecast with SPPT	
hinc_rn	R	21600	second	time increment for drawing a new field of random numbers	
dlat_rn	R	0.1	deg	random number coarse grid point distance in meridional direction	
dlon_rn	R	0.1	deg	random number coarse grid point distance in zonal direction	
range_rn	R	0.8		max magnitude of random numbers	
stdv_rn	R	1.0		standard deviation of the gaussian distribution of random numbers	

Defined and used in: `src/namelists/mo_sppt_nml.f90`

2.41. synsat_nml[1]

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.

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

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

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/namelists/mo_synsat_nml.f90`

2.42. synradar_nml

The list of diagnostic output variables in ICON incorporates some fields related to synthetic radar reflectivity on the model grid:

- `'dbz', 'dbz_850', 'dbz_cmax', 'dbz_ctmax'`
- `'echotop', 'echotopinm'`

By default, these are based on a simple analytic so-called Rayleigh-approximation for single-particle backscattering.
If ICON is configured with the flag `--enable-emvorado` and compiled with the pre-processor flag `-DHAVE_RADARFWO`, some alternative, more accurate Mie- or T-matrix methods from the radar forward operator EMVORADO can be used by namelist choice (see below), particularly for improving the simulation of the so-called "bright band", the enhanced reflectivity in the melting layer.
EMVORADO is the Efficient Modular VOlume RADar Operator for simulating radar volume scans for cloud- and weather radar wavelengths, see

- EMVORADO User's Guide
  in ICON's EMVORADO submodule `./externals/emvorado/DOC/TEX/emvorado_userguide.pdf`
  or on the COSMO web page (www.cosmo-model.org → Documentation → EMVORADO)
  `http://www.cosmo-model.org/content/model/documentation/core/emvorado_userguide.pdf`
- A COSMO Technical Report No. 28
  on the COSMO web page (www.cosmo-model.org → COSMO Tech Reports)
  `http://www.cosmo-model.org/content/model/documentation/techReports/cosmo/docs/techReport28.pdf`

for detailed information.

Parameter	Type	Default	Unit	Description	Scope
synradar_meta This type contains: synradar_meta%itype_refl and many other parameters which are only relevant if itype_refl is not the default (4)	TYPE(dbzcalc_params) I	 4		Instance of the derived type **dbzcalc_params** from EMVORADO to specify details of the radar reflectivity calculation for related outputs ('dbz', 'dbz_850', 'dbz_cmax', 'dbz_ctmax', 'echotop', 'echotopinm'). The type is documented in detail in the EMVORADO User's Guide. The most important component is **itype_refl:** **1:** Mie-scattering from EMVORADO assuming spherical particles and including a detailed melting scheme for the radar "bright band". **3:** Rayleigh-Oguchi approximation from EMVORADO including a simple melting scheme, but not producing pronounced "bright bands". **4:** Traditional Rayleigh approximation from ICON, also without pronounced "bright bands". **This is the default.** **5:** T-matrix scattering from EMVORADO assuming oblate spheroids, otherwise similar to Mie-option 1. **6:** T-matrix scattering from EMVORADO assuming spherical particles, only for sanity-checks against Mie-option 1. For options 1, 5, 6 there are many more relevant type components.	iforcing=3, ICON configure'd with `--enable-emvorado`
ydir_mielookup_write	C	' '		For reflectivity calculations: directory for storing new automatically created reflectivity lookup tables in case of EMVORADO-methods that employ reflectivity lookup tables to boost efficiency (synradar_meta%itype_refl=1, 5, 6 together with synradar_meta%llookup_mie=.TRUE.)	iforcing=3, ICON configure'd with `--enable-emvorado`, synradar_meta%itype_refl=1, 5, 6 synradar_meta%llookup_mie=.TRUE.

Parameter	Type	Default	Unit	Description	Scope
ydir_mielookup_read	C	' '		For reflectivity calculations: directory for reading the reflectivity lookup tables in case of EMVORADO-methods that employ reflectivity lookup tables to boost efficiency (synradar_meta%itype_refl=1, 5, 6 together with synradar_meta%llookup_mie=.TRUE.)	iforcing=3, ICON configure'd with `--enable-emvorado`, synradar_meta%itype_refl=1, 5, 6 synradar_meta%llookup_mie=.TRUE.

Defined and used in: `src/namelists/mo_synradar_nml.f90`

2.43. 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	0.	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 `io_nml:dt_checkpoint`. Only if the value of `dt_checkpoint` resulting from model default or user's specification is longer than `dt_restart`, it will be reset (by the model) to `dt_restart` so that at least one restart file is generated during the restart cycle. If `dt_restart` is larger than but not a multiple of `dt_checkpoint`, restart file will *not* be generated at the end of the restart cycle.	
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	

Parameter	Type	Default	Unit	Description	Scope
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".

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

Parameter	Type	Default	Unit	Description	Scope
lvadv_tracer	L	.TRUE.		Main switch for vertical tracer transport. TRUE/FALSE : compute/do not compute vertical tracer advection. If vertical advection is switched off, the tracer mass fraction q is kept constant.	
ihadv_tracer	I(ntracer)	2		Tracer specific method to compute horizontal advection: 0: no horiz. transport. The tracer mass fraction q is kept constant. 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 $z = 22\,\mathrm{km}$ (see nonhydrostatic_nml/hbot_qvsubstep).	 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. The tracer mass fraction q is kept constant. 1: upwind (1st order) 2: Parabolic Spline Method (PSM): allows for CFL > 1	lvadv_tracer=TRUE

Parameter	Type	Default	Unit	Description	Scope
				3: Piecewise parabolic method (PPM): allows for CFL > 1	
itype_hlimit	I(ntracer)	4		Type of limiter for horizontal transport: 0: no limiter 3: monotonic flux limiter (FCT) 4: positive definite flux limiter	
itype_vlimit	I(ntracer)	1		Type of limiter for vertical transport: 0: no limiter 1: semi-monotonic reconstruction filter 2: monotonic reconstruction filter 3: positive definite flux limiter	
ivlimit_selective	I(ntracer)	0		Reduce detrimental effect of vertical limiter by applying a method for identifying and avoiding spurious limiting of smooth extrema. 1: on 0: off	itype_vlimit=1, 2
nadv_substeps	I(max_dom)	3		Tracer substepping: Number of time integration substeps per fast physics/advective time step dtime. If only one value is specified, it is copied to all child domains, implying that the same value 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.	only active for the schemes ihadv_tracer=20, 22, 32, 42, 52. Starts at minimum height height hbot_qv_substep for the schemes 22, 32, 42, 52, whereas it is applied throughout the entire domain for scheme 20.
beta_fct	R	1.005		global boost factor for range of permissible values $[q_{max}, q_{min}]$ in the 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_hlimit = 3
iadv_tke	I	0		Type of TKE advection 0: no TKE advection 1: vertical advection only 2: vertical and horizontal advection	inwp_turb=1

