Brazilian developments on the Regional Atmospheric Modeling System BRAMS - version 5.2

DESCRIPTION OF THE MODEL INPUT NAMELIST PARAMETERS

S. R. Freitas, L. F. Rodrigues, J. Panetta, K. Longo, D. Moreira, E. Freitas, M. Longo, A. Fazenda, R. Fonseca, R. Stockler, G. Camponogara
http://brams.cptec.inpe.br

Document Edition 1.0

March 2016

Updated document developed upon the previous versions

By

- Robert L. Walko and Craig J. Tremback (RAMS).
- Jairo Panetta, Saulo R. Freitas, Alvaro L.Fazenda, Edmilson D. Freitas and Eduardo H.Enari (BRAMS)

Table of Contents

Introduction

This document describes the input namelist parameters for the BRAMS v. 5.1 atmospheric model with recent developments. Specifying values for each of the variables in the atmospheric model namelists is the main way that a user may configure the model and select from the many options available for a particular model run. The namelists RAMSIN in BRAMS 5.1 are standard FORTRAN namelists.

The atmospheric model component of BRAMS contains 5 namelists, namely:

- **$MODEL_GRIDS**
- **$MODEL_FILE_INFO**
- **$MODEL_OPTIONS**
- **$MODEL_SOUND**
- **$MODEL_PRINT**,

The atmospheric chemistry modules (CCATT and TEB-SPM) coupled to BRAMS are described in 2 sections, one for each different model:

- **$CCATT_INFO**
- **$TEB_SPM_INFO**

while the initialization and boundary conditions ISAN namelist are:

- **$ISAN_CONTROL**
- **$ISAN_ISENTROPIC**.

Also new in this version, there are three additional namelists

$POST
$DIGITALFILTER
$METEOGRAM

These namelists are all contained in the data file called RAMSIN. This is a default name; the input namelist name can be set as a command line argument. Each namelist in this file begins with one of the above identifiers, and ends with the character string "$END". All variables in these namelists have sample values assigned to them in the **RAMSIN** file, but the user will need to edit many of them for a specific simulation or forecast. A sample RAMSIN file is available at ./BRAMS directory.

The following sections contain descriptions of all atmospheric model parameters and namelist variables. These descriptions include the meaning of each parameter or variable and provide acceptable or recommended settings for them. Cross-referencing to related variables provides additional information. For this reason, we recommend a complete read-through of all namelist variables. An index of the configuration and namelist variables can be found at the end of this document.

IMPORTANT: *Note as a matter of syntax that each **value** assigned to any namelist variable in the **RAMSIN** file must be followed by a comma.*

Brief description of the BRAMS development

BRAMS was originally developed as part of a joint project between ATMET/USA and the Brazilian institutions IME/USP, IAG/USP and CPTEC/INPE, and funded by FINEP (Brazilian Funding Agency). This project aimed to produce a new version of the Regional Atmospheric Modeling System (RAMS) tailored to address some of the environmental problems of the tropics. The main objective was to provide a single model for Brazilian regional weather and research centers.

The first version (BRAMS 1.0) was based on RAMS 5.0, with the inclusion of modeling of physical phenomena such as Shallow Cumulus and New Deep Convection (mass flux scheme with several closures, based on Grell et al., 2002), improvements in software quality (leading to binary reproducibility with just 1 grid and higher portability) and a higher resolution vegetation data file (1 km vegetation data derived from IGBP 2.0 + IBGE/INPE dataset LEAF-3 with observed parameters for South American biomes).

The version 2.0 was based on RAMS 5.04, and included all of modifications described above and a new surface parameterization using SiB 2.5 submodel, a new scheme to assimilate a heterogeneous Soil Moisture profile based on satellite data, binary reproducibility with nested grids, corrections for Lite and Mean variables output and improvements in software quality trough checking if all variables were properly initialized.

The following stable version, named BRAMS 3.2, was the first distributed in CC-GNU GPL in an official home-page (http://www.cptec.inpe.br/brams). It included the version 2.0 plus enhanced portability and software quality, improvements on the heterogeneous soil moisture assimilation procedure, an operational assimilation cycle and forecast procedure and improvements in serial and parallel performance.

The version BRAMS 4.0 was an extension of the version 3.2, with enhancements in portability (running in NEC SX-6 and others compilers) and software quality (including new procedure to read RAMSIN. It also featured a new scheme to build executable code, etc), improvements on serial and parallel performance (best vectorization rates in some codes, improvements on performance for advection scheme and improvements in master-slave communications), corrections in Shaved ETA scheme and LEAF scheme based on RAMS 6.x, inclusion of a new scheme for radiation (CARMA parameterization), inclusion of a new option to allow manual domain decomposition and also include the option of 2 new emission model running coupled to BRAMS: the CATT (Coupled Aerosol and Tracer Transport) scheme and TEB-SPM (Town Energy Budget – Simple Photochemical Module) scheme.

The current version of the BRAMS (5.2 as in December, 2015) has several new features described in the table below and illustrated by the figure 1.

BRAMS v. 5.1	Features
Advection for scalars	Highly accurate and monotonic scheme (Walcek 2000, Freitas et al. 2012)
Sub-grid scale convective transp. and wet deposition	Grell and Deveny (2002) cumulus scheme – (deep / shallow) New Cumulus scheme (aerosol and scale aware – Grell and Freitas 2014)
PBL Turbulence	Nakanishi & Nino (2004)
Dry deposition	Resistance approach coupled with surface and PBL schemes.
Emissions	MEGAN, EDGAR, RETRO, 3BEM, 3BEM-FRP (Pereira et al., 2009), GFED, Volcanoes. Urban emissions inventory for SA (Alonso et al., 2010).
Plume rise – Veg. Fires	1-D in-line cloud model forced by the fire heat flux (Freitas et al., 2007/2010)
Gas/Aqueous Phase Chemistry	SPACK pre-processor (RACM, RELACS, CB07, etc) with Rosenbrock 2^{nd} and 3^{rd} order solvers (Longo et al., 2013)
Photolysis	LUT, FAST-JX, FAST-TUV (in-line, aerosol and clouds effects)
Surface scheme	JULES with carbon cycle, biogenic VOCs (Moreira et al., 2013)
Radiation	CARMA (Longo et al., 2006, Rosário et al., 2013), RRTMG (LW, SW)
Aerosols	Simple mono-disperse for biomass burning/urban sources and the complex modal aerosol model "MATRIX" (Bauer et al 2008).
Aerosol direct / indirect effects	Included in radiation (CARMA, RRTMG), 2-moments cloud microphysics (RAMS and GT) and cumulus scheme (GF).
Microphysics	• Double moment from RAMS 6.0 CSU version • Single moment in cloud liquid water from G. Thompson (NCAR) • Double moment in cloud liquid water and aerosol aware from G. Thompson (NCAR).
Computational scalability	Efficiency at 10^3 - 10^4 cores (MPI)
Lagrangian Particle Model	Output to drive the STILT Lagrangian particle model.

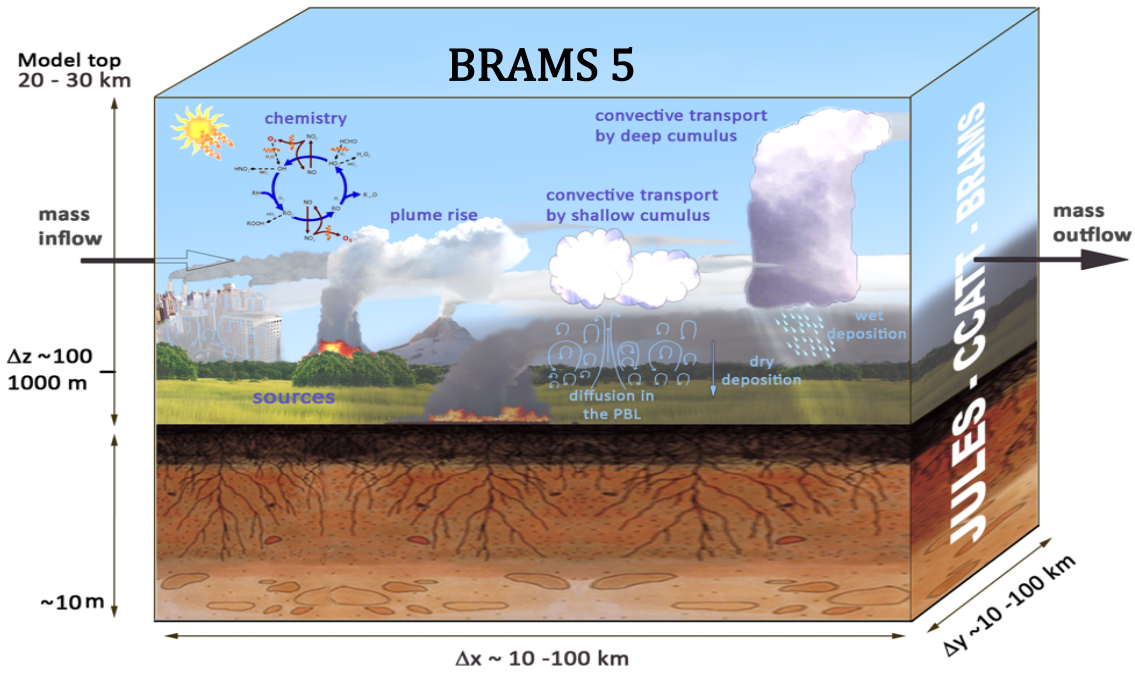

Figure 1. Several sub-grid scale physical and chemical processes simulated by the BRAMS v. 5.1+ model.

Some of the basic references for BRAMS 5.2 are described below:

1. Bela, M. M., Longo, K. M., Freitas, S. R., et al.: Ozone production and transport over the Amazon Basin during the dry-to-wet and wet-to-dry transition seasons, Atmos. Chem. Phys., 15, 757, 2015.
2. Grell, G. A., and S.R. Freitas: A scale and aerosol aware stochastic convective parameterization for weather and air quality modeling. Atmos. Chem. Phys., 14, 5233, 2014.
3. dos Santos, A. F., Freitas, S. R., de Mattos, J. G. Z., et al.: Using the Firefly optimization method to weight an ensemble of rainfall forecasts from the Brazilian developments on the Regional Atmospheric Modeling System (BRAMS). Adv. Geosci., 35, 123, 2013.
4. Moreira, D., Freitas, S. R., Bonatti, J. P., et al.: Coupling between the JULES land-surface scheme and the CCATT-BRAMS atmospheric chemistry model: applications to numerical weather forecasting and the CO2 budget in South America. Geos. Model Devel., 6, 1243, 2013.
5. Longo, K. M., Freitas, S. R., Pirre, M., et al.: The chemistry CATT-BRAMS model (CCATT-BRAMS 4.5): a regional atmospheric model system for integrated air quality and weather forecasting and research. Geos. Model Devel., 6, 1389, 2013.
6. Rosário, N. E., Longo, K. M., Freitas, S. R., et al.: Modeling South America regional

smoke plume: aerosol optical depth variability and shortwave surface forcing. Atmos. Chem. Phys. 13, 2923, 2013.

7. Freitas, S.R., L.F. Rodrigues, K.M. Longo, J. Panetta: Impact of a monotonic advection scheme with low numerical diffusion on transport modeling of emissions from biomass burning. Journal of Advances in Modeling Earth Systems, 4, M01001, 2012.
8. Freitas, S. R., Longo, K. M., Alonso, M. F. et al.: PREP-CHEM-SRC 1.0: a preprocessor of trace gas and aerosol emission fields for regional and global atmospheric chemistry models. Geosci. Model Devel., 4, 419, 2011.
9. Freitas, S. R., Longo, K., Trentmann, J., Latham, D.: Technical Note: Sensitivity of 1D smoke plume rise models to the inclusion of environmental wind drag. Atmos. Chem. Phys., 10, 585, 2010.
10. Longo, K. M., Freitas, S. R., Andreae, M. O., et al.: The Coupled Aerosol and Tracer Transport model to the Brazilian developments on the Regional Atmospheric Modeling System (CATT-BRAMS) Part 2: Model sensitivity to the biomass burning inventories. Atmos. Chem. Phys., 10, 5785, 2010.
11. Alonso, M. F., Longo, K. M., Freitas, S. R., et al.: An urban emissions inventory for South America and its application in numerical modeling of atmospheric chemical composition at local and regional scales. Atmospheric Environment, 44, 5072, 2010.
12. Pereira, G., Freitas, S. R., Moraes, E. C. et al.: Estimating trace gas and aerosol emissions over South America: Relationship between fire radiative energy released and aerosol optical depth observations. Atmospheric Environment, 43, 6388, 2009.
13. Freitas, S. R., Longo, K. M., Silva Dias, M. A. F., et al.: The Coupled Aerosol and Tracer Transport model to the Brazilian developments on the Regional Atmospheric Modeling System (CATT-BRAMS) – Part 1: Model description and evaluation, Atmos. Chem. Phys., 9, 2843, 2009.
14. Freitas, S. R., K. M. Longo, R. Chatfield, et al.: Including the sub-grid scale plume rise of vegetation fires in low resolution atmospheric transport models. Atmos. Chem. Phys., 7, 3385 2007.
15. Freitas, E. D.; Rozoff, C. M.; Cotton, W. R. et al: Interactions of an urban heat island and sea breeze circulations during winter over the Metropolitan Area of São Paulo - Brazil. Boundary - Layer Meteorology, v. 122(1), p. 43-65, 2007.
16. Freitas, S. R., K. M. Longo, M. Silva Dias, et al.: Monitoring the transport of biomass burning emissions in South America. Environmental Fluid Mechanics, 5, 135, 2005.
17. Nakanishi, M., H. Niino: An improved Mellor-Yamada Level-3 Model with condensation physics: its design and verification. Boundary-Layer Meteorol. 112, 1-31, 2004.
18. Miller, S. M, D. M. Matross, A. E. Andrews, et al.: Sources of carbon monoxide and formaldehyde in North America determined from high-resolution atmospheric data.

Atmos. Chem. Phys. 8, 7673-7696, 2008.
19. Medvigy, D., P. R. Moorcroft, R. Avissar, R. L. Walko: Mass conservation and atmospheric dynamics in the Regional Atmospheric Modeling System (RAMS). Environ. Fluid Mech., 5, 109-134, 2005.
20. Souza, E. P., Rennó, N. O., Dias, M. A. F. S.: Convective circulations induced by surface heterogeneities. Journal of the Atmospheric Sciences, v. 57, n.17, p. 2915-2922, 2000.

Model Configuration Parameters – Fortran 90 Module: "grid_dims.f90"

There are 11 parameters in the "grid_dims.f90" file (found at <BRAMS_ROOT_DIRECTORY>/src/brams/modules/), 9 of them are used as array dimensions for many variables in the in the atmospheric model global memory and 2 of then are used to set array dimensions exclusive to coupled TEB emission model. Whenever any of these parameters is changed, the entire model code must be recompiled, according to dependencies described in the build structure. These parameters set limits on maximum numbers of grid points, grids, etc. Normally, model simulations should run with the default dimensions, however, for very large simulations, one may need to increase these parameters. The parameters should be set large enough for the required model space, and for convenience, large enough to fit any anticipated expansions of required model space, but not so large that excessive computer memory is unnecessarily allocated. Most significant memory space is now dynamically allocated, so these parameters are not as important in determining memory usage as in the past. The grid_dims.f90 parameters are described in the following table.

MAXGRDS integer	The maximum number of grids that may be used in a model run. The number of grids actually used in a run is specified by the namelist variable **NGRIDS**. **NGRIDS** may be changed from one model run to the next using the same executable, as long as **NGRIDS** does not exceed the value of **MAXGRDS** that was set when compiling the model. Thus, **MAXGRDS** should be set to the largest number of grids that will be used in a series of runs made from the same compiled code.
NXPMAX integer	The maximum number of grid points in the x-direction to be used on any grid. The actual numbers of grid points spanning the x-direction on individual grids are specified in the multiple values of the namelist variable **NNXP**. **NXPMAX** must be equal to or larger than each of the **NNXP** values.
NYPMAX integer	The maximum number of grid points in the y-direction to be used on any grid. The actual numbers of grid points spanning the y-direction on individual grids are specified in the multiple values of the namelist variable **NNYP**. **NYPMAX** must be equal to or larger than each of the **NNYP** values.
NZPMAX integer	The maximum number of grid points in the z-direction to be used on any grid. The actual numbers of grid points spanning the z-direction on individual grids are specified in the multiple values of the namelist variable **NNZP**. **NZPMAX** must be equal to or larger than each of the **NNZP** values.
NZGMAX integer	The maximum number of vertical levels to be used on any grid in the soil model. The actual numbers of levels on individual grids are specified in the namelist variable **NZG**. **NZGMAX** must be equal to or larger than **NZG**.
MAXSCLR integer	The maximum number of scalars that may be automatically added to a simulation. The actual number of added scalars is specified in namelist

parameter **NADDSC**. **MAXSCLR** must be equal to or larger than **NADDSC**.

MAXDIM integer	**MAXDIM** must be set to the largest of the values in **NXPMAX**, **NYPMAX**, **NZPMAX**+10, and **NZGMAX**.
MAXMACH integer	The maximum number of nodes that may be used for a parallel model run. Applies only to running the model in parallel processing mode using MPI.
VARIABLES EXCLUSIVE FOR THE COUPLED TEB EMISSION MODEL	
MAXSTEB	Set the maximum number of layers used in TEB.
MAXUBTP	Set the maximum number of urban types used in TEB.

Model Namelists

The following sections will describe the input namelist parameters for the ***BRAMS*** and ***RAMS*** atmospheric model, the coupled ***CCATT*** and ***TEB-SPM*** chemistry models and the ***RAMS*/ISAN** (ISentropic ANalysis) package.

Specifying values for each of the variables in the atmospheric model namelists is the principal way that a user sets up the desired model configuration and selects the many options available for a particular model run. The namelists is a standard FORTRAN namelist.

The atmospheric model component of ***BRAMS/RAMS*** namelist contains 5 sections, which have the names **$MODEL_GRIDS**, **$MODEL_FILE_INFO**, **$MODEL_OPTIONS**, **$MODEL_SOUND**, and **$MODEL_PRINT**, for the emission models coupled to the atmospheric model BRAMS (CCATT and TEB-SPM), the namelist are: **$CCATT_INFO** and **$TEB_SPM_INFO**, while the ISAN namelist are called **$ISAN_CONTROL**, and **$ISAN_ISENTROPIC**. These namelists are all contained in the data file called *RAMSIN*. This is a default name; the input namelist name can be set as a command line argument. Each section of namelist in this file begins with one of the above identifiers, and ends with the character string "$END". All variables in these namelists have sample values assigned to them in the *RAMSIN* file, but the user will need to edit many of them for a specific simulation or forecast. In this version 5.1 of BRAMS, is not possible to run both chemistry models (CCATT and TEB-SPM) at the same time. This feature is planed to be available only in future versions.

$MODEL_GRIDS Namelist

The **$MODEL_GRIDS** namelist provides information to the model primarily on the structure of the one or more nested grids used in a simulation, including location, mesh size, number of mesh points, spatial nesting relationships, time step length, and time and duration of the run. Most of these variables are arrays dimensioned to the parameter **MAXGRDS**, which is set in the file **grid_dims.f90**. Each value in the array applies to a different grid, several of which may be activated if the user wishes to employ grid nesting. In the **$MODEL_GRIDS** and other namelists, the multiple values are separated by commas following the variable name and an equals sign. The *RAMSIN* file must have at least as many values of the grid-dependent variables specified as the value assigned to **NGRIDS**, but no more than the value of **MAXGRDS**. Values in the NAMELIST corresponding to grid number greater than NGRIDS are ignored by the model.

Variable Name / FORTRAN type	Description
EXPNME character	Character string of up to 64 characters into which the user may put any message (i.e., the experiment name) identifying the particular simulation or run. The character string is written in the standard output listing file and simply serves as a convenient means of labeling the output for the user's own purposes.
RUNTYPE character	Variable specifying one of six ways in which the model is to be run. In many situations, a RUNTYPE = 'MAKESFC' should be the first run that is made. This type of run will start the model and create surface characteristic datasets for all grids that will be used during subsequent runs. These characteristics include topography, soil textural class, sea surface temperature, vegetation class, and subgrid distribution of soil textural class, water surface, and vegetation type, all of which are available in standard RAMS global datasets from which they can be interpolated onto each model grid. These characteristics will be initialized on the model grids according to the specifications of ITOPTFLG, ISSTFLG, IVEGTFLG, ISOILFLG, and NDVIFLG, then surface files, one for each grid, be created with a filename prefix specified by the variable SFCFILES, and SST files, one for each grid and for each SST data time available, will be created with a filename prefix specified by the variable SSTFPFX. These files can then be quickly re-read on subsequent runs of the model or ISAN. An important benefit of the 'MAKESFC' option is that topography for all grids can bedefined with a fine grid present, and the fine grid can then be excluded for the first part of the simulation, to be added later on a history restart. This allows the model to utilize topography on its coarser grids that will be consistent with the future addition of the fine grid. If RUNTYPE is 'MAKESFC', the model run will stop immediately after the appropriate files are generated and will not proceed with integration. A value of 'MAKEVFILE' is the choice that runs ISAN which performs an objective analysis of one or more observational datasets, and produces one or more resultant *variable initialization files,* or *'varfiles'*. The varfiles contain a set of atmospheric fields defined on the model grids configured as specified by other variables in the **$MODEL_GRIDS** namelist, and are to be used in a later run for initializing a simulation, and optionally as 4-D assimilation data during the course of the simulation. The MAKEVFILE option is used only if the model is to be initialized from a complete 3-D objective analysis, (the INITIAL variable in the **$MODEL_GRIDS** namelist set to 2), and is the means by which the observational data are processed and interpolated to the model grid(s). The most common way to use this option is to generate varfiles from the objective analysis package every 12 hours, since that is the time interval at which upper-air data are normally collected, or every 6 hours to correspond with the NCEP reanalysis data times. The total number of varfiles required depends on the duration of the simulation to be performed and the particular grids which will be initialized from them (and optionally will use them for 4DDA). The first varfile must correspond to the starting time of the simulation, while the last varfile must equal to or later than the end of the simulation. If RUNTYPE is set to 'INITIAL', the current model run is designated to be the first of a simulation (see definitions of *run* and *simulation* in the entry for TIMMAX below). This means that a simulation is begun at time zero, and that all atmospheric and soil prognostic variables are initialized either horizontally homogeneously from sounding

	and soil data in the **$MODEL_SOUND** and **$MODEL_OPTIONS** namelists (INITIAL=1 in this case), read from a varfile prepared in an earlier ISAN run (INITIAL=2), or interpolated from a previous RAMS history file (INITIAL=3). If RUNTYPE is set to 'HISTORY', the model is to be history restarted, meaning that the atmospheric and soil prognostic variables are read from a history file, which was written by the model on a previous run. This option is used when a simulation is carried out over a series of two or more runs. (Also see IOUTPUT and HFILOUT.)
TIMEUNIT character	Time units in which the variables **TIMMAX** and **TIMSTR** are expressed. The allowable values of **TIMEUNIT** are: 's' to denote 'seconds', 'm' to denote 'minutes', 'h' to denote 'hours', 'd' to denote 'days'. This option allows numerical values for the above variables to be specified within convenient ranges for any simulation, which may range in duration from seconds to years.
TIMMAX real	Time during a simulation in units of seconds, minutes, or hours (see **TIMEUNIT**) when the current run is to stop. It is important to distinguish the terms simulation and run. A simulation is the entire integration of the atmospheric model from the initial state to the final time. It consists of one or more runs, which are individual submissions of the model code to the computer for execution. The **TIMMAX** variable represents the total time elapsed from the beginning of an entire simulation, starting with the first run, and is cumulative over successive runs. However, it denotes the time at which the current run is to terminate, rather than when the entire simulation is to end. (also see **TIMSTR**.) If **RUNTYPE** is set to 'MAKEVFILE' in order to run ISAN, **TIMMAX** specifies the duration of the time period over which to process observational data and generate varfiles.
LOAD_BAL integer	flag activating a dynamic load balance in a parallel run. This was an experimental capability and currently deprecated. The default value of zero must not be changed.
IMONTH1 integer	Month of the year when the simulation begins. It is used in conjunction with the namelist parameters **IYEAR1**, **IDATE1** and **ITIME1** to determine the proper solar declination angle for a simulation and to coordinate the model clock with dates and times of various observational datasets and vegetation seasonal cycles.If **RUNTYPE** is set to 'MAKEVFILE' in order to run ISAN, **IMONTH1** specifies the month of the beginning of the time period over which to process observational data and generate varfiles.
IDATE1 integer	Date of the month when the simulation begins. It is used in conjunction with the namelist parameters **IYEAR1**, **IMONTH1** and **ITIME1** to determine the proper solar declination angle for a simulation and to coordinate the model clock with dates and times of various observational datasets and vegetation seasonal cycles. If **RUNTYPE** is set to 'MAKEVFILE' in order to run ISAN, **IDATE1** specifies the date of the beginning of the time period over which to process observational data

	and generate varfiles.
IYEAR1 integer	Year when the simulation begins. It serves as an identifier of the simulation, along with **IMONTH1**, **IDATE1**, and **ITIME1** to coordinate the model clock with dates and times of various observational datasets. A 4-digit year must be specified for **IYEAR1**. If **RUNTYPE** is set to 'MAKEVFILE' in order to run ISAN, **IYEAR1** specifies the year of the beginning of the time period over which to process observational data and generate varfiles.

ITIME1 integer	Coordinated Universal Time (UTC) (or GMT) in hours and minutes (syntax is hhmm) when a model simulation begins (see definition of simulation in the description of TIMMAX). This parameter is used in conjunction with the namelist parameters IYEAR1, IMONTH1, and IDATE1 to determine the proper solar declination angle for a simulation and to coordinate the model clock with dates and times of various observational datasets and vegetation seasonal cycles. If RUNTYPE is set to 'MAKEVFILE' in order to run ISAN, IMONTH1 specifies the month of the beginning of the time period over which to process observational data and generate varfiles.
NGRIDS integer	Specifies the number of grids to be activated for the model run. If set to 1, only a single grid covering the entire spatial domain of the simulation will be activated. A setting of 2 will activate a nested grid within the first grid. A nested grid is used to attain a higher spatial resolution in a limited area of the simulation domain. This finer-resolution second grid communicates with the coarser-resolution parent grid via two-way interaction following the scheme designed by Clark and Farley [1] and Walko et al. [2]. Figure 1 shows an example of a lattice consisting of two small nested grids. The example is a horizontal cross section showing the relative positions of the horizontal velocity components (u, v) and the thermodynamic variables (at locations denoted by t) according to the Arakawa-C grid stagger used in RAMS. Upper-case letters denotes values on the coarse grid, while those on the fine grid are indicated in lower-case letters. NGRIDS can be set to as many nested grids as desired. However, the variable MAXGRDS must be set to a value at least as large as NGRIDS. It is recommended that NGRIDS be set to 1 the first time a simulation is begun, in order to test, at the relatively low cost incurred by the single grid, the remainder of the model configuration specified by the other namelist variables and other data inputs for the particular application. Once these tests are completed, standard procedure is to add finer grids when necessary in regions where higher resolution is desired, such as over complex terrain, or when trying to resolve a particular meteorological feature such as a sea-breeze circulation or a thunderstorm. It is common for a given simulation to consist of a series of sequential runs, each new run advancing the simulation forward in time beyond the previous run. RAMS allows the user to increase the value of NGRIDS from one run to the next (on a history restart), thus spawning new fine grids during the course of the simulation, or to decrease NGRIDS to remove one or more grids for the next run. It must be remembered that the variable MAXGRDS, needs to be set to at least to at least the maximum value of NGRIDS which will be used at any time during the simulation. Important: in BRAMS 5.x ONLY **one** grid is allowed. Working is in progress to restore the two-way nested grids capabilities of the previous versions.