Parameter	Type	Default	Unit	Description	Scope
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 AES physics packages are switched on.	iforcing≠ inwp, iaes'
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.	
init_formula	C	' '		Comma-separated list of initialization formulas for additional passive tracers.	npassive_tracer > 0
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 3: based on shape function derivatives for a three-node triangular element (Fish. J and T. Belytschko, 2007)	ihadv_tracer=2, 20
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/namelists/mo_advection_nml.f90`

2.45. turb_vdiff_nml

The parameterization of vertical diffusion (VDIFF) module is configured by a a set of parameters, each of which is a 1-dimensional array extending over all domains. The parameters provide control over some of the parametrized effects (only active when nwp_phy_nml%inwp_turb = 6):

Parameter	Type	Default	Unit	Description	Scope
General					
lsfc_mom_flux	L	.TRUE.		switch on/off surface momentum flux	
lsfc_heat_flux	L	.TRUE.		switch on/off surface heat flux	
turb	S	'tte'		'tte': TTE scheme '3dsmag': 3D Smagorinsky scheme	
z0m_min	R	1.5×10^{-5}	m	Minimum roughness length for momentum	
z0m_ice	R	0.001	m	Roughness length for momentum over ice	
z0m_oce	R	0.001	m	Roughness length for momentum over ocean	
fsl	R	0.4		fraction of first-level height at which surface fluxes are nominally evaluated, tuning param for sfc stress	
TTE Scheme					
pr0	R	1.0		neutral limit Prandtl number, can be varied from about 0.6 to 1.0, fixes f_theta0	
f_tau0	R	0.17		neutral non-dimensional stress factor (0.1 - 0.22)	
f_tau_limit_fraction	R	0.25		Fraction of f_tau0 for large Ri numbers (0 - 0.6)	
f_theta_limit_fraction	R	0.		Fraction of f_theta0 for large Ri numbers (0 - 0.3)	
f_tau_decay	R	4.		Decay constant of f_tau0 for large Ri numbers (0.5 - 5)	
f_theta_decay	R	4.		Decay constant of f_theta0 for large Ri numbers (1 - 10)	
ek_ep_ratio_stable	R	3.		Ratio of TKE to TPE for large positive Ri (Mauritsen: $1/(0.3 \pm 1) - 1$)	
ek_ep_ratio_unstable	R	2.		Ratio of TKE to TPE for large negative Ri (Mauritsen: 1)	
c_f	R	0.185		mixing length: coriolis term tuning parameter	
c_n	R	2.0		mixing length: stability term tuning parameter	
wmc	R	0.5		ratio of typical horizontal velocity to wstar at free convection	
fbl	R	3.0		1/fbl: fraction of BL height at which lmix hat its max	
lmix_max	R	150	m	maximum mixing length	
3D Smagorinsky Scheme					
km_min	R	0.001	Pa s	minimum mass weighted turbulent viscosity	
turb_prandtl	R	1/3		Turbulent Prandtl number	

Parameter	Type	Default	Unit	Description	Scope
min_sfc_wind	R	1.	m/s	minimum surface wind speed in free-convection limit	

The limit fractions L and decay constants D for f_τ and f_θ are defined with respect to the ansatz

$$f_\tau(\mathrm{Ri}) = f_\tau(0)\left(L + \frac{1-L}{1+D\,\mathrm{Ri}}\right).$$

Defined and used in: `src/namelists/mo_turb_vdiff_nml.f90`

2.46. 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 *imode_turb* 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 *icldm_turb* but only for the transfer layer.	
itype_wcld	I	2		Type of water cloud diagnosis within the turbulence scheme: 1: employing a scheme based on relative humitidy 2: employing a statistical saturation adjustment.	icldm_turb=2 or icldm_tran=2
q_crit	R	1.6		Critical value for normalized super-saturation.	itype_wcld=2

Parameter	Type	Default	Unit	Description	Scope
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.	
ltkeshs	L	.TRUE.		Consider TKE-production by separated horizontal shear eddies.	
imode_shshear	I	2		Mode of calculat. the separated horizontal shear mode: 0: with a constant length scale and based on 3D-shear and incompressibility 1: with a constant length scale and considering the trace constraint for the 2D-strain tensor 2: with a Ri-number depend. length-scale correct. and the trace constraint for the 2D-strain tensor.	ltkeshs=.TRUE. and a_hshr>0
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: with Ri-number dependent reduction factor for Ri>1 3: as "2", but with additional reduction for dx < 2 km.	ltkesso=.TRUE.
ltkecon	L	.FALSE.		Consider TKE-production by sub grid convective plumes.	inwp_conv = 1
ltmpcor	L	.FALSE.		Consider thermal TKE sources in enthalpy equation.	
lcpfluc	L	.FALSE.		Consideration of fluctuations of the heat capacity of air.	
tur_len	R	500.0	m	Asymptotic maximal turbulent distance (where $\kappa\cdot$ *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 master length-scale. In case of *a_stab*=0, this turbulent length scale is not reduced for stable stratification.	
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.	ltkeshs=.TRUE.
tkhmin	R	0.75	m^2/s	Basic minimum vertical diffusion coefficient for scalar properties like heat and moisture (being corrected by an empirical factor proportional to $Ri^{-2/3}$).	
tkhmin_strat	R	0.75	m^2/s	Enhanced value of *tkhmin* valid for the stratosphere above 17.5 km (tropics above 22.5 km) (being corrected by an empirical factor proportional to $Ri^{-1/3}$).	
tkmmin	R	0.75	m^2/s	Basic minimum vertical diffusion coefficient for momentum (being corrected by an empirical factor proportional to $Ri^{-2/3}$).	
tkmmin_strat	R	4	m^2/s	Enhanced value of *tkmmin* valid for the stratosphere above 17.5 km (tropics above 22.5 km) (being corrected by an empirical factor proportional to $Ri^{-1/3}$).	
imode_tkemini	I	1		Mode of adapting q=SQRT(2*TKE) and the entire Turbulence-Model (TMod) to Lower Limits for Diff. Coeffs. (LLDCs), which are the "minimal vertical diffusion coefficients" *tkhmin* or *tkmmin* (or their stratopheric extensions): 1: LLDCs treated as corrections of stability lengths without any further adaptation. 2: TKE adapted to that part of LLDCs representing so far missing shear forcing, while the assumed part of LLDCs, representing missing drag-forces, has no feedback to the TMod. 3: Tuned variant of "2" that is suitable for operational forecasts, particularly excluding the effect of stratospheric extensions on the TKE adaptation.	any (extended) LLDC is active
alpha0	R	0.0123	1	Standard Charnock parameter.	