NNXP integer array	Number of grid cells to span the computational domain in the x-direction. It is an array dimensioned to **MAXGRDS**, and has a value for each grid activated. These values are written in a single line in the namelist and are separated by commas. The first value in the line refers to grid number 1, the second to grid 2, and so on. The number of grid cells spanning each spatial direction (x,y,z) consists of interior cells where normal evaluation of all terms (such as advective tendencies) in the governing equations is performed, plus a boundary point at each end where variables are computed only by boundary conditions based on interior values. In the example of Figure 1, NXP (which is **NNXP** for grid 1) is set to 8, and the first and last grid cells in the x-direction (index values I = 1 and I = 8) are the boundary points. Thus, the actual computational domain may be considered to comprise only the interior cells, those shown bounded by lattice lines in the figure, and the user may wish to set **NNXP** to a value 2 larger than the intended domain size. For the U velocity points, boundary conditions are applied at I = 1 and I = 7, with normal evaluation of the governing equations being performed at points in between. The U values at I = 8 are dummy points and are never used. The fine grid in Figure 1 also has nxp (which is **NNXP** for grid 2) equal to 8. (Also see **NSTRATX** and **NINEST**). When ***RAMS*** is run on a global domain, values of **NNXP** and **NNYP** must be equal to each other on each hemispheric coarse grid and also equal between the two hemispheric coarse grids.
NNYP integer array	Similar to **NNXP**, but applies to the y-direction. The example in Figure 1 has NYP (**NNYP** for the coarse grid) equal to 9 and nyp (**NNYP** for the fine grid) equal to 6. If **NNYP** is set to 1, the simulation is two-dimensional, and **NNYP** must then also be set to 1 for all grids activated. A two-dimensional simulation may not be done with **NNXP** equal to 1 and **NNYP** greater than 1, *i.e.*, only x-z cross sections can be used for 2-D runs. However, the x-direction need not be aligned with geographic east, but may represent any compass direction with minor changes to the code or unusual placement of the polar stereographic pole point. Pseudo-one-dimensional simulations may be configured by setting **NNYP** to 1 and **NNXP** to 5 coupled with cyclic boundary conditions in the x-direction (see **IBND**). For the example in Figure 1, boundary conditions are applied to T and U at J = 1 and J = 9, and to V at J = 1 and J = 8. V at J = 9 is a dummy value and is never used. When ***RAMS*** is run on a global domain, values of **NNXP** and **NNYP** must be equal to each other on each hemispheric coarse grid and also equal between the two hemispheric coarse grids.
NNZP integer array	Similar to **NNXP**, but applies to the z-direction. Figure 2 shows an x-z cross section through the model grid analogous to the x-y cross section of Figure 1. Vertical indices for the coarse and fine grids are K and k, respectively, and NZP and nzp (which are the **NNZP** values for grids 1 and 2, respectively) are the corresponding upper limits of these indices. Vertical velocity components are W and w. The example consists of 7 interior grid cell levels on the coarse grid (K = 2 to K = 8 of T cells) with boundary cells K = 1 below and K = 9 above. The ground surface is located at the K = 1 W level, while the model top is considered to be at the K = 8 W level. It is at this top level where W is set to 0 for the rigid lid model top boundary condition. W at K = 9 is a dummy point and is never used. If the LEAF2 submodel is activated in ***RAMS***, **NNZP** must be at least **NZG**+**NZS**+3, and **NZG**+**NZS** must be at least 8. When ***RAMS*** is run on a global domain, values of **NNZP** must be identical between grid 1 and the other hemispheric coarse grid.

NZG integer	Number of soil layers to be used in LEAF-3, a submodel of ***RAMS*** that prognoses energy and moisture in soil, snowcover, vegetation, and canopy air. The soil model consists of a grid representing finite volumes in the uppermost meter or two of the ground. Soil temperature and moisture are prognosed on this grid based on equations governing their internal transport and external exchange with the atmosphere. The horizontal dimensions and resolution of the soil model are identical with those of the atmospheric model grid above, with **NNXP** and **NNYP** applying equally to both models. **NZG** can be set to around 10 for most applications, and **NZG+NZS** must be at least 8 if LEAF-3 is used.
NZS integer	Maximum number of snowpack layers allowed to occur in a simulation. It is relevant only when the LEAF-3 submodel is activated. In ***RAMS***, **NZS** must be at least 1, and has no upper limit other than a practical one. If snow is not expected in a simulation, there is no need for **NZS** to be greater than 1; in this case the layer will represent water ponding. If snowcover will occur, and if its evolution is of some importance to the simulation, a value around 5 is suggested for **NZS** in order to allow multiple snow layers when snowcover is sufficiently deep. **NZG+NZS** must be at least 8
NXTNEST integer array	Identifies the parent grid of each nested grid, *i.e.*, the coarser grid with which the nested grid has direct communication. Defining values in this array is necessary because different options are available for specifying the interrelation between grids. The rule must always be followed that all grids from 1 up to **NGRIDS** are numbered consecutively with no integers being skipped, and the coarser grid in which a finer grid is nested always has a lower grid number than that finer grid. We consider first the most common case where the ***RAMS*** domain is of limited area, covering no more than about a hemisphere of the earth. For this case, the above rule implies that grid 1 is always the coarsest grid, covering the entire computational domain, and grid 2 is always nested directly within grid 1. Thus, **NXTNEST** for grid 2 (the second value for **NXTNEST**) must always be set to 1. However, grid 3 can either be nested in grid 1 (by setting **NXTNEST** for grid 3 equal to 1), meaning that it is at an equal nesting level alongside grid 2, or grid 3 can be nested within grid 2 (by setting **NXTNEST** for grid 3 equal to 2), such that it is two nesting levels finer than grid 1. **NXTNEST** must be set to zero for any grid having no parent: Grid 1 never has a parent, and for a limited area domain, it is the only grid without a parent.

DOMAIN_FNAME character	File name that contains parallel decomposition of all grids, overwriting default BRAMS domain decomposition. Default domain decomposition is used if empty DOMAIN_FNAME, or if pointed file does not exist or if it exists but its contents are inconsistent with grid dimensions and number of processors. If file resides at the same directory as the executable, it suffices to inform file name. If at any other directory, it is required to inform file's full path. File should be ASCII, with as many lines as there are subdomains (one line for each slave process at each grid). Sub-domains and grids may be informed at any order, but the set of lines should partition all grids. Each line must contain the following six integers (in free- form format): grid number (from 1 to NGRIDS), node number (from 1 to number of slave processes), xbeg, xend, ybeg, yend. Example using 2 grids and 16 slaves nodes: 1 1 2 23 2 18 1 2 24 46 2 18 1 3 47 68 2 18 1 4 69 90 2 18 1 5 2 23 19 35 1 6 24 46 19 35 1 7 47 68 19 35 1 8 69 90 19 35 1 9 2 23 36 52 1 10 24 46 36 52 1 11 47 68 36 52 1 12 69 90 36 52 1 13 2 23 53 69 1 14 24 46 53 69 1 15 47 68 53 69 1 16 69 90 53 69 2 1 2 26 2 38 2 2 27 52 2 38 2 3 53 77 2 38 2 4 78 102 2 38 2 5 103 128 2 38 2 6 129 153 2 38 2 7 2 31 39 70 2 8 32 62 39 70 2 9 63 92 39 70 2 10 93 123 39 70 2 11 124 153 39 70 2 12 2 31 71 101 2 13 32 62 71 101 2 14 63 92 71 101 2 15 93 123 71 101 2 16 124 153 71 101
IF_ADAP integer	Flag to choose the vertical coordinate system. IF_ADAP=0: terrain-following sigma_z coordinate is used. IF_ADAP=1: the ADaptive APerture coordinate is used. The ADAP coordinate is a fully Cartesian grid where the grid cells intersect the topography. ADAP coordinate allows partial grid cells along the topography. Important: in BRAMS 5.x ONLY IF_ADAP = 0 is well tested and recommended. Also, the CCATT module and the output for STILT particle model are currently implemented only for IF_ADAP = 0

IHTRAN integer	Type of horizontal grid transformation to be used in a model simulation. IHTRAN=0: model grid will be Cartesian, with uniform horizontal spacing used throughout the domain. This option is often used in small computational domains, or where the simulation is of an idealized situation, not corresponding to any particular geographic location. The basic assumption of this grid is that earth curvature effects are negligible. **IHTRAN=1:** sets up a polar stereographic coordinate system for the simulation. While the model grid spacing itself appears uniform and Cartesian, map factors are activated which cause the earth distances mapped from model distances to be of variable resolution. A polar stereographic projection is a mapping between the spherical earth and a plane (assumed to correspond to zero elevation in the Cartesian model grid) tangent to the earth, with all projection lines emanating from the point on the earth's surface opposite or antipodal to the point of tangency. The Cartesian model plane is centered around the point of tangency, which may be placed anywhere on the earth (the pole in this case refers to the point of tangency, specified in namelist variables **POLELAT** and **POLELON**, not to either geographic pole). Thus, size ratio in the center of the model domain is unity between the model grid and the earth, and slowly increases outward toward the model lateral boundaries. This transformation allows a single model grid to cover geographic domains up to hemispheric in size, and avoids the numerical problems encountered near the geographic poles when latitude-longitude coordinates are used.
DELTAX real	Horizontal grid spacing or grid cell size in the x-direction of the model coarse grid (grid 1) in units of meters. This very important parameter determines the scale of meteorological features that are resolvable on the grid, and together with the number of grid cells in the x-direction, **NNXP**, determines the total x-direction span of the model domain. All spacings on the finer nested grids are derived from this value of **DELTAX** along with the individual grid spacing ratios **NSTRATX**. In choosing **DELTAX** for a simulation, the user needs to carefully consider, among other things, what essential meteorological phenomenon is to be modeled, what the required resolution is for adequately simulating this phenomenon, how large a domain is required to contain the environment of the phenomenon, what the duration of the simulation will be, whether nested grids will be used, and how much computer memory and CPU time are available.

DELTAY real	Similar to **DELTAX**, but specifies the y-coordinate, rather than x-coordinate dimension of a grid cell on the model coarse grid. One should set **DELTAX** and **DELTAY** to the same value, since different values may numerically force horizontal anisotropy in the simulated fields. **DELTAY** should still be set equal to **DELTAX** even in a 2D run.
DELTAZ real	Similar to **DELTAX**, but specifies the vertical, rather than x-coordinate, dimension of a grid cell on the model coarse grid. To be more precise, **DELTAZ** is the vertical grid spacing of the lowest level of the coarsest model grid (grid 1), and is equal to the vertical grid spacing at higher levels only if constant grid spacing is used in the vertical. (Also see **DZRAT** and **DZMAX**.) However, if **DELTAZ** is set to 0., it serves as a flag to instruct the model to read the **ZZ** variable from the namelist for obtaining vertical grid levels directly. As for the case of **DELTAX**, selecting an optimal value for **DELTAZ** involves a number of considerations. An additional consideration in this case is that a small value for **DELTAZ** in a domain containing steeply-sloping topography requires high horizontal resolution for computational stability. A rule of thumb here is that the terrain height difference between adjacent grid cells should not exceed about 3-5 times **DELTAZ**. If ***RAMS*** is run on a global domain, the second hemispheric coarse grid automatically has the same vertical grid spacing at all levels as grid 1.
DZRAT real	Used in conjunction with **DELTAZ**, and specifies the ratio in vertical grid cell dimension between adjacent levels on the coarse grid. **DZRAT** is used as a convenient means for vertically stretching the grid, which is a very common practice to obtain high vertical resolution near the ground and lower resolution at higher levels.This method establishes a geometrically-stretched grid, wherein the expansion ratio is constant between consecutive levels. Geometric stretching minimizes the maximum expansion ratio between consecutive levels for a given net stretching over a given number of vertical levels. It is desirable to keep the ratio between consecutive levels small, as large ratios destroy the second-order accuracy of the vertical differencing in the model. Values for **DZRAT** of 1.1 or even 1.2 are considered acceptable. To illustrate with an example, if **DELTAZ** is set to 100 and **DZRAT** is set to 1.1, the first (lowest) grid level will be 100 m thick, the next 1.1 times the first or 110 m, the third 1.1 times the second or 121 m, etc.
DZMAX real	Upper bound on vertical grid spacing anywhere in the model coarse grid, and is used in conjunction with **DZRAX** to prevent the automatic geometric grid stretching from causing overly large vertical grid cell sizes in the upper levels of the model. For example, if **DZRAT** is set to 1.1, the vertical grid spacing will increase by an order of magnitude over 25 levels. This means that if the first level above the surface is 200 m thick, the 26th level will be over 2 km thick. By specifying **DZMAX** to a number such as 1000, the geometric expansion will halt when the 1000 m vertical thickness is reached, and all levels above will be made 1000 m thick.

ZZ real array	Heights of coarsest grid levels in meters, beginning at the ground (**ZZ =** 0) and continuing to the top level in the model. The number of values specified for **ZZ** must equal **NNZP** for grid 1, which is specified in the **$MODEL_GRIDS** namelist. Specifying **ZZ** directly instead of using **DELTAZ**, **DZRAT**, and **DZMAX** is an alternative means of setting the vertical grid structure, and of course offers more flexibility because **ZZ** can be specified in many ways other than constant or geometrically expanded spacing. However, when using direct specification via **ZZ**, care needs to be taken not to cause two consecutive levels to have grossly different vertical thicknesses, such as a factor of 2, as this degrades the accuracy of the model's vertical differencing schemes. For the model to use the specified **ZZ** values, **DELTAZ** must be set to 0.
IDELTAT integer	Flag that controls how ***RAMS*** obtains values of **DTLONG**, **NNDTRAT**, and **NACOUST**. If **IDELTAT =** 0, values specified in the **$GRIDS** namelist are used and held fixed in time. If **IDELTAT =** 1, 2, 3, 4, or 5, ***RAMS*** will automatically compute values for **DTLONG**, **NNDTRAT**, and **NACOUST** based on grid spacings of all grids, model domain height, and other factors, and will hold these values constant in time. **IDELTAT =** 1 produces the longest, riskiest timesteps that experience has shown will still lead to stable model runs most of the time. **IDELTAT =** 2 produces somewhat shorter, safer timesteps that cause instability in even fewer cases, and higher values for **IDELTAT** cause progressively shorter long timesteps to be used. These larger values would be appropriate if it is expected that wind speeds on the grid will be relatively large. Since these parameters collectively determine the long and short timesteps on all grids, and hence the computational stability, if the CFL stability limit is exceeded and ***RAMS*** is instructed to keep **DTLONG**, **NNDTRAT**, and **NACOUST** fixed, the model will stop and output a CFL message. If **IDELTAT =** -1, -2, -3, -4, or –5, ***RAMS*** will compute initial values for **DTLONG**, **NNDTRAT**, and **NACOUST** as for **IDELTAT =** 1, 2, 3, 4, or 5, respectively. Additional constraints based on CFL criterion will be applied, both initially and during the model run, adjusting **DTLONG**, **NNDTRAT**, and **NACOUST** to more stable values when necessary and to less stable values when the model solution is well within the CFL limits. However, timesteps on individual grids will not be adjusted to longer, less stable values than specified by the **IDELTAT** flag. For example, if **IDELTAT** is set to –2, and the CFL criterion requires adjustment to smaller timesteps but subsequently allows the timestep to be lengthened, the timestep will not be lengthened beyond the values that correspond to **IDELTAT** = 2. If the CFL limit requires timesteps to become shorter than values that would correspond to **IDELTAT** = 5, the model will stop and output a CFL message.

DTLONG Real	Length of the timestep in seconds at which most processes on the coarse grid will be updated. This is the primary timestep in the model to which all others relate. For example, the values set for **NNDTRAT**, set integer ratios of timesteps between the coarse and fine grids, such that an appropriate fraction of **DTLONG** is the actual timestep used on a fine grid. One should set **DTLONG** reasonably close to, but below the limiting value for computational stability of the model. Given the full model equation set, this value is determined by a number of factors, including wind velocity, internal gravity wave speed, external gravity wave speed (related in part to the height of

	the model domain), vertical and horizontal grid spacing, and maximum terrain slope. It is not easy to provide a comprehensive formula for estimating an optimal value for **DTLONG**, and the user should generally try a few values when beginning a new simulation to find a suitable value. However, some guidelines can be provided based on experience. For a deep model domain, say 20 km or more, the model timestep can usually be 5 or 6 seconds for a 1 km horizontal spacing, 30 seconds for a 10 km spacing, 90 seconds for a 60 km spacing, and 180 seconds or more for spacings over 150 km. In a shallow domain 3 km deep, we have been able to set **DTLONG** to 6 seconds with a 125 meter horizontal spacing. When steep terrain is used (the relevant measure of steepness appears to be the maximum change in terrain height between two adjacent grid cells in vertical grid spacing units, with a value over about 5 being considered steep), **DTLONG** is often required to be smaller. If **IDELTAT** is not set to zero, the model will compute **DTLONG** itself based on grid spacing and other factors, and the value specified in the namelist will be ignored.
NACOUST **Integer**	Number of *short* or *acoustic* timesteps performed for each *long* timestep on a grid. If **IDELTAT** is set to zero, the value specified for **NACOUST** in the namelist is used for all grids. If **IDELTAT** is not equal to zero, the namelist value is ignored and the model will compute an appropriate value which may differ between grids. The *short* timesteps are used for evaluating the pressure gradient force and divergence equation terms and applying them to the velocity and pressure fields, *i.e.*, the terms involving the propagation of sound waves. The *long* timestep is used to evaluate most other terms, including advection, diffusion, Coriolis force, and microphysical tendencies. This practice of *time splitting* of the prognostic equations allows full explicit computation (on the short timestep) of the terms governing the rapidly moving sound waves, while the terms governing slower processes can be performed on the long timestep. **NACOUST** has nothing to do with the relation between timesteps of one grid and another. On the coarse grid, the long timestep is defined by **DTLONG**, defined above. A separate, grid-dependent variable **DTLONGN**, which is not defined in the namelists but is internal to the code, is the specific timestep length for each grid. It is equal to **DTLONG** (defined above) on the coarse grid, and is determined for the finer grids from a combination of **DTLONG** and **NNDTRAT**, defined below in this Section. The length of the short (sound wave) timestep is twice the ratio **DTLONGN/NACOUST**. This short timestep must be short enough (by using a sufficiently large value of **NACOUST**) that sound cannot travel more than one half the horizontal grid spacing during the short timestep. Sound normally travels more than 300 meters per second, but the model will artificially slow it down if on any grid the short timestep is too large for the grid spacing. A message is printed out stating how much the sound speed has been reduced, with a warning if it is too slow. If this occurs, **DTLONGN** should be reduced, either by reducing **DTLONG**, by increasing **NNDTRAT**, or both.

NSTRATX integer array	Number of grid cells in the x-direction in a fine grid which span the x dimension of a single grid cell of the coarser parent grid to that fine grid. In other words, **NSTRATX** is the factor of increase in resolution in the x-direction between consecutive nesting levels. For example, if the third value of **NSTRATX** is set to 4 and the third value of **NXTNEST** is set to 2, grid 2 will be the parent of grid 3, and grid 3 will have 4 times the resolution or one fourth the grid cell size of grid 2 in the x-direction. If in the same example, the second value of **NSTRATX** were also 4, grid 3 would have 16 times the resolution of grid 1 in the x-direction. The example shown in Figure 1 has **NSTRATX** set to 2 for grid 2. An important constraint between the values of **NSTRATX** and **NNXP** must be observed for fine nested grids. A fine grid must exactly cover an integer number of grid cells in its coarser parent grid with its *interior* cells (defined in the entry for **NNXP**), and must have the two additional boundary cells. Thus, **NNXP** for any nested grid must be some integer number (of coarser grid cells covered **-** 2 in the example of Figure 1) times **NSTRATX** for that nested grid (3 in the example of Figure 1) plus 2. For **NSTRATX** and for other related nest configuration variables, if the required combinations of these are not met, the model will stop and issue a message suggesting an acceptable value. Values of 3, 4, and 5 have been most commonly used for **NSTRATX**. A value of 2 is not usually considered a sufficiently large increase in resolution to make nesting worthwhile, while values of 6 and larger are more prone to cause reflections at the nesting boundaries due to the larger disparity of resolvability between the two grids. (Also see **NINEST**.)
NSTRATY integer array	Similar to **NSTRATX**, but applies to the grid resolution ratio in the y-direction. In the example of Figure 1, **NSTRATY** is set to 2. If **NNYP** is set to 1, so that the simulation is 2-D, **NSTRATY** must be set to 1 for all grids activated.
NNDTRAT integer array	Number of timesteps which a fine grid is to be executed to advance it in time the same amount as a single timestep of the coarser parent grid. Because the fine grid has a smaller cell size, it must normally be run with a smaller time step for numerical stability, and the time step length will automatically be made inversely proportional to **NNDTRAT**. As an example, if the second value of **NNDTRAT** is set to 4, and the timestep on grid 1 happens to be set to 20 seconds (see **DTLONG**), grid 2 will run on a 5-second timestep, four of which will be performed for each timestep of grid 1. The longest timestep on which a grid can be run stably is not directly proportional to the cell size, but tends to increase more slowly. Thus, a finer grid nested within a coarse grid at say a 3:1 ratio can normally be run at longer than the expected 1/3 of the coarse grid timestep; 1/2 of the coarse grid timestep may be acceptable. If **IDELTAT** is set to zero, the values specified in the namelist for **NNDTRAT** will be used. If **IDELTAT** is not zero, values specified in the namelist for **NNDTRAT** will be ignored and the model will instead compute them automatically.

NESTZ1 integer	Parameter whose absolute value denotes the number of the nested grid, if any, that is to be nested with a higher vertical resolution than its parent (see **NSTRATZ1**). Such vertical nesting may be done for only one of all model nested grids. Any additional grids that are nested within this grid will use the same vertical grid spacing. Because the vertical nesting ratio can be made variable with height between this nested grid and its parent, the vertical grid spacing of the nested grid may jump suddenly from one vertical level to the next. In order to reduce the relative amount of jump (that is, the ratio of vertical grid spacing between consecutive levels), the model can automatically compensate somewhat by making adjustments to the vertical grid spacings in the parent and coarser grids. This option is activated by giving **NESTZ** a negative value (of the grid to which it applies). If **NESTZ** is set to 0, no vertical nesting will be done.
NSTRATZ1 integer array	Related to **NSTRATX**, but applies to the grid resolution ratio in the z-direction. Unlike **NSTRATX**, however, **NSTRATZ** is not a grid-dependent parameter, but refers to the vertical nest ratio between a particular nested grid, specified by the user in the namelist variable **NESTZ**, and its parent. **NSTRATZ** is nevertheless a multi-valued parameter, to allow the vertical nest ratio to be varied with height. Each value specified signifies the number of fine grid levels contained in a single level of the parent grid, beginning with the K = 1 level of the parent grid and continuing upward (see Figure 2). Only those levels up to the last one having a vertical nest ratio greater than 1 need be entered in the namelist, because all others are assigned a value of 1 by default. Also, the first value specified in the namelist is not actually used, because it is reassigned in the code. Nevertheless, some value must be entered to serve as a placeholder in the namelist so that the subsequent values are in the proper location. This feature has most often been used to enhance vertical resolution close to the ground on a nested grid, in which case a decreasing sequence such as **NSTRATZ** = 4, 4, 4, 4, 3, 3, 3, 2, 2, 2, may be used. In this example, the lowest 3 levels of the parent grid will have 4 fine grid levels within each one, the next 3 will have 3 fine grid levels within each one, and the next 3 will have 2 fine grid levels within each one. The remaining levels will have a 1-to-1 vertical nesting ratio. The feature may also be used to enhance vertical resolution in a selected elevated region, such as a cloud layer. Values specified for **NSTRATZ** 1must not differ by more than 1 between consecutive entries. Use of vertical nesting is not necessarily the best means of increasing vertical resolution in all cases. The alternative method of increasing vertical resolution over a limited vertical extent of the model domain (but over the entire horizontal extent) has been to stretch the vertical grid spacing (see **DZRAT**). For this choice, **NSTRATZ** could be set to 1, although a combination of both methods of stretching is possible, if desired.
NESTZ2 integer	Like NESTZ1 except that it is used only if ***RAMS*** is run on a global domain, and applies only to the set of nested grids within the second hemispheric coarse grid (NESTZ1 will always apply to nested grids within the first (hemispheric) grid (grid 1).
NSTRATZ2 integer array	Like **NSTRATZ1** except that it is used only if ***RAMS*** is run on a global domain, and applies only to the set of nested grids within the second hemispheric coarse grid (**NSTRATZ1** will always apply to nested grids within the first (hemispheric) grid (grid 1). **NSTRATZ2** may be different from **NSTRATZ1**, allowing different nesting design in each hemisphere.