Parameter	Type	Default	Unit	Description	Scope
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.	
imode_charpar	I	2		Options for specifying the Charnock parameter: 1: constant at *alpha0* 2: wind-speed dependent with maximum at *alpha0_max* 3: as "2", but decreasing again at speeds above about 25 m/s in order to improve pressure-speed relationship in tropical cyclones.	
lconst_z0	L	.FALSE.		TRUE: horizontally homogeneous roughness length z0.	
const_z0	R	0.001	m	Value for horizontally homogeneous roughness length z0.	lconst_z0=.TRUE.
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	10.0	1	Scaling factor of the laminar boundary layer for scalars (heat and vapor). The larger rlam_heat, the larger is the laminar resistance.	
rat_lam	R	1.0	1	Vapour/Heat ratio of laminar scaling factors (over land). The larger *rat_lam*, the larger is the laminar resistance for evaporation compared to sensible heat.	
rat_sea	R	0.8	1	Sea/Land ratio of laminar scaling factors for scalars (heat and vapor). The larger *rat_sea*, the larger is the laminar resistance for a sea surface compared to a land surface.	

Parameter	Type	Default	Unit	Description	Scope
rat_glac	R	3.0	1	Glacier/Land ratio of laminar scaling factors for scalars (heat and vapor). The larger *rat_glac*, the larger is the laminar resistance over glaciers compared to other land surfaces.	
tkesmot	R	0.15	1	Time smoothing factor within $[0, 1]$ for TKE. In case of *tkesmot*=0, no smoothing is active.	
frcsmot	R	0.0	1	Vertical smoothing factor within $[0, 1]$ for TKE forcing terms. In case of *frcsmot*=0, no smoothing is active.	
imode_frcsmot	I	1		1: Apply vertical smoothing 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).	frcsmot>0
imode_snowsmot	I	1		Mode to treating the aerodynamic surface-smoothing by snow: 0: no smoothing active at all 1: no impact on SAI, but full smoothing of land-use R-length 2: "1", but with full smoothing of SAI: full smoothing of R-length and SAI 3: dynamical smoothing of R-length and SAI dependent on snow- and R-height.	itype_z0 $\geq$ 2
lsflcnd	L	.TRUE.		Use lower flux condition for vertical diffusion calculation (TRUE) instead of a lower concentration condition (FALSE).	
lfreeslip	L	.FALSE.		Use a free-slip lower boundary condition (TRUE), i.e. neither momentum nor heat/moisture fluxes (use for idealized runs only!).	
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).	
ldiff_qi	L	.FALSE.		Turbulent diffusion of cloud ice (TRUE).	

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

2.47. upatmo_nml

Parameter	Type	Default	Unit	Description	Scope
Extrapolation to determine the inital state of the upper atmosphere					itype_vert_expol = 2
expol_start_height	R	70000	m	Height above which extrapolation of initial data starts.	
expol_blending_scale	R	10000	m	Vertical distance above expol_start_height within which blending of linearly extrapolated state and climatological state takes place.	
expol_vn_decay_scale	R	10000	m	Scale height of vertically exponentially decaying factor multiplied to the extrapolated horizontal wind (to alleviate stability-endangering wind magnitudes).	
expol_temp_infty	R	400	K	Exospheric mean reference temperature of the climatology for the extrapolation blending.	
lexpol_sanitycheck	L	.FALSE.		.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.)	
Upper-atmosphere physics					(iforcing = 2 (AES) & "coming soon") or (iforcing = 3 (NWP) & lupatmo_phy = .TRUE.)
orbit_type	I	1		Orbit model for upper-atmosphere radiation (compare aes_rad_nml: l_orbvsop87): 1: vsop87 → standard and accurate model 2: kepler → simple model appropriate for idealized work	

Parameter	Type	Default	Unit	Description	Scope
solvar_type	I	1		Solar activity: 1: normal 2: low 3: high	
solvar_data	I	2		Data set for solar activity: 1: G. Rottman data 2: J. Lean data	
solcyc_type	I	2		Solar cycle: 1: standard cycle 2: 27-day cycle	
nwp_grp_<groupname>%...				Configuration of the upper-atmosphere process groups under NWP-forcing (compare time control of processes in aes_phy_nml): <groupname> = imf: ion drag, molecular diffusion and frictional heating <groupname> = rad: radiation and chemical heating	iforcing = 3 lupatmo_phy = .TRUE.

Parameter	Type	Default	Unit	Description	Scope
...imode	I(max_ dom)	1		Group mode: 0: all processes clustered in the group <groupname> are switched off 1: all processes are switched on 2: all processes run in offline-mode, i.e. tendencies are computed, but not coupled to the dynamics Example of usage for multi-domain applications: • set nwp_grp_imf%imode = 1 to switch on the IMF-group for all domains (default) • set nwp_grp_rad%imode = 1,1,0 to switch on the RAD-group for domain 1 and 2, but to switch it off for domain 3 Please note: if imode = 1 or 2 for a domain, but lupatmo_phy = .FALSE. for this domain, imode is set to 0 and the group is switched off.	
...dt	R(max_ dom)	$300.0\|_{\mathrm{imf}}$, $600.0\|_{\mathrm{rad}}$	s	Tendency update period. New tendencies from all processes of a group are computed every dt (temperature, wind and water vapor tendencies in case of IMF, and temperature tendencies in case of RAD). Please note: internal processing will round dt to the next multiple of the domain-adjusted value of run_nml: dtime, which in turn might have been rescaled, if grid_nml: grid_rescale_factor $\neq 1$. In case of a domain-wise assignment in a multi-domain application, $dt(1) \geq dt(2) \geq \ldots$ is required.	

Parameter	Type	Default	Unit	Description	Scope
...t_start ...t_end	C	" "		Tendencies from all processes of a group are computed within the time interval [t_start, t_end]. Outside this interval the tendencies are set to zero. Format as for time_nml: ini_datetime_string, e.g. nwp_grp_imf%t_start = "2008-09-01T00:00:00Z". Empty strings will be replaced by the simulation start and/or end date and time of the domain. t_start and t_end apply to all domains, no domain-wise specification possible!	
...start_height	R	-999.0	m	All processes of a group compute tendencies above start_height. Below start_height the processes are inactive and all tendencies are set to zero. A negative value means that the default start heights of each process, listed in src/upper_atmosphere/mo_upatmo_impl_const: startHeightDef, are applied. Please note: start_height applies to all domains. If it is above the top of one domain, the group is switched off for that domain (imode(idom) is set to 0).	
nwp_gas_<gasname>%...				Configuration of the radiatively active gases in the upper atmosphere under NWP-forcing (compare radiation_nml and aes_rad_nml): <gasname> = o3: ozone (O_3) <gasname> = o2: dioxygen (O_2) <gasname> = o: atomic oxygen (O) <gasname> = co2: carbon dioxide (CO_2) <gasname> = no: nitric oxide (NO) (Dinitrogen (N_2) is determined diagnostically.)	iforcing = 3 lupatmo_phy = .TRUE. nwp_grp_rad%imode > 0