POLELAT **POLELON** real	Geographic latitude and longitude in degrees (ranging from -90 to 90 and -180 to 180, respectively) of the pole point, or point of tangency between the model polar stereographic grid and the earth. This location also serves as the origin (x = 0,y = 0) on the polar stereographic grid. If ***RAMS*** is run on a global domain, the second hemispheric grid and any nested grids within it use a polar stereographic projection tangent to the earth at the point opposite **POLELAT** and **POLELON**.
CENTLAT **CENTLON** real array	Grid-dependent parameters for the geographic latitude and longitude in degrees (ranging from -90 to 90 and -180 to 180, respectively) of the center of each model grid. They are used to define the position of the grids in earth coordinates for the purpose of determining the Coriolis force, the solar radiation, and the location of the model domain relative to observed meteorological and land surface data. For the coarsest model grid, **CENTLAT** and **CENTLON** are the only means of specifying the center location. Nested grid locations may also be specified relative to their parent grids from variables **NINEST**, **NJNEST**, **NSTRATX**, and **NSTRATY**. Only if **NINEST** and **NJNEST** are given zero values will **CENTLAT** and **CENTLON** be used to determine the nested grid location. In such cases, because of the precise positioning of grid lattice points required between a nested grid and its parent, **CENTLAT** and **CENTLON** will be only approximate, and the precise location of the nested grid will be the closest allowable position to that which they indicate.If ***RAMS*** is run on a global domain, values of **CENTLAT** and **CENTLON** are ignored for the two coarse hemispheric grids. **CENTLAT** and **CENTLON** for grid 1 are set equal to **POLELAT** and **POLELON**, and **CENTLAT** and **CENTLON** for the other hemispheric coarse grid are set to the location on earth opposite **POLELAT** and **POLELON**.
NINEST integer array	Location within a coarser parent grid where the western edge of a nested grid is to be located. To be specific, we refer to the example in Figure 1. Here, the western edge of grid 2 (the western edge always being defined by the lattice line passing through the u-velocity points having an index i = 1) is located along the coarse mesh line passing through the U-velocity points having an index I = 3. This configuration would be established by setting the second value (for grid 2) of **NINEST** to 3. Note that the value of **NINEST** for a nested fine grid always refers to the I-index in the coarser parent grid of a row of U-velocity points. An important constraint involving **NINEST**, **NNXP**, and **NSTRATX** is that the edge of a fine nested grid cannot be any closer than two cells of the coarser parent grid from the lateral parent grid boundary. The example in Figure 1 is right at this limit on both the eastern and western boundaries of the fine grid: **NINEST** is set to 3, which places the western boundary of the fine grid just two coarse cells from the coarse-grid western boundary, and the eastern boundary of the fine grid is along the I = 5 index row of coarse grid U-velocity points, which is just 2 coarse cells away from the coarse-grid eastern boundary at I = 7.
NJNEST integer array	Similar to **NINEST**, but locates the southern boundary of a fine grid within the coarser parent grid. Here, the J-index of the coarse-grid V-velocity points is what is referred to by the value of **NJNEST**. In the example of Figure 1, the second value of **NJNEST** (for grid 2) would be 3. The constraints described for **NINEST** likewise apply to **NJNEST** for the y-direction.

NKNEST integer array	Related to **NINEST**, but locates the bottom boundary of a fine grid within the coarser parent grid. Here, the K-index of the coarse-grid W-velocity points is what is referred to by the value of **NKNEST**. Unlike the case for **NINEST** and **NJNEST**, **NKNEST** can be set to 1, if desired. Doing so would cause the nested grid to begin at the ground, like the coarser parent grid. If the fine nested grid is to begin at some height above the ground, **NKNEST** can be set to a larger number. **NKNEST** can never be set to 2, however; nor can the combination of **NKNEST**, **NSTRATZ**, and **NNZP** be such that the top of a fine nested grid be located exactly one cell of the coarser parent grid below the top of that parent grid. Another constraint is that **NKNEST** may not be set to 1 for a nested grid if its direct parent grid does not itself begin at the ground. The configuration where a fine nest does not extend to both the ground and the model domain top has been relatively little used. In some cases, vertical motion impinging on a top or bottom nesting boundary of differential horizontal resolution causes strong numerical reflections. In addition, two model algorithms that involve vertical transport, radiative transfer and sedimentation of hydrometeors have not been programmed for transfer between different nested grids. For this reason, a nested grid should begin at the ground and extend to the model top except in a few specialized applications.
NNSTTOP integer array	Grid-dependent flag to indicate whether each grid extends to the coarse grid top. It must be set to 1 to indicate that a grid does reach the domain top, and set to 0 to indicate that the top boundary of a grid is below the model domain top. In view of the recommendation given under the entry for **NKNEST**, **NNSTTOP** should normally be set to 1. Of course, since grid 1 always extends to the model domain top, the first value of **NNSTTOP** must always be set to 1. The specified values of **NNZP**, **NKNEST**, and **NSTRATZ** will themselves determine where the top of a fine nested grid is in relation to the model domain top. The user must insure that this is in agreement with the setting of **NNSTTOP**. If not, the model will give a fatal error message and stop.
NNSTBOT integer array	Similar to **NNSTTOP**, but indicates whether a nested grid has its lower boundary at the lower model boundary (the ground) or above it. **NNSTBOT** must be set to 1 if the fine nested grid begins at the ground, and 0 if it does not. Similar to the case for **NNSTTOP**, the model will check for agreement between **NNSTBOT** and the values of **NKNEST**, and will stop with a fatal error message if agreement is not met.

GRIDU **Real array**	*Currently not working in v5.x*
GRIDV **Real array**	*Currently not working in v5.x*

3. The **$CCATT_INFO** namelist consists of variables that control the activation of the Coupled Chemistry-Aerosol and Tracer Transport (CCATT) chemistry model, and specifies the variables needed.

CCATT Integer Range: 0, 1	The main flag to activate the chemistry and aerosol models in BRAMS 0= OFF 1= ON
CHEMISTRY integer: Range: -1 to 4	Defines the complexity level of chemistry/aerosol sub-models to be executed. **-1:** this option turn OFF the chemistry/aerosol models. Only the meteorological model will be executed. **0**: activates the tracer transport model where emissions, advection, diffusion and dry and wet depositions are turned ON but not chemical reactivity. All processes are in-line and consistent with the meteorological model. **1**: the same as 0 but chemical reactivity is turned ON using the QSSA solver. **Important**: this solver is obsolete and should not be used. Future model versions will remove this option. 2: the same as 0 but chemical reactivity is turned ON using the Rosenbrock 2nd order solver. 3: the same as 0 but chemical reactivity is turned ON using the Rosenbrock 2nd order solver together with a varying chemical time step to optimize accuracy, stability and computational efficiency. 4: the same as 0 but chemical reactivity is turned ON using the Rosenbrock 3rd order (RODAS3) solver together a varying chemical time step to optimize accuracy, stability and computational efficiency. The option 4 is strongly recommended for simulations with chemistry, mainly for long-term runs.
SPLIT_METHOD Character	Defines the kind of splitting operator to be applied to integrate the full mass continuity equation for each chemical tracer. The values for this variable are: • **'PARALLEL':** parallel time splitting as originally developed by the RAMS model. The same as applied for the scalars theta_il, tke, etc.

	In this this case, the parameter CHEM_TIME_STEP (see below) must be set equal do 'DTLONG'. • **'SEQUENTIAL':** sequential (or serial) time splitting. In this method, the tracer chemical mixing ratio is first integrated using the transport, emission and removal processes. Then using the mixing ratio just updated, the chemical solver is applied. • **'SYMMETRIC':** sequential, symmetric time splitting: dynamics ->chemistry -> dynamics. Important: SYMMETRIC splitting is recommended
CHEM_TIMESTEP Real	This parameter provides de maximum time step to be applied by the chemical solver. It is be specified in seconds and must be a multiple of the dynamics time step (the RAMSIN parameter named "dtlong"). In this way, it must be defined as: **CHEM_TIMESTEP = N* DTLONG** Where : N = 1, 2, 3 ,4, 6 for the parallel or sequential splitting methods. N = 2, 4, 6 for the symmetric splitting method. As an example, if the user chooses dtlong = 10, chem_timestep should be set as 10, 20, 30, 40 or 60. Important: it is strongly recommend do not use N greater than 6 for keeping reasonable accuracy of the chemical solver. The recommended optimal value (accuracy and efficiency) for the parameter N is 4. However, it is always recommended to the users to make sensitive testing of your case study using N from 1 to 6 to check the convergence of the solution and if the error is acceptable.
CHEMISTRY_AQ Integer Range: 0,1	Parameter that activates (1) or deactivates (0) aqueous phase chemistry. Only for RACM chemical mechanism.
CHEM_ASSIM Integer Range: 0, 1	Activates the four-dimensional data assimilation (nudging) of chemical species together with the meteorological fields. 0 : turn OFF 1 : activates
RECYCLE_TRACERS Integer Range: 0, 1	Activates the tracer initialization from a previous model run. 0 : turn OFF 1 : activates When activates, the model analysis used to initiate the tracer field must be informed at RAMSIN parameter PASTFN do RAMSIN (section $MODEL_FILE_INFO). As an example: PASTFN = 'RACM-A-2005-06-30-000000-head.txt',
SRCMAPFN Character	PREFIX nama of the emission files. As an example : SRCMAPFN = '/user1/model/SRC_RACM/experiment-saopaulo', In case of emission will not be included in the model simulations, set

	SRCMAPFN = **'NONE'**
DEF_PROC_SRC Character	Tells to the model what to do in case of an emission file does not exist or is not find in the directory during the run time: 'STOP' : model execution is aborted. 'LAST_SOURCES': model uses the previous day emission file (model will repeat the emission fields for the current day simulation).
DIUR_CYCLE Sequence of 4 integers with the range 0 or 1.	A value of 1 will apply prescribed diurnal cycles of emissions. This is the default setting. In this case, emission files will be read each 24 hours. A value 0 is reserved for future implementation of hourly interpolation. The order of these parameters are: 'antro ' ,'bburn', 'bioge ,'geoge'
NA_EXTRA2D NA_EXTRA3D Both integers Range >= 0	Number of extra arrays 2d (nx,ny) and 3d (nz,nx,ny) to be allocated for the user customized fields.
PLUMERISE Integer Range : 0 or 1	Controls the activation of the plume rise transport of emissions associated with biomass burning. 0 : turn off (all emissions are released in the first model layer) 1 : activates the plume rise mechanism for the flaming phase emissions. An injection layer is prognostic in-line with the model evolution for the fraction of the emissions produced during the flaming phase.
PRFRQ Real	Set the frequency (in seconds) that the plume rise model is called to update the injection height of biomass burning emissions. Must be a multiple of DTLONG parameter. Typical tested values are 1800, 3600 and 7200 seconds.
VOLCANOES Integer Range: 0 or 1	Controls the activation of volcanic emissions (currently SO2 and ash). 0 : turn OFF 1 : activates
AEROSOL Integer Range: 0, 1, 2	Parameter that activates or not the aerosol models 0 : turn OFF 1 : activates the simplified aerosol model. 2: activates MATRIX aerosol model (currently, not fully tested yet) Important: The direct effect of aerosols is implemented in CARMA radiation scheme only (ISWRTYP = 4 and ILWRTYP = 4)
AER_TIMESTEP REAL Range: 1 to 4 times dtlong	Aerosol model time step integration (sec). Must be an multiple of dtlong (max 4) only for the matrix aerosol model (AEROSOL=2)

.

$TEB_SPM_INFO namelist

The **$TEB_SPM_INFO** namelist consists of variables that control the activation of the Town Energy Budget (TEB) and the Simple Photochemical Module (SPM) emission model, and specifies the variables needed in this special case of simulation.

Variable name	Description
TEB_SPM	Flag that allows active (1) the TEB_SPM section, including options to run the Town Energy Budget (TEB) and the Simple Photochemical Module (SPM) emission model, or deactive (0).
FUSFILES Character array	File path and prefix for local time files. If the file does not exist the model will generate from a global time files (defined in IFUSFN).
IFUSFLG Integer array	Array of integer where the user can set values for each grid used. The value 1 set the model to read from standard Lat/Lon local time data file, 0 define that the model will calculate the values of local time.
IFUSFN Character array	File path and prefix for global time files.
ICHEMI Integer	Photochemical module activation –(1=on, 0=off). Activation requires ISOURCE to be equal to 1
ICHEMI_IN Integer	Use initial values from previous run (1=yes,0=no) for Photochemical data.
CHEMDATA_IN Character	File with Photochemical data from a previous run to be used as initial value (if ICHEMI_IN=1).
ISOURCE Integer	Urban vehicular and industrial emission activation –(1=on, 0=off).
WEEKDAYIN Character	Initial weekday of the simulation. This will be also used by TEB.
RUSHH1 Real	Morning Rush Hour (Local Time in Hours).
RUSHH2	Afternoon/Evening Rush Hour (Local Time).

Real	
DAYLIGHT Real	Daylight saving time.
EFSAT Real	Emission factor (fraction of weekdays) for Saturdays. It is used in the emission module and TEB.
EFSUN Real	Emission factor (fraction of weekdays) for Sundays. It is used in the emission module and TEB.

EINDNO Real	Industrial emissions for NO in kg/s/m2.
EINDNO2 Real	Industrial emissions for NO2 in kg/s/m2.
EINDPM Real	Industrial emissions for Particulate Material in kg/s/m2.
EINDCO Real	Industrial emissions for CO in kg/s/m2.
EINDSO2 Real	Industrial emissions for SO2 in kg/s/m2.
EINDVOC Real	Industrial emissions for VOC in kg/s/m2.
EVEINO Real	Vehicular emissions for NO in kg/day/m2.
EVEINO2 Real	Vehicular emissions for NO2 in kg/day/m2.
EVEIPM Real	Vehicular emissions for Particulate Material in kg/day/m2.
EVEICO Real	Vehicular emissions for CO in kg/day/m2.
EVEISO2 Real	Vehicular emissions for SO2 in kg/day/m2.
EVEIVOC Real	Vehicular emissions for VOC in kg/day/m2.
TMINBLD Real	Minimum internal building temperature (degrees Celsius)
NTEB Integer	Number of roof, road and wall layers used in TEB (Maximum of 3)
HC_ROOF Real	Heat capacity for ROOF layers.
TC_ROOF Real	Thermal conductivity for ROOF layers.

D_ROOF Real	Depth for ROOF layers.
HC_ROAD Real	Heat capacity for ROAD layers.
TC_ROAD Real	Thermal conductivity for ROAD layers.
D_ROAD Real	Depth for ROAD layers.
HC_WALL Real	Heat capacity (J/m3/K 10e6) for WALL layers.
TC_WALL Real	Thermal conductivity (0.81 W/m/K) for WALL layers.
D_WALL Real	Depth for WALL layers.
NURBTYPE Integer	Number of urban types (maximum of 3).
ILEAFCOD Integer array	Leaf class code to identify each urban type. The number of elements for this array is determined by the values in NURBTYPE.
Z0_TOWN Real array	Urban type roughness length (between 5 and 1). The number of elements for this array is determined by the values in NURBTYPE.
BLD Real array	Fraction occupied by buildings in the grid cell for each urban type.
BLD_HEIGHT Real array	Building Height for each urban type.
BLD_HL_RATIO Real array	Vertical/Horizontal rate (between 3 and 0.5) for each urban type.
AROOF Real array	Roof albedo for each urban type.
EROOF Real array	Roof emissivity for each urban type.

AROAD Real array	Road albedo for each urban type.
EROAD Real array	Road emissivity for each urban type.
AWALL Real array	Wall albedo for each urban type.
EWALL Real array	Wall emissivity for each urban type.
HTRAF Real array	Maximum value of sensible heat released by Traffic (W/m2) for each urban type.
HINDU Real array	Maximum value of sensible heat released by Industry (W/m2) for each urban type.
PLETRAF Real array	Maximum value of latent heat released by Traffic (W/m2) for each urban type.
PLEINDU Real array	Maximum value of latent heat released by Industry (W/m2) for each urban type.

$MODEL_FILE_INFO Namelist

The **$MODEL_FILE_INFO** namelist consists primarily of variables that control data input to and data output from the model. The names of files containing these data are specified here, as well as the times during a model simulation when the data are read or written. The information in this namelist also controls some aspects of how input data is to be used.

INITIAL integer	Specifies how atmospheric fields in the model are to be initialized. **INITIAL** = 1: the atmospheric variables are initialized horizontally homogeneously from a single sounding, which the user supplies in namelist $MODEL_SOUND. This option is typically used when the model domain is of relatively limited size (a few hundred kilometers across or less), and is required if the model is run in 2-D. Lateral boundaries in this case are open, and are updated based on interior values or are held constant in time. **INITIAL** = 2: the sounding in namelist $MODEL_SOUND is not used, and instead, the model inputs complete 3-D fields of atmospheric data from *varfiles,* which have been earlier generated by the model from isentropic output from ISAN (see **RUNTYPE).** These fields are used both as initial fields in the model and as time-dependent fields to which the lateral boundary region of the model is nudged during integration. **INITIAL** = 3: the initial fields will be defined by interpolation from a previous RAMS' history file, as specified in HFILIN. The new run will start at model time 0. The grids in the current run do not have to have the same structure as the grids on the history file, except that the new grid 1 must fit within the history file grid 1. All other nested grids in the current run are interpolated from the current coarse grid by standard nesting procedures. This capability is mainly intended for situations where the new coarse grid is very different than the old grid structure. This should not be used if a nested grid is added or subtracted; use the standard **RUNTYPE='HISTORY'** instead. **INITIAL =** 4: initial fields are defined by a using prior model's analysis output (in vfm format), like a downscaling. The prefix name and directory address of these files are set using the **VARFPFX** variable. Further details regarding the use of the initialization options are given in descriptions of several variables in this Section.

NUD_TYPE integer	Specifies what type of analysis nudging scheme to use in the 4-dimensional data assimilation (4DDA). The term analysis nudging refers to a scheme where all or a portion of the domain is nudged toward a gridded data analysis. This can also be applied to nudging toward a previous simulation of RAMS or other model output. NUD_TYPE = 0: no analysis nudging is performed. This would generally be the setting if INITIAL=1. NUD_TYPE = 1: analysis nudging is performed using data from RAMS history files, with the nudging strengths and settings controlled by the analysis nudging parameters below. All grids are interpolated or defined from the highest resolution information available on the history file. This is a way to activate a one-way nesting scheme, although we only recommend using a one-way nest only in special circumstances. NUD_TYPE = 2: analysis nudging is performed using data from the varfiles, with the nudging strengths and settings controlled by the analysis nudging parameters below. NUD_TYPE = 3 is not in use. NUD_TYPE = 4: analysis nudging is performed using output analysis data from a prior run (see description about the ÏNITIAL" parameter), with the nudging strengths and settings controlled by the analysis nudging parameters below.
VARFPFX character	Prefix that begins the names of the initialization files called varfiles, (including path names if applicable) to be read into the model, for cases when the INITIAL=2 (and/or NUD_TYPE=2). These files are output from a previous model run with RUNTYPE set to 'MAKEVFILE' (see VARPFX). If INITIAL is set to 1 or 3, this parameter is ignored. For cases when the INITIAL=4 (and/or NUD_TYPE=4) the files come from a previous model run.
VWAIT1 real	Used only for automated operational model runs in which an attempt is made to acquire the varfiles data for input to RAMS. If this attempt is unsuccessful and a varfile is not available, the model is instructed to wait a time interval given by VWAIT1 (seconds), before attempting again.

VWAITTOT real	Similar to VWAIT1 and likewise used only for automation of operational runs. Specifies the total amount of time to continue waiting for a completed varfile before giving up altogether.
NUD_HFILE character	Analysis state files to be read for use in the history file nudging scheme (NUD_TYPE=1). The character string placed in this variable must be the full name of the state header file, including the appropriate path if applicable. However, only the prefix information is used to determine the set of state files to be used. Further details on the standard syntax of the analysis file names are given in the description of AFILOUT.

NUDLAT integer **TNUDLAT** **TNUDCENT** **TNUDTOP** **ZNUDTOP** real	Parameters that control the option of four-dimensional data assimilation (4DDA) by Newtonian relaxation (nudging). This is also termed *analysis nudging*, since we will be nudging toward a gridded data analysis. This 4DDA option requires that the **INITIAL** variable is set to 2 or 3, so that the model is initialized from varfiles or a previous RAMS run. The varfiles or history files contain time series of gridded horizontal wind, potential temperature, and total water mixing ratio values, usually analyzed from observations or sometimes large-scale model forecasts. The atmospheric model solution is relaxed toward the analyzed data during time integration. The strength of the nudging is given by (I-M)/T, where I is a varfile data value at a particular location, M is the corresponding model value, and T is a user-specified relaxation (e-folding) time scale. ***RAMS*** contains a 3-D array of T values for each grid so that the nudging strength can be specified in a customized way by modifying code. However standard distributions of T may be easily defined through the namelist variables **TNUDLAT**, **TNUDTOP**, and **TNUDCENT**, which define relaxation time scales at the lateral boundary, top boundary, and center regions, respectively, of the model domain. The influence of **TNUDLAT** extends inward from the lateral boundary of the model domain (coarse grid) by a number of grid cells specified by the user in namelist parameter **NUDLAT**. The influence function (inverse time scale) increases outward parabolically beginning from the parabola vertex located **NUDLAT** points in from the boundary. The nudging time scale at that vertex and deeper in the interior of the grid is defined by **TNUDCENT**. Thus, **TNUDCENT** can be used to specify a lower bound on nudging strength throughout the domain. The influence of **TNUDTOP** extends downward from the model domain top to a height specified in namelist variable **ZNUDTOP**. The influence function increases linearly between these two heights, reaching the minimum value defined by **TNUDCENT** at and below **ZNUDTOP**. Note that as time scales are inversely proportional to nudging strength, **TNUDCENT** should have a larger value than **TNUDLAT** and **TNUDTOP** for nudging to strengthen toward the lateral and top boundaries. Use of **TNUDLAT** and **NUDLAT** is a replacement for the specification of the Davies lateral boundary nudging in 2c and earlier versions of ***RAMS***. Use of **TNUDTOP** and **ZNUDTOP** functions analogously to the Rayleigh friction top boundary condition (see descriptions for **DISTIM** and **NUPTS**), but can be used for observational data fields which are not horizontally homogeneous. This provides a means for damping upward-propagating gravity waves which could otherwise reflect off the top model boundary. **TNUDCENT** is a part of the data assimilation capability in ***RAMS***, and is definable in the namelist in order to conveniently activate domain-wide nudging. Users of the nudging option should experiment with all three nudging time scales to determine the values which work best for a particular application. The only constraint for numerical stability is that none of the time scales can be less than the model timestep, except that *setting **TNUDLAT**, **TNUDTOP**, and **TNUDCENT** to zero turns off the nudging option*. Experience has shown that values of **TNUDLAT** should typically be in the range of 900-1800 seconds, **TNUDCENT** = 3600 corresponds to very strong nudging and usually is better in the 7200-14400 second range. **TNUDTOP** frequently does not need to be used and should only be activated at levels well into the stratosphere. Furthermore, if **TNUDTOP** is used, special care should be taken to examine its impact on the model solution, because the impact can be quite strong.

WT_NUDGE_GRID real array	relative weights applied to the nudging weights for each grid. The main intent is to have user control over the 4DDA analysis nudging weights on individual grids. The base weights are determined from the timescale TNUDCENT, then multiplied by WT_NUDGE_GRID. A value of 2., for example, will double the nudging strength on that grid. Care must be taken that all adjustments of nudging strength fall with the numerical stability limits
WT_NUDGE_UV **WT_NUDGE_TH** **WT_NUDGE_PI** **WT_NUDGE_RT** real	relative weights applied to the nudging weights for specific variables. The main intent is to have user control over the 4DDA analysis nudging weights. The base weights are determined from the timescale TNUDCENT, then multiplied by WT_NUDGE_UV, etc. WT_NUDGE_UV is applied to the u and v components, WT_NUDGE_TH is applied to potential temperature, WT_NUDGE_PI is applied to the Exner function, and WT_NUDGE_RT is applied to total water mixing ratio. The complete computation of the nudging weight for the analysis nudging 4DDA (in the "center" of the domain) is then computed by: (WT_NUDGE_GRID(ngrid) * WT_NUDGE_UV) / TNUDCENT
NUD_COND integer	flag to activate a condensate nudging 4DDA scheme. This scheme follows the same types of procedures as the history nudging scheme (NUD_TYPE=1), but only nudges total water mixing ratio at grid points where condensate exists. This is intended primarily for assimilation purposes in operational forecast cycles.
COND_HFILE character	History files to be read for use in the condensate nudging scheme (NUD_COND=1). The character string placed in this variable must be the full name of the history header file, including the appropriate path if applicable. However, only the prefix information is used to determine the set of history files to be used. Further details on the standard syntax of the history file name are given in the description of **HFILOUT**.
TCOND_BEG **TCOND_END** real	beginning and ending model time (seconds) to start and stop the condensate nudging scheme.
WT_NUDGEC_GRID real array	relative weight of the condensate nudging scheme for each grid
T_NUDGE_RC real	timescale for the condensate nudging scheme. The complete computation of the condensate nudging weight is then computed by: (WT_NUDGEC_GRID(ngrid) * WT_NUDGE_RC) / T_NUDGE_RC

IF_ODA integer	Flag to active the ODA (observational data assimilation) feature, a generalized observational nudging scheme. The ODA scheme will examine each station, interpolate in time to each timestep (if the observations are close enough), then perform a "kriging" interpolation to produce three- dimensional value and covariance fields. The combination of the value and covariance fields will only nudge the model fields in locations where the observations are "close enough". The parameters below will control the behavior of the ODA scheme.

ODA_UPA_PREFIX **ODA_SFC_PREFIX** character	File name prefixes (including path information) for the input surface (SFC) and upper air (UPA) data for the ODA scheme. The file must be standard RALPH II format and currently must contain one data time per file.
FRQODA real	Frequency (seconds) at which to update the interpolated observational value and covariance fields with the kriging scheme. This could be done every timestep, but probably less often for efficiency. 10-20 minutes should be an adequate compromise with standard synoptic observations, but it is dependent on the specifics of the simulation and the available observations.
TODABEG **TODAEND** real	Beginning and ending model times (seconds) to start and stop the ODA scheme.
TNUDODA real	Main nudging ODA timescale (seconds). This determines, along with WT_ODA_GRID and WT_ODA_UV, etc., the strength of the nudging. For example, the complete computation of the weight for the ODA nudging for the u and v components is computed by: (WT_ODA_GRID(ngrid) * WT_ODA_UV) / TNUDODA
WT_ODA_GRID real array	Relative weights applied to the ODA nudging weights for each grid. See TNUDODA.
WT_ODA_UV **WT_ODA_TH** **WT_ODA_PI** **WT_ODA_RT** real	Relative weights applied to the ODA nudging weights for specific variables. WT_ODA_UV is applied to the u and v components, WT_ODA_TH is applied to potential temperature, WT_ODA_PI is applied to the Exner function, and WT_ODA_RT is applied to total water mixing ratio. See TNUDODA.
RODA_SFCE **RODA_SFC0** real array	Radii (meters) for the kriging scheme to control the smoothness of the analysis at the surface. SFC0 is the distance from an observation where the influence becomes zero, while SFCE is the distance where the influence drops by a factor of e^{-2}. These values are grid-dependent.
RODA_UPAE	Radii (meters) for the kriging scheme for the upper air observations to

RODA_UPA0 real array	control the smoothness of the analysis in the upper air. UPA0 is the distance from an observation where the influence becomes zero, while UPAE is the distance where the influence drops by a factor of e^{-2}. These values are grid-dependent.
RODA_HGT real array	The height (meters) above the ground used to determine the vertical structure of the kriging radii. RODA_SFC0 will apply at the ground, while RODA_UPA0 will begin at a height of RODA_HGT. The radius will be held constant at a value of RODA_UPA0 from RODA_HGT to the model top, while a linear interpolation from the ground to RODA_HGT will be performed. Typically, some height above the expected boundary layer depth would be chosen. This value is grid-dependent.