Parameter	Type	Default	Unit	Description	Scope
...imode	I	2		Gas mode (comparable, but generally not identical to the irad_<gasname> in radiation_nml and aes_rad_nml). 0: zero gas concentration 1: constant gas concentration (independent of space and time), specified via nwp_gas_<gasname>%vmr 2: external data; meridionally, vertically and monthly varying gas concentrations are read from a file with name nwp_extdat_gases%filename	
...vmr	R	0.0	m^3/m^3	Constant volume mixing ratio for a radiatively active gas.	nwp_gas_<gasname>%imode = 1
...fscale	R	1.0		Scaling factor the gas concentration in each grid cell is multiplied with.	nwp_gas_<gasname>%imode > 0
nwp_extdat_<extdatname>%...				Configuration of the external upper-atmosphere data: <extdatname> = gases: concentrations of the radiatively active gases <extdatname> = chemheat: temperature tendencies from chemical heating Please note: the standard NWP physics use other external gas data (e.g., for ozone)!	nwp_grp_rad%imode > 0
...dt	R	86400.0	s	Update period for the time interpolation of the external data. Currently, the external data provide monthly mean values. In order to avoid too strong jumps in the transition from one month to the next, the parameters are "smoothed" in time by a linear interpolation that is computed every dt. A value of the order of a day should be entirely sufficient for this purpose.	

Parameter	Type	Default	Unit	Description	Scope
...filename	C	"upatmo_gases_chemheat.nc"		Name of the file containing the external data. The file of the default name can be found in the folder data/, to which a link has to be set in the run script, following the typical examples of nwp_phy_nml: lrtm_filename and cldopt_filename. May contain the keyword <path> which will be substituted by model_base_dir (e.g., "<path>upatmo_gases_chemheat.nc"). Please note: if you would like to use other external data files, their data structure has to follow *exactly* the data structure of data/upatmo_gases_chemheat.nc (variable and dimension names and units, zonally averaged monthly mean gas concentrations on pressure levels, zonally averaged monthly mean temperature tendencies from chemical heating on geometric height levels etc.). Any other structure cannot be processed for the time being!	

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

Some notes on the output of upper-atmosphere-specific variables (under NWP-forcing):
An output of upper-atmosphere fields is only possible, if upper-atmosphere physics are switched on.
Upper-atmosphere fields cannot be output in the GRIB format (output_nml: filetype = 2).
Upper-atmosphere fields entered on output_nml: m/h/pl_varlist need the prefix "upatmo_".

The following fields can be output, if ...

...lupatmo_phy = .TRUE.:

`upatmo_mdry`	Mass of dry air
`upatmo_amd`	Molar mass of dry air
`upatmo_cpair`	Heat capacity of (moist) air at constant pressure
`upatmo_grav`	Gravitational acceleration of Earth

...lupatmo_phy = .TRUE. & nwp_grp_rad%imode > 0:

`upatmo_sclrlw`	Scaling factor for standard long-wave radiation heating rate from radiative processes

	out of local thermodynamic equilibrium
`upatmo_effrsw`	Efficiency factor for standard short-wave radiation heating rate from chemical heating
`upatmo_o3`	Mass mixing ratio of ozone (member of `group:upatmo_rad_gases`)
`upatmo_o2`	Mass mixing ratio of dioxygen (member of `group:upatmo_rad_gases`)
`upatmo_o`	Mass mixing ratio of atomic oxygen (member of `group:upatmo_rad_gases`)
`upatmo_co2`	Mass mixing ratio of carbon dioxide (member of `group:upatmo_rad_gases`)
`upatmo_no`	Mass mixing ratio of nitric oxide (member of `group:upatmo_rad_gases`)
`upatmo_n2`	Mass mixing ratio of dinitrogen (member of `group:upatmo_rad_gases`)
`upatmo_ddt_temp_srbc`	Temperature tendency due to absorbtion by O2 in Schumann-Runge band and continuum (member of `group:upatmo_tendencies`)
`upatmo_ddt_temp_nlte`	Temperature tendency due to radiative processes out of local thermodynamic equilibrium (member of `group:upatmo_tendencies`)
`upatmo_ddt_temp_euv`	Temperature tendency due to heating from extreme ultraviolet radiation (member of `group:upatmo_tendencies`)
`upatmo_ddt_temp_no`	Temperature tendency due to NO heating at near infrared (member of `group:upatmo_tendencies`)
`upatmo_ddt_temp_chemheat`	Temperature tendency due to chemical heating (member of `group:upatmo_tendencies`)

…lupatmo_phy = .TRUE. & nwp_grp_imf%imode > 0:

`upatmo_ddt_temp_vdfmol`	Temperature tendency due to molecular diffusion (member of `group:upatmo_tendencies`)
`upatmo_ddt_temp_fric`	Temperature tendency due to frictional heating (member of `group:upatmo_tendencies`)
`upatmo_ddt_temp_joule`	Temperature tendency due to Joule heating from ion drag (member of `group:upatmo_tendencies`)
`upatmo_ddt_u_vdfmol`	Zonal component of wind tendency due to molecular diffusion (member of `group:upatmo_tendencies`)
`upatmo_ddt_v_vdfmol`	Meridionl component of wind tendency due to molecular diffusion (member of `group:upatmo_tendencies`)
`upatmo_ddt_u_iondrag`	Zonal component of wind tendency due to ion drag (member of `group:upatmo_tendencies`)
`upatmo_ddt_v_iondrag`	Meridionl component of wind tendency due to ion drag (member of `group:upatmo_tendencies`)
`upatmo_ddt_qv_vdfmol`	Tendency of specific humidity due to molecular diffusion (member of `group:upatmo_tendencies`)

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/iforcing=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)	
i_ice_albedo	I	1		Switch for albedo model. Only one is implemented so far.	
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	Defaults to 1 when i_ice_dyn=0 and 2 otherwise.
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. Ocean waves specific namelist parameters

The following Namelists become active, if ICON is configured with --enable_waves, and if model_type=98 is selected in the Namelist master_model_nml.

4.1. energy_propagation_nml (used if run_nml/ltransport=.TRUE.)

Parameter	Type	Default	Unit	Description	Scope
itype_limit	I	0		Type of limiter for wave energy transport: 0: no limiter 3: monotonic flux limiter (FCT) 4: positive definite flux limiter	
beta_fct	R	1.005		global boost factor for range of permissible values $[q_{max}, q_{min}]$ in the 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_limit = 3
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 3: based on shape function derivatives for a three-node triangular element (Fish. J and T. Belytschko, 2007)	
lgrid_refr	L	.TRUE.		.TRUE.: calculate grid refraction	

Defined and used in: src/waves/config/mo_energy_propagation_nml.f90

4.2. initwave_nml

Parameter	Type	Default	Unit	Description	Scope
dt_shift	R	0.	s	time interval by which the actual model start date (tc_start_date) is shifted backwards in time.	

Defined and used in: src/waves/config/mo_initwave_nml.f90

4.3. wave_nml

Parameter	Type	Default	Unit	Description	Scope
ndirs	I	24		number of direction of wave spectrum	
nfreqs	I	25		number of frequencies of wave spectrum	
fr1	R	0.04177248	Hz	first frequency of wave spectrum	
co	R	1.1		frequency ratio	
iref	I	1		frequency bin number of reference frequency	
alpha	R	0.018		Phillips parameter	
fm	R	0.2	Hz	peak frequency and/or maximum frequency	
gamma_wave	R	3.0		overshoot factor	
sigma_a	R	0.07		left peak width of wave spectrum	
sigma_b	R	0.09		right peak width of wave spectrum	
fetch	R	300000.	m	fetch	
fetch_min_energy	R	25000.	m	fetch used for calculation of minimum allowed energy level	
roair	R	1.225	kg/m3	air density	
rnuair	R	1.5e-5	m2/s	kinematic air viscosity	
rnuairm	R	0.11*rnuair	m2/s	kinematic air viscosity for momentum transfer	
rowater	R	1000.	kg/m3	water density	
xeps	R	roair/rowater		air water density ratio	
xinveps	R	1./xeps		inverse air water density ratio	
betamax	R	1.20		parameter for wind input (ECMWF cy45r1)	
zalp	R	0.0080		shifts growth curve (ECMWF cy45r1)	
jtot_tauhf	I	19		dimension of high freuency wave stress (wtauhf)	must be odd
alpha_ch	R	0.0075		minimum Charnock constant (ecmwf cy45r1)	
depth	R	0.	m	ocean depth if not 0, then constant depth	
depth_min	R	0.2	m	allowed minimum of model depth	
depth_max	R	999.0	m	allowed maximum of model depth	
niter_smooth	I	1		number of smoothing iterations for wave bathymetry	
nsubs_refrac	I	1		number of substeps in wave refraction calculation (recommendation: use the same ratio as between atmosphere and wave model)	
xkappa	R	0.40		von Karman constant	
xnlev	R	10.0	m	windspeed reference level	
linput_sf1	L	.TRUE.		.TRUE.: calculate wind input source function term	

Parameter	Type	Default	Unit	Description	Scope
linput_sf2	L	.TRUE.		.TRUE.: update wind input source function term	
ldissip_sf	L	.TRUE.		.TRUE.: calculate dissipation source function term	
lwave_brk_sf	L	.TRUE.		.TRUE.: calculate depth-induced wave breaking dissipation source function term	
lnon_linear_sf	L	.TRUE.		.TRUE.: calculate non linear source function term	
lbottom_fric_sf	L	.TRUE.		.TRUE.: calculate bottom friction source function term	
lwave_stress1	L	.TRUE.		.TRUE.: calculate wave stress	
lwave_stress2	L	.TRUE.		.TRUE.: update wave stress	
impl_fac	R	1.0		Implicitness factor for time integration scheme of total source function Range of permissible values: $[0.5, \dots, 1]$ 0.5: second order Crank-Nicholson scheme 1.0: first order Euler backward scheme	
forc_file_prefix	C			common prefix of forcing files if not empty, the names of forcing files will be consctructed as: forc_file_prefix+_wind.nc - for 10m wind forc_file_prefix+_ice.nc - for sea ice concentration forc_file_prefix+_slh.nc - for sea level height forc_file_prefix+_osc.nc - for ocean surface currents Data for all time steps in the current simulation should be prepared in a single file. Variables should be named `u_10m`, `v_10m`, `fr_seaice`, `uosc`, `vosc`	coupled_mode= .FALSE. in coupling_mode_nml coupled_mode= .FALSE. in coupling_mode_nml
peak_u10	R	9.0	m/s	peak value of 10 m U wind component for test case	
peak_v10	R	9.0	m/s	peak value of 10 m V wind component for test case	
peak_lat	R	-60.0	degree	latitude of wind peak value	
peak_lon	R	-140.0	degree	longitude of wind peak value	
impl_fac	R	1.0		first order Euler backward time integration scheme for total source function	

Defined and used in: `src/waves/config/mo_wave_nml.f90`

5. Namelist parameters for testcases (NAMELIST_ICON)

The ICON model code includes several experiments, so-called test cases, for the 2 and 3-dimensional atmosphere. Depending on the specified experiment, initial conditions and boundary conditions are computed internally.

5.1. nh_testcase_nml (Scope: ltestcase=.TRUE. 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. '**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.	 is_plane_torus=.TRUE.

Parameter	Type	Default	Unit	Description	Scope
				'**APE_nwp, APE_aes, APE_nh, APEc_nh,** ': Initializes the APE experiments. With the jabw test case, including moisture.	
				'**wk82**': Initializes the Weisman Klemp test case	l_limited_area =.TRUE.
				'**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	
				'**dcmip_bw_11**': Initializes (moist) baroclinic instability/wave (**DCMIP2016**)	
				'**dcmip_pa_12**': Initializes Hadley-like meridional circulation pure advection test case.	
				'**dcmip_rest_200**': atmosphere at rest test (Schaer-type mountain)	lcoriolis = .FALSE.
				'**dcmip_mw_2x**': nonhydrostatic mountain waves triggered by Schaer-type mountain	lcoriolis = .FALSE.
				'**dcmip_gw_31**': nonhydrostatic gravity waves triggered by a localized perturbation (nonlinear)	
				'**dcmip_gw_32**': nonhydrostatic gravity waves triggered by a localized perturbation (linear)	l_limited_area =.TRUE. and lcoriolis = .FALSE.
				'**dcmip_tc_52**': tropical cyclone test case with with full physics in Aqua-planet mode	lcoriolis = .TRUE.
				'**CBL**': convective boundary layer simulations for LES package on torus (doubly periodic) grid	is_plane_torus= .TRUE.
				'**bb13**': linear gravity- and sound-wave expansion in a channel (Baldauf, Brdar (2013) QJRMS)	is_plane_torus= .TRUE.
				'**SCM**' Single Column Mode	is_plane_torus= .TRUE.
is_toy_chem	L	.FALSE.		Terminator toy chemistry activated when .TRUE.	