RODA_ZFACT real	Vertical exaggeration factor to control smoothness in the kriging scheme. This is somewhat related to the ratio of horizontal grid spacing and vertical grid spacing. A value of the order 100 is reasonable.
ODA_SFC_TIL **ODA_SFC_TEL** real	Time interpolation limit (TIL) and time extrapolation limit (TEL) (both seconds) for the surface data in the ODA scheme. These are used to determine the use of the observations at the time of the kriging scheme updates. At an update time, each station will be examined. A past and future observation time (with non-missing data) relative to the update time will be found. If the future minus the past time is less than the TIL, then the data values will be interpolated to the update time for use in the data analysis. If this condition is not true, then the past and future times will be checked tosee if it is less than the TEL. If, for example, the past time is less than the TEL, then that data value will be used in the analysis. For hourly surface observations, reasonable settings might be ODA_SFC_TIL=7200., and ODA_SFC_TEL=900.
ODA_UPA_TIL **ODA_UPA_TEL** real	Time interpolation limit (TIL) and time extrapolation limit (TEL) (both seconds) for the upper air data in the ODA scheme. See ODA_SFC_TIL. For standard 12 hourly surface observations, reasonable settings might be ODA_UPA_TIL=43200. and ODA_UPA_TEL=3600.
IF_CUINV integer	Flag to activate the reading and nudging toward heating and moistening rates produced by the cumulus inversion scheme. This is a new experimental feature where the convective rates are produced before the actual run by a separate process that reads observed precipitation rates (from observations, satellite, or radar) and produces the convective tendencies. Files for each time corresponding to the availability of the precipitation rate data are produced. The rates are further weighted with a typical nudging manner to allow for control over the strength.
CU_PREFIX character	File name prefix (including path information) of the files produced from the cumulus inversion program.
TNUDCU real	Main nudging CU timescale (seconds). This determines, along with WT_CU_GRID, the strength of the nudging. The complete computation of the weight for the CU nudging is computed by: WT_CU_GRID(ngrid) / TNUDCU
WT_CU_GRID real	Relative weights applied to the CU nudging weights for each grid. See TNUDCU.
TCU_BEG **TCU_END** real	Beginning and ending model times (seconds) to start and stop the CU nudging scheme.
CU_TEL **CU_TIL** real	Time interpolation limit (TIL) and time extrapolation limit (TEL) (both seconds) for the CU scheme. See ODA_SFC_TIL. Reasonable settings will be dependent on the time resolution of the original precipitation rate data.

TIMSTR real	Time in seconds, minutes, hours, or days (see **TIMEUNIT**) already accumulated in a simulation when a history restart of the model is to commence. It must be equal to the cumulative simulation time at which the history file to be used to initialize the current run was written. The time interval obtained by subtracting **TIMSTR** from **TIMMAX** is the actual simulation time that will elapse during the current run. **TIMSTR** is relevant only if **RUNTYPE** is set to 'HISTORY'. If instead a history restart is not being done, **TIMSTR** is ignored, and the model time counter is initialized at time 0.
HFILIN character	History file to be read for initializing the current run (INITIAL='HISTORY',). The character string placed in this variable must be the full name of the history header file, including the appropriate path if applicable. Further details on the standard syntax of the history file name are given in the description of **HFILOUT**.
IPASTIN	Flag to activate a "recycle" feature, where certain fields from a previous analysis file may be read upon initialization of the current run. Currently, only the LEAF variables are included in this process. All soil and vegetation variables are read and any default initialization values are replaced. The grid structure of the previous run must exactly match the grid structure of the current run.
PASTFN	Analysis header file name, if IPASTIN=1, from which to read the recycle fields. For flexibility, the file time does not necessarily have to match the initialization time of the current run.

IOUTPUT integer	Flag controlling whether the model will output any history and analysis files. If set to 0, neither type of file will be generated. Setting **IOUTPUT** to 1 causes the model to generate both types of output files, with both the history and analysis files written in ASCII. Setting **IOUTPUT** to 2 causes the model to generate both types of output files, with the history file written in binary and the analysis file in ASCII. The purpose of a history file is solely to store prognostic model variables and some additional information generated in a model run so that a subsequent model run can be initialized from the file to continue a simulation. The purpose of an analysis file is to store model output to be used in producing graphic visualization of the model output. In order for the generation of visual products to be performed on a machine other than the one used for executing the model (this is useful, for example, when the model is run on a supercomputer and the graphical output is produced on a workstation), the analysis file currently is written in a packed ASCII format for easy transfer between machines. Full precision of the model variables is not required in generating visualization products, so only 2 or 3 bytes of information are written to the analysis file for each model word in order to minimize file storage space. Each of these 8-bit bytes contains 6 bits of actual information from the model variables, a ratio achieved by translation of model numerical values into a set of 64 different ASCII characters. If set to 5 will generate a special variable-resized output in order to be used

	within INITIAL=4 feature. This output contains only U, V, PI, THETA and RH variables by default, but if chemistry is active, it will add all concentrations species of a given chemical mechanism to create a complete initial/boundary condition file. FRQANL controls the writing frequency for those files.

IPOS Integer Range : 0,1,2	Flag to provides GrADS output during the model time run. See "POST" namelist for details. 0-no files, 2-grads files, 1-hdf5(not available yet)
HFILOUT character	Prefix for the history files output from the model. Only a single character value is assigned to **HFILOUT** even if more than one history file is to be written in the current run. **HFILOUT** is used as a prefix to the one or more file names, making up the first part of each name, including directory path information. The model itself appends the last part of each history file name, according to the time during the simulation when the file is written and the grid number. That is, a separate history file is written for each grid number, as well as for each history write time. In addition, a single history *header* file is written at each history write time. As the model appends both the time and the grid number onto the history file prefix (given by **HFILOUT**), each file name written during a simulation has a distinct name, but all still share the common prefix. It is useful to include the letter 'h' in the character string specified for this variable to help identify the output file as a history file. **HFILOUT** is only relevant if **IOUTPUT** is set to 1, indicating that history files are to be created.
AFILOUT character	Similar to **HFILOUT**, but applying to the analysis files, rather than the history files, to be output from a model run. It is useful to include the letter 'a' in the character string specified for this variable to help identify the output file as an analysis file.
ICLOBBER integer	A flag indicating what the model should do if history and/or analysis files already exist from a prior run, and the model is about to write new files of the same name in the same directory. If **ICLOBBER** is set to 0, the model will stop and not overwrite the older files. If **ICLOBBER** is set to 1, the model will continue and will write new files, overwriting the old ones.
IHISTDEL integer	A flag indicating what the model should do with history files from the current model simulation or run that were written at previous times, once the latest history file has been successfully written. If **IHISTDEL** is set to 0, the previous history files are retained. If **IHISTDEL** is set to 1, the previous history files are deleted and only the latest one is retained. This latter option saves disk space while keeping the latest history file to restart from in case the model stops for some reason.

FRQHIS real	Time interval in seconds at which successive history files are to be output from the model. The model variable TIME begins at zero at the start of a new simulation (and at the value of **TIMSTR** for a history file start) and counts up to the value of **TIMMAX** for each run. (see **RUNTYPE**, **TIMSTR**, and **TIMMAX**). When TIME reaches a multiple of **FRQHIS**, a model history file is written. In addition, a history file is written out when a model run terminates, even if **TIMMAX** is not set to an exact integer multiple of **FRQHIS**. A history file is also written out when a model run begins, even if the TIME variable is not then a multiple of **FRQHIS**. In this latter case, if the model run is a history start, the initial history file will be written at the same time as the history file that is read to initialize the run. Because this could lead to a file name conflict (the files corresponding to the same simulation time), the upper-case letter 'R' is appended to the initial output history file following the **HFUNITS** suffix to denote a 'Restart'.

FRQANL real	Similar to **FRQHIS**, but specifies the time interval between successive writes of analysis files, rather than history files, from the model. Analysis files (and history files) may only be output at the end of a coarse grid timestep (whose length is specified by **DTLONG**, but see **IDELTAT** also). Thus, unless **FRQANL** is an exact multiple of **DTLONG**, the actual time interval between analysis file write times will be irregular. Thus, the user should usually make sure that **FRQANL** is an exact multiple of **DTLONG**. If **IDELTAT** is not zero, in which case the model will choose the timestep, and model will take the value of **FRQANL** into account to make sure that analysis files are written at regular intervals. There now exist 4 different types of standard analysis files that ***RAMS*** can output: (1) the conventional analysis file whose frequency is controlled by **FRQANL**, (2) a 'lite' analysis file which is intended for a smaller set of output data so that it can be written more frequently, (3) a 'mean' analysis file that contains time-averaged fields, and (4) a 'both' analysis file that combines lite and time-averaged properties. **FRQLITE**, **FRQMEAN**, and **FRQBOTH** control the output frequencies of these latter 3 files.
FRQLITE real	Similar to **FRQHIS** and **FRQANL**, but specifies the time interval between successive writes of 'lite' analysis files. The lite files are intended for writing variables at much more frequent intervals than analysis files, for use in post-processing applications such as graphical display, but with much less information. For example, a user may need surface variables to be output frequently, but not three-dimensional upper air fields. Only the required fields would be specified for output in the lite files. Alternatively, subsets of fields, rather than the entire fields, may be output through use of namelist variables **XLITE**, **YLITE**, and **ZLITE** (see below), although this option is not yet available in ***RAMS*** version 5.0.
XLITE character	Specifies beginning and end of range of i-index values of a subset of any output field to be written to 'lite' analysis files. See **FRQLITE** above. Special syntax of the form '/0:0/' is used where the two integer values, when positive, indicate the actual starting and ending i-index values of the range of values to be output, and when negative, specify the number of grid points in from the left and right grid boundaries of the range limits. This option is not yet implemented in ***RAMS*** so only the entire field may be written to lite files.
YLITE character	Similar to **XLITE** but specifies the range of j-index values in the y-coordinate direction (usually approximately north-south). Not yet implemented so only the entire field may be written to lite files.
ZLITE character	Similar to **XLITE** but specifies the range of k-index values in the vertical coordinate direction. Not yet implemented so only the entire field may be written to lite files.
NLITE_VARS integer	Number of variables to write to the "lite" analysis file. The actual variable names are specified in the variable **LITE_VARS**.

LITE_VARS char array	Variable names of the fields that will be written to the "lite" analysis files. The variable names correspond to the RAMS' internal character ID strings, which now can be found in the modules/mem_*.f90 files. For example, the u-component is "UP", potential temperature is "THETA", etc. Any variable that is in memory may be included, but care must be taken in parallel runs to ensure that the field has been passed back to the master process.
AVGTIM real	Averaging time period for special time-averaged arrays that may be output to analysis file. Positive values for **AVGTIM** cause the file write time to be centered in time within this averaging period, while negative values cause the file write time to occur at the end of the averaging period.
FRQMEAN real	Similar to **FRQANAL**, but applies to the output frequency of 'mean' analysis files, which are one of the 4 standard types of analysis files that ***RAMS*** can write (see **FRQANL**).
FRQBOTH real	Similar to **FRQANAL**, but applies to the output frequency of 'both' analysis files, which are one of the 4 standard types of analysis files that ***RAMS*** can write (see **FRQANL**).
KWRITE integer	Flag specifying whether to write horizontal and vertical turbulent mixing coefficients for scalars to analysis files (0 = no, 1 = yes). If they are to be written, two additional arrays are allocated, the mixing coefficients are copied to these arrays when they are computed in the model, and the values are saved in these arrays for when the analysis files are written. The mixing coefficient values are sometimes needed for plotting, and they are needed for the ***HYPACT*** model, which runs from ***RAMS*** output.
FRQPRT real	Time interval in seconds at which various field values are to be output from the model to the standard output file generated by a model run. The actual fields or portions thereof to be output are specified via several variables in the **$MODEL_PRINT** namelist. The relation between **FRQPRT** and the model simulation time is the same as for **FRQHIS**.
INITFLD integer	Flag specifying whether the field values indicated in the $MODEL_PRINT namelist for output to the standard output file are to be output at the initial time of the model run. A value of 0 will cause the values to not be output at the initial time, while a value of 1 will cause them to be output at the initial time.
SFCFILES character	Filename prefix (including path information) for the surface files used in a model simulation. These files contain topography, soil textural class, vegetation type, and subgrid distribution of soil textural class, vegetation type, and water surface areas defined on each model grid. Surface files are always used in a model run. They are commonly made in a separate model run that does not proceed with prognosing atmospheric fields by setting **RUNTYPE** to 'MAKELAND' or 'MAKESFC'. However, if **RUNTYPE** is instead set to 'MAKEVFILE' or 'INITIAL', and if surface files do not exist or are found to be inconsistent with model grid size, location, etc., new ones will be generated. If they already exist from a prior run, surface data is read from them. A separate surface file is generated for each grid, and the grid number is appended to each filename in the format of SFCFILES.g01, SFCFILES.g02, etc.