Parameter	Type	Default	Unit	Description	Scope
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’
jw_u0	R	35.0	m/s	maximum zonal wind in jabw test case	nh_test_name=’jabw’
jw_temp0	R	288.0	K	horizontal-mean temperature at surface 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_width	R	1000.0	m	width of mountain	
mount_width_2	R	100.0	m	a 2nd width scale of mountain	nh_test_name=’schaer’

Parameter	Type	Default	Unit	Description	Scope
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'
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.
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'

Parameter	Type	Default	Unit	Description	Scope
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_aes'
ape_sst_val	R	29.0	degC	aqua planet SST for ape_sst_case='sst_const'	nh_test_name= 'APE_nwp', 'APE_aes'
linit_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 used to vary the buoyancy	nh_test_name='wk82'
u_infty_wk	R	20.	m/s	zonal wind at infinity height range 0. - 45. used to vary the wind shear	nh_test_name='wk82', 'bb13'
bub_amp	R	2.	K	maximum amplitud of the thermal perturbation	nh_test_name='wk82'
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_x	R	0.0	m	x-position of the center of the thermal perturbation	is_plane_grid=.TRUE.
bubctr_y	R	0.0	m	y-position of the center of the thermal perturbation	is_plane_grid=.TRUE.
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'

Parameter	Type	Default	Unit	Description	Scope
itype_topo_ana	I	1		kind of orography: 1 schaer 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
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
t_poly	R(nlayers_poly)	288., 213.	K	temperature at the base of each of the polytropic layers	nh_test_name= 'g_lim_area' and itype_atmo_ana=2
rh_poly	R(nlayers_poly)	0.8, 0.2	%	relative humidity at the base of each of the polytropic layers	nh_test_name= 'g_lim_area' and itype_atmo_ana=2

Parameter	Type	Default	Unit	Description	Scope
rhgr_poly	R(nlayers _poly)	5.e-5, 0.	%	relative humidity gradient at each of the polytropic layers	nh_test_name= 'g_lim_area' and itype_atmo_ana=2
nlayers_linwind	I	2		Number of the desired layers with constant U gradient	nh_test_name= 'g_lim_area' and itype_anaprof_uv=1
h_linwind	R(nlayers _lin-wind)	0., 2500.	m	height of the base of each of the linear wind layers	nh_test_name= 'g_lim_area' and itype_anaprof_uv=1
u_linwind	R(nlayers _lin-wind)	5, 10.	m/s	zonal wind at the base of each of the linear wind layers	nh_test_name= 'g_lim_area' and itype_anaprof_uv=1
ugr_linwind	R(nlayers _lin-wind)	0., 0.	1/s	zonal wind gradient at each of the linear wind layers	nh_test_name= 'g_lim_area' and itype_anaprof_uv=1
vel_const	R	20.	m/s	constant zonal/meridional wind (itype_anaprof_uv=2,3)	nh_test_name= 'g_lim_area' and itype_anaprof_uv=2,3
mount_lonc_deg	R	90.	deg	longitud of the center of the mountain	nh_test_name= 'g_lim_area'
mount_latc_deg	R	0.	deg	latitud of the center of the mountain	nh_test_name= 'g_lim_area'
schaer_h0	R	250.	m	h0 parameter for the schaer mountain	nh_test_name= 'g_lim_area' and itype_topo_ana=1
schaer_a	R	5000.	m	-a- parameter for the schaer mountain, also half width in the north and south side of the finite ridge to round the sharp edges	nh_test_name= 'g_lim_area' and itype_topo_ana=1,2
schaer_lambda	R	4000.	m	lambda parameter for the schaer mountain	nh_test_name= 'g_lim_area' and itype_topo_ana=1
lshear_dcmip	L	FALSE		run dcmip_mw_2x with/without vertical wind shear FALSE: dcmip_mw_21: non-sheared TRUE : dcmip_mw_22: sheared	nh_test_name= 'dcmip_mw_2x'
halfwidth_2d	R	10000.	m	half length of the finite ridge in the north-south direction	nh_test_name= 'g_lim_area' and itype_topo_ana=1,2
m_height	R	1000.	m	height of the mountain	nh_test_name= 'g_lim_area' and itype_topo_ana=2,3

Parameter	Type	Default	Unit	Description	Scope
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)	nh_test_name= 'g_lim_area' and itype_topo_ana=2,3
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

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

6. External data

6.1. 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)	
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(n_dom)	0.	m	RMS height difference to neighbor grid points at which the smoothing pre-factor fac_smooth_topo reaches its maximum value (linear proportionality for weaker slopes)	n_iter_smooth_topo > 0
heightdiff_threshold	R(n_dom)	3000.	m	height difference between neighboring grid points above which additional local nabla2 diffusion is applied	
pp_sso	I	1		1: Postprocess SSO standard deviation and slope over glaciers based on the ratio between grid-scale and subgrid-scale slope: both quantities are reduced if the subgrid-scale slope calculated in extpar largely reflects the grid-scale slope. 2: Optimized tuning for MERIT/REMA orography data: the reduction is also applied at non-glacier points in the Arctic, and the adjustment of the SSO standard deviation to orography smoothing is turned off.	n_iter_smooth_topo > 0
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
itype_lwemiss	I	1		Type of data used for longwave surface emissivity: 0: No data; use constant fallback value instead 1: Read and use emissivities derived in extpar from landuse classes 2: Read and use monthly climatologies derived from satellite measurements	itopo = 1

Parameter	Type	Default	Unit	Description	Scope
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/namelists/mo_extpar_nml.f90`

7. Serialization

Some developments must not change model results. Serialbox allows reading and writing data at any point in ICON into savepoints. These savepoints can be used to restore model variables to some reference or compare different model versions. The simplest application of Serialbox is using mo_ser_debug.f90 (or writing a similar routine fitting ones needs). Following this method will allow reading and writing manually specified fields in ICON. This can be very useful for small subroutines where input and output are clearly specified (i.e. do not involve derived types) and can thus easily be translated to Serialbox read/write statements. For larger components (basically everything hanging from nh_stepping.f90, e.g. nwp_physics) the interface is specified by the in and out types. The actual fields that are read or written to in these subroutines are not specified. For this purpose, serialize_all has been implemented. It provides a wrapper for Serialbox read and write statements by looping through variable lists. This approach does not require managing lists of fields to read or write by Serialbox. At the level of mo_nh_stepping.f90 and mo_nh_interface_nwp.f90 many components are wrapped by such serialize_all calls that allow testing these components. Each of these hard-coded calls to serialize_all has a name and for each name there is a namelist switch specifying the following triplet (e.g. 0,12,12):

- If 0 do not use this savepoint, else use this savepoint at every time step
- the relative threshold for errors (given as N for N in 10^{-N})
- the absolute threshold for errors (given as N for N in 10^{-N})

Parameter	Type	Default	Unit	Description	Scope
ser_initialization	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for initial data (Checked after regular initialization at model start as well as after initialization of nested domains during model run)	
ser_output_diag_dyn	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for output diagnostics of dynamics fields	
ser_output_diag	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for output diagnostics	
ser_output_opt	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for optional output	
ser_latbc_data	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine recv_latbc_data	
ser_nesting_save_progvars	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine save_progvars which is related to nesting	
ser_dynamics	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine perform_dyn_substepping	
ser_diffusion	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine diffusion	
ser_nesting_compute_tendencies	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine compute_tendencies (related to nesting)	
ser_nesting_boundary_interpolation	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine boundary_interpolation (related to nesting)	
ser_nesting_relax_feedback	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine relax_feedback (related to nesting)	
ser_step_advection	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine step_advection	
ser_physics	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_nh_interface	
ser_physics_init	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_nh_interface during initialization	
ser_lhn	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine organize_lhn	
ser_nudging	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the nudging computations	
ser_surface	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_surface	
ser_microphysics	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_microphysics	
ser_turbtrans	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_turbtrans	

Parameter	Type	Default	Unit	Description	Scope
ser_turbdiff	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_turbdiff	
ser_convection	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_convection	
ser_cover	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine cover_koe	
ser_radiation	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_radiation	
ser_radheat	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the computations involving radiative heating	
ser_gwdrag	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Serialization switch for the subroutine nwp_gwdrag	
ser_time_loop_end	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Check the state at the end of the time loop (does not read in data)	
ser_reset_to_initial_state	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Check the reset to initial state after the first phase of IAU	
ser_all_debug	I (3)	0,12,12	-, 10^{-N}, 10^{-N}	Additional calls to serialize_all (for debugging purposes) can be controlled using this switch.	
ser_nfail	R	1.0	%	Fields that fail more elements than the percentage specified by ser_nfail will be reported.	
ser_nreport	I	10		The detailed serialization report will include the ser_nreport elements with largest relative differences to the reference	
ser_debug	L	.FALSE.		Activates the debug serialization defined in mo_ser_debug.f90	

Defined and used in: `src/namelists/mo_ser_nml.f90`

8. External packages

8.1. Community Interface (ComIn)

More details on the Comminity Interface (ComIn) can befound in ComIn's documentation. This namelist is only usable (and needed) if ComIn has been enabled during ICON's configure process. Several `plugin_list(pg)` definitions can be specified, each holding the following namelist settings. The numbering (pg) of the plugins has to be contiguous without any gaps.

Parameter	Type	Default	Unit	Description	Scope
plugin_list(pg)%plugin_library	C	""		Path to the plugin library file. If omitted the primary constructor specified by `primary_constructor` is loaded from the ‘icon‘ executable (static linkage)	`configure -enable-comin`
plugin_list(pg)%name	C	""		Name of the plugin – currently only used for messages.	`configure -enable-comin`
plugin_list(pg)%primary_constructor	C	"comin_main"		Name of the symbol in the plugin library that holds the primary constructor.	`configure -enable-comin`
plugin_list(pg)%options	C	""		The options string passed to the plugin. This namelist parameter is necessary for certain plugins only, e.g. the Python adapter.	`configure -enable-comin`
plugin_list(pg)%comm	C	""		Name of the MPI communicator used in the second MPI Handshake. (At the first MPI Handshake "comin" is used). This namelist parameter is necessary only when using ComIn plugins in combination with external processes.	`configure -enable-comin`

9. Information on vertical level distribution

The atmospheric model needs hybrid vertical level information (i.e. the so called vertical coordinate tables `vct_a`, `vct_b` specifying the distribution of coordinate surfaces) to generate the terrain following height based coordinates. The 1D fields `vct_a`, `vct_b` are created within ICON during the setup phase, given that no input file is provided (`grid_nml:vct_filename=''`). For the SLEVE vertical coordinate (`ivctype=2`), the creation of `vct_a`, `vct_b` is controlled by the Namelist `sleve_nml` together with the parameter `num_lev` (`run_nml`). For the Gal-Chen vertical coordinate (`ivctype=1`), the user has only very limited control regarding its ICON internal creation. It is e.g. possible to create an equidistant level distribution for idealized testcases, by specifying the parameters `layer_thickness` and `n_flat_level` (`nh_testcase_nml`). For more general grids, it is recommended to read the vertical coordinate tables from file. Example files and information on the required format can be found in <icon home>/vertical_coord_tables, as well as in the ICON tutorial. Note that for the SLEVE coordinate, only `vct_a` must be provided in the input file. It is recommended to set `vct_b` to zero.

10. 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, run the configure script with the '--enable-mixed-precision' flag.

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 povided 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 described 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
`log()`, `exp()`	1	natural logarithm and its inverse function.
`sin()`, `cos()`	1	trigonometric functions
`sqrt()`	1	square root
`erf()`	1	Gauss error function
`min()`, `max()`	2	minimum and maximum of two values
`if(`*value*`,`*then*`,`*else*`)`	3	conditional expression (*value* `> 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 ommitted 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

Change:	**var_names_map_file, out_varnames_map_file**
Date of Change:	**2013-04-25**
Revision:	**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.

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

[1]This 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).

- 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 fiel dwdfg_filename. If false, surface analyis 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 invervals. 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_dblprec, 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 $0.2\,\mathrm{m}^2/\mathrm{s}$ to $0.75\,\mathrm{m}^2/\mathrm{s}$.

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:dtime` 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.

Change: **echam_cld_nml / echam_cov_nml**
Date of Change: **2019-06-07**
Revision: **icon-aes:icon-aes-cover 09233f275f207d59d2cb6ad75bd13adf81c0d0c2**

- The control parameters for the cloud cover parameterization (crs, crt, nex, jbmin, jbmax, cinv, csatsc) are shifted to the new namelist echam_cov_nml.

Change: **echam_cov_nml / echam_cov_nml**
Date of Change: **2019-06-12**
Revision: **icon-aes:icon-aes-cover 419e7ed54faa6db86a7151ece33b8e0b24737129 and e66e8e0f9cd439b81d7db63e0a4e03004d7f8144**

- The control parameters jks, jbmin and jbmax, specifying heights by the index of the vertical grid, are replaced by parameters zcovmax, zinvmax, and zinvmin, respectively, which directly specify the heights of interest. The change is as follows:
  - jks=15 –> zmaxcov=echam_phy_config%zmaxcloudy
  - jbmin=43 –> zmaxinv=2000m
  - jbmax=45 –> zmininv=300m

Change: **echam_cld_nml / echam_cld_nml**
Date of Change: **2019-06-12**
Revision: **icon-aes:icon-aes-cover ab95fc16a944dde96a76aeb1f63a7c847d78da06 and e66e8e0f9cd439b81d7db63e0a4e03004d7f8144**

- The control parameters jks, specifying height by the index of the vertical grid, is replaced by the parameters zcldmax, which directly specify the height of interest. The change is as follows:
  - jks=15 –> zmaxcld=echam_phy_config%zmaxcloudy

Change: **extpar_nml**
Date of Change: **2019-11-29**
Revision: **icon-nwp:icon-nwp-dev 21a16daf65aaf8df6fb581daa7dca66e2c915b94**

- The logical namelist parameter `l_emiss` has been replaced by the integer parameter `itype_lwemiss`. The code executed by default does not change.

Change: **transport_nml**
Date of Change: **2020-06-17**
Revision: **icon-nwp:icon-nwp-dev 616b4698e3a59c641a5ebe90637da2841c6f6a3a**

- The logical namelist parameter `lstrang` has been deleted. The default behaviour of the code is unchanged.

Change:	**extpar_nml**
Date of Change:	**2021-02-01**
Revision:	**icon-nwp:icon-nwp-dev ebac2edb0**

- The functionality of `itype_vegetation_cycle`=3 has been replaced by setting the new namelist parameter `icpl_da_sfcevap` in initicon_nml to a value of 1.

Change:	**ha_dyn_nml / ha_testcase_nml**
Date of Change:	**2021-03-29**
Revision:	**icon-nwp:icon-nwp-dev 599f03e5**

- The namelists for configuring the hydrostatic model `ha_dyn_nml` as well as the hydrostatic testcases `ha_testcase_nml` have been removed completely, as the hydrostatic model is no longer part of the official code.

Change:	**dynamics_nml**
Date of Change:	**2021-03-30**
Revision:	**icon-nwp:icon-nwp-dev 959fb5db**

- `iequations=0,1,2` (shallow water and hydrostatic atmosphere (T or $\theta \cdot dp$)) no longer supported.
- removed obsolete Namelist parameter `sw_ref_height` (reference height of shallow water model)

Change:	**diffusion_nml**
Date of Change:	**2021-04-16**
Revision:	**icon-nwp:icon-nwp-dev 806be7b0**

- removed obsolete Namelist parameter `k2_pres_max` and `k2_klev_max`, which were specific to the hydrostatic dynamical core.
- removed horizontal diffusion options `hdiff_order=24,42`

Change:	**transport_nml**
Date of Change:	**2022-05-07**
Revision:	**icon-nwp:master 8a351b13**

- removed Namelist parameter `iord_backtraj`, as the option for 2nd order accurate backward trajectory calculation has been removed. The default behaviour of the code is unchanged.

Change: **radiation_nml**
Date of Change: **2022-08-16**
Revision: **icon-nwp:master 6e49e2a7**

- removed unused Namelist parameter `ldiur`, `nmonth`, `lyr_perp` and `yr_perp`.

Change: **radiation_nml**
Date of Change: **2022-10-10**
Revision: **icon-nwp:master 61a1ac77**

- Removed Tanre aerosol option irad_aero=5.

Change: **radiation_nml**
Date of Change: **2022-11-03**
Revision: **icon-nwp:master 58a5aed0**

- Renamed ecRad-specific namelist settings `llw_cloud_scat` to `ecrad_llw_cloud_scat`, `iliquid_scat` to `ecrad_iliquid_scat` and `iice_scat` to `ecrad_iice_scat`.

Change: **nonhydrostatic_nml**
Date of Change: **2023-05-22**
Revision: **icon-nwp:master 1fff9207**

- Removed Namelist switch `l_open_ubc`. The upper boundary condition for vertical velocity w is unconditionally set to $w = 0$ (with the exception of vertically nested domains).

Change: **Optional output diagnostics, see table 22 on page 52**
Date of Change: **2023-06-13**
Revision: **icon-nwp:master 0d921fd4**

- Removed optional output diagnostics `vor_u` (*zonal component of relative vorticity*) and `vor_v` (*meridional component of relative vorticity*). Reason for removal: the two diagnostics have proven an unfavorable cost-benefit ratio.

Change: **nonhydrostatic_nml**
Date of Change: **2023-07-04**
Revision: **icon-nwp:master 6e5730d5**

- Removed Namelist switch `lhdiff_rcf`. Option to compute diffusion at dynamics time steps has been removed. It is only computed at advection time steps (in combination with divergence damping in the dynamical core).

Change: **parallel_nml**
Date of Change: **2023-07-25**
Revision: **icon-nwp:master 9a3c46e8**

- Removed Namelist switch `itype_comm`. Option to switch on asynchronous halo communication for the dynamical core has been removed.

Change: **nonhydrostatic_nml**
Date of Change: **2023-07-25**
Revision: **icon-nwp:master 9a3c46e8**

- Removed Namelist switch `nest_substeps`. Option to change the number of substeps for the child patches has never been functional.

Change: **diffusion_nml**
Date of Change: **2023-08-02**
Revision: **icon-nwp:master 78b68550**

- Removed option for Smagorinsky ∇^2 diffusion `hdiff_order=3`. Use `hdiff_order=5` in combination with `hdiff_efdt_ratio<=0` (deactivated background diffusion), instead.

Change: **gridref_nml**
Date of Change: **2023-08-07**
Revision: **icon-nwp:master 64ea30c9**

- Removed Inverse Distance Weighting (IDW) option for parent-child interpolation of edge-based variables. Options `grf_intmethod_e=1/3/5` are no longer available. The related namelist switches for specifying the exponent of the generalized IDW function `grf_idw_exp_e12/34` are removed as well.

Change: **upatmo_nml and nh_testcase_nml**
Date of Change: **2023-08-18**
Revision: **icon-nwp:master 255072ef**

- Removed Namelist switches `lnontrad`, `lconstgrav`, `lcentrifugal` and `ldeepatmo2phys` without substitution. From now on, `dynamics_nml/ldeepatmo = .TRUE.` means implicitly `lnontrad = .TRUE.`, `lconstgrav = .FALSE.` and `lcentrifugal = .FALSE.`. The switch `ldeepatmo2phys` has never been effective anyway.
  In addition, deep-atmosphere testcase `nh_test_name =` '**lahade**' has been removed without substitution.

Change: **gridref_nml**
Date of Change: **2023-08-14**
Revision: **icon-nwp:master 62819ba6**

- Remove optional mass conservation correction in incremental feedback routine `l_mass_consvcorr=TRUE/FALSE`.

Change: **nonhydrostatic_nml**
Date of Change: **2023-08-14**
Revision: **icon-nwp:master 62819ba6**

- Remove optional mass conservation correction for nested domains `l_masscorr_nest=TRUE/FALSE`.

Change: **nonhydrostatic_nml**
Date of Change: **2023-09-08**
Revision: **icon-nwp:master ff5a51a9**

- Removed Namelist switch `idiv_method` which allowed to select different methods for horizontal divergence computations. The possibility of divergence averaging has been removed. For divergence computation we now make unconditional use of the standard Gaussian integral in combination with averaged normal components of horizontal velocity.).

Change: **coupling_mode_nml**
Date of Change: **2023-09-07**
Revision: **icon-nwp:master e99436e7**

- Remove logical switch `coupled_mode`. Coupling with specific model components is now described by component specific switches.

Change: **dynamics_nml**
Date of Change: **2023-12-11**
Revision: **icon-nwp:master a1a19a69**

- Remove Namelist switch `iequations`. This switch became obsolete, as there exists only one set of governing equations for the atmosphere and ocean each.

Change: **lnd_nml**
Date of Change: **2025-02-10**
Revision: **icon-nwp:master xxx**

- Remove `itype_interception=2` option. The namelist switch `itype_interception` is still accepted but the model is aborted when set to a value different from 1.