SSTFPFX character	Filename prefix (including path information) for SST files used in a model simulation. These files contain sea surface temperature data defined on each model grid, and usually for multiple times. This data type is available on standard ***RAMS*** data files defined globally on a latitude-longitude grid with climatological values for each month of the year. Data in surface files may be interpolated from these standard files, from other data sources, or may be defined by other means. These choices are controlled by **ISSTFLG**. SST files are always used in a model run. They are commonly made in a separate model run that does not proceed with prognosing atmospheric fields by setting **RUNTYPE** to 'MAKESST' or 'MAKESFC'. If **RUNTYPE** is instead set to 'MAKEVFILE' or 'INITIAL', and if SST files do not exist or are found to be inconsistent with model grid size, location, etc., new ones will be generated. If they already exist from a prior run, surface data is read from them. A separate SST file is generated for each grid and for each time available in the files from which the data are interpolated.

ITOPTFLG integer array	Grid-dependent flag specifying how terrain height data is to be obtained or computed for each model grid. If **ITOPTFLG** is set to 0 (allowed for a nested grid only), topography for that grid will be interpolated from its parent grid. This will result in relatively smooth topography on the nested grid, but is required if the nested grid is required to move in time (see **GRIDU** and **GRIDV**). If **ITOPTFLG** is set to 1, topography will be interpolated to the grid from a standard ***RAMS*** topography dataset, currently available at 30 arc-second intervals of latitude and longitude for all land areas of the globe. A prefix for all filenames in this dataset is specified in **ITOPTFN**. This is by far the most common way to initialize topography in ***RAMS*** unless the experiment requires idealized or flat topography. If **ITOPTFLG** is set to 2 for a grid, the model will call subroutine TOPTINIT in the model file *rsurf.f90* to assign the terrain height, which by default sets it to zero. To override any of these choices, subroutine TOPTINIT_USER in the file *rsurf.f90*, which is always called but contains only commented out executable lines in its standard form, may be modified. The ruser.f90 file is specifically intended for user modification when such a need arises. Subroutine TOPTINIT_USER is normally constructed with an IF block checking for the grid number (NGRID). The terrain height values must be assigned for the appropriate grid number. It must be pointed out that if nested grids are used, it is required that the terrain heights be compatible between different grids. The compatibility requirement is two-sided: Fine grid values corresponding to a single coarse grid cell must locally average to the coarse grid terrain height value in that cell, and the fine grid values near the lateral boundary of the fine grid must be those which would result from a bi-quadratic interpolation from coarse grid values, according to the standard nesting procedure. The model is designed to make the terrain heights compatible automatically, and will alter the values assigned to a grid in subroutine TOPTINIT or TOPTINIT_USER if they are not already compatible between grids. A special word of caution is in order here. If a fine grid that is to contain fine resolution terrain height data is to be spawned on a history start at a time after the initialization time of a simulation, the terrain height data on the coarser grids must already have been compatible with it in the preceding portion of the simulation. This may be done either by simply initializing the topography of the spawned grid from its parent, or by generating surface files for all grids, including the one to be spawned, before the simulation started. The latter choice provides for the better-resolved topography on the spawned grid, but will also result in very fine (at the 2-delta-x scale) features on its parent grid. This may cause some model noise problems during that earlier part of the simulation on the portion of the coarser grids where the fine grid will be later inserted, but this is normally at acceptable levels (see **TOPTWVL**).

ISSTFLG integer array	Grid-dependent flag specifying how the sea surface temperature variable is to be acquired or computed for a grid. Values of 0, 1, or 2 are applicable for this flag, and have the same meaning as for **ITOPTFLG** and **IPCTLFLG** described above. In contrast to the terrain height data controlled by **ITOPTFLG**, however, the strict compatibility between coarse and fine grid values of sea surface temperature is not required, although it would be rather senseless for coarse and fine grid values to be grossly different. Sea surface temperature is a variable defined for each surface grid cell in the model, and indicates the temperature of any surface water in that cell. For **ISSTFLG** set to 2, the model calls subroutine SSTINIT in model file rsurf.f90 that by default sets the sea surface temperature to the value of namelist variable **SEATMP**. If for a fine grid **ISSTFLG** is set to 0, the sea surface temperature values for that grid will by default be interpolated from its parent grid. For **ISSTFLG** set to 1, values for sea surface temperature are interpolated from standard ***RAMS*** SST data files. This latter option is used in conjunction with namelist variables **ISSTFN**. To override any of the above options, subroutine SFCINIT_USER in the file ruser.f90 is called, and there a user may customize the sea surface temperature field as described above for modifying TOPT in subroutine TOPTININT_USER.
IVEGTFLG integer array	Grid-dependent flag specifying how the vegetation type variable is to be acquired or defined for a grid. Values of 0, 1, or 2 are applicable for this flag. If **IVEGTFLG** is set to 0 (permitted only for a nested grid), values of vegetation type are directly assigned from the local grid cell of the parent grid. This applies to each individual subgrid patch as well (see description of LEAF2 in Section 1c), If **IVEGTFLG** is set to 1, vegetation class data are obtained from a ***RAMS***-specific database which is defined globally at 30-arc-second intervals of latitude and longitude, and whose filename prefix is specified in namelist variable **IVEGTFN**. When this option is used, and when **IVEGPAT** is greater than 1, multiple subgrid land patches are filled from the ***RAMS*** database according to their relative prevalence in the grid cell. If **IVEGTFLG** is set to 2, vegetation class values are defined in subroutine SFCINIT in the rsurf.f90 file. In this latter case, the value specified in namelist variable **NVGCON** is used by default for the entire domain. To override any of the above options, subroutine SFCINIT_USER in the file ruser.f90 is called, and there a user may customize the vegetation type field as described above for modifying TOPT in subroutine TOPTINIT_USER.
ISOILFLG integer array	Flag specifying how soil textural class is to be initialized in ***RAMS***. If **ISOILFLG =** 0 (only allowed for nested grids), soil class is specified from the local value on its parent grid. If **ISOILFLG** = 1, soil textural class is read from standard ***RAMS*** data files. If **ISOILFLG =** 2, soil textural class for the grid is filled horizontally homogeneously to the value specified by **NSLCON**. As for vegetation class values described for **IVEGTFLG**, soil textural class values may be customized in subroutine SFCINIT_USER in the file ruser.f90.

NDVIFLG integer array	Flag specifying how NDVI (Normalized Difference Vegetation Index) is to be initialized in ***RAMS***. If **NDVIFLG =** 0 (only allowed for nested grids), NDVI is specified from the local value on its parent grid. If **NDVIFLG** = 1, NDVI textural class is read from standard ***RAMS*** data files. If **NDVIFLG =** 2, NDVI for the grid is filled horizontally homogeneously to the value specified in leaf3_init.f90.As for vegetation class values described for **IVEGTFLG**, NDVI values may be customized in subroutine SFCINIT_USER in the file ruser.f90.
NOFILFLG integer array	Grid-dependent flag similar to **ITOPTFLG**, **ISSTFLG**, **IVEGTFLG**, and **ISOILFKG**, but applies to variables in LEAF2 (the soil, snowcover, vegetation, and canopy air submodel of ***RAMS***) that are not available on standard ***RAMS*** data files. These variables are: soil temperature, soil moisture content, snowcover (or temporary surface water) amount, snowcover temperature, snowcover depth, vegetation temperature, vegetation surface water, canopy air temperature, and canopy air vapor mixing ratio. Setting **NOFILEFLG** to 0 (allowable only for nested grids) causes initial values of these quantities to be copied from local values on the parent grid, while setting **NOFILFLG** to 2 causes the values to be initialized in a default manner. For default initialization of soil moisture and temperature, see **SLMSTR** and **STGOFF**.Default initial values of vegetation and canopy air temperatures, and canopy air vapor mixing ratio, are the values for the lowest model atmospheric layer. Snowcover and vegetation surface moisture are initialized to zero by default, although snowcover (ground surface water) is set to a positive value in swamps, bogs, and marshes where such land surface types are read from the standard ***RAMS*** files (see **IVEGTFLG**). Note that any of these initial values may be overridden by modifying subroutine SFCINIT_USER in the file ruser.f90. This subroutine is often used, for example, to insert soil moisture and snowcover observational data.
IUPDNDVI integer	Flag for specifying whether observed NDVI values are to be held constant in time during a model run (**IUPDNDVI** = 0) or are to be continually linearly interpolated in time between the observing times immediately before and after the current model runtime (**IUPDNDVI** = 1). Important: it is strongly recommended the setting **IUPDNDVI** = 1.
IUPDSST integer	Flag for specifying whether observed sea surface temperatures are to be held constant in time during a model run (**IUPDSST** = 0) or are to be continually linearly interpolated in time between the observing times immediately before and after the current model runtime (**IUPDSST** = 1). Important: it is strongly recommended the setting **IUPDSST** = 1.

ITOPTFN character array	Grid-dependent character variable specifying the name of the terrain height data *header* file, with path names if applicable, to be read in conjunction with namelist variable **ITOPTFLG** set to 1. These data files are named according to a convention that identifies the latitude and longitude of their southwest corners. The standard files prepared for ***RAMS*** each comprise an area of 5 degrees of longitude by 5 degrees of latitude with values defined at 30-arc-second intervals of latitude and longitude, and cover all land areas of the earth. These files have names such as EL35N115W, indicating a southwest corner at 35 degrees north latitude and 115 degrees west longitude, with the characters 'EL' serving as a filename prefix for all files in the dataset. The value specified for **ITOPTFN** would be 'EL' in this case, as the model supplies the remainder of the file name according to the file(s) it needs to read. These terrain height files are written in the same high-information-density ASCII format as the analysis files (see **IOUTPUT**). Sometimes in model applications, fine grids are located within the bounds of a special limited-area terrain height dataset, such as the one described above, and can obtain their terrain height information from that dataset, while coarser grids extend outside the high resolution dataset, and utilize a coarser-resolution dataset of global coverage, not needing the high resolution data. This is why **ITOPTFN** is made grid-dependent.
ISSTFN character array	Grid-dependent character variable indicating the filename prefix (with path names if applicable) of the header file for the sea surface temperature data for initializing each grid. It is similar to namelist variable **ITOPTFN**.
IVEGTFN character array	Grid-dependent character variable indicating the name (with path names if applicable) of the header file for the vegetation type data for initializing each grid. It is similar to namelist variable **ITOPTFN**.
ISOILFN character array	Grid-dependent character variable indicating the name (with path names if applicable) of the header file for the soil textural class data for initializing each grid. It is similar to namelist variable **ITOPTFN**..
NDVIFN character array	Grid-dependent character variable indicating the name (with path names if applicable) of the header file for the NDVI data for initializing each grid. It is similar to namelist variable **ITOPTFN**.

ITOPSFLG real array	Flag that controls the type of processing of topographic data from input files that are specified in **ITOPTFN** to final values defined on a ***RAMS*** grid. This is a 3-step process that involves topography information being defined successively on 4 different grids, which we call the 'O', 'P', 'Q', and 'R' grids. First, a horizontal interpolation is carried out in order to transfer data from the 'observed' or O grid of the input file, to a polar stereographic grid of comparable resolution, which is our P grid. The P grid uses the same projection as the ***RAMS*** grid (the R grid) where the data will reside in its final state, but is usually of much higher resolution. Second, data is averaged from this P grid to a lower-resolution Q grid, which is also polar stereographic and has a horizontal grid spacing that is an integer multiple of that on the P grid. This step automatically filters out small scale variations which are not desired on the model grid (see **TOPENH**). In this second averaging step, a choice of averaging algorithms exists and **ITOPSFLG** is the flag that selects the choice to be used. If **ITOPSFLG** is set to 1, a conventional mean is used where terrain heights for all P grid cells in a single Q grid cell are summed and divided by that number of P values, to obtain the value for that Q cell. For **ITOPSFLG =** 1, both the conventional mean and a silhouette average are computed, and the value assigned to the Q grid cell is a weighted average of these, with the weights controlled by **TOPENH**. The silhouette average finds the mean height of the silhouette, as viewed from the east or west, of the set of P grid terrain heights contained within a single Q grid cell, and a separate silhouette height as viewed from the north or south and averages the two silhouette heights together. This becomes the computed silhouette height for that coarse-grid cell. While the conventional average preserves total terrain volume above sea level, the silhouette average adds mass by filling in valleys. It is used to maintain the effective mean barrier height that air must rise to when crossing a topographic barrier such as a ridge. The conventional average lowers this barrier height, particularly when the barrier height is poorly resolved. When **ITOPSFLG =** 2, an envelope topography scheme is used to obtain Q grid values from P grid values, and this scheme is an alternative method of attempting to preserve barrier heights. When **ITOPSFLG =** 3, a reflected envelope topography scheme is used which aims to preserve both barrier heights and valley depths. Naturally, this method leads to the steepest topography in ***RAMS***, while still filtering the shortest wavelengths. In the third and final step, topography is interpolated from the Q grid to the R grid, and the R grid is usually of moderately higher resolution than the Q grid (see **TOPTWVL**).
TOPENH real array	Grid-dependent variable specifying the weight given to the *silhouette* average of terrain height in topographic initialization when **ITOPSFLG =** 1 (see above), or an envelope orography enhancement factor when **ITOPSFLG =** 2 or 3. (Also see **TOPTWVL**.)

TOPTWVL real array	Grid-dependent variable specifying the wavelength, in grid-cell size units, of the smallest horizontal modes of terrain height data which are to be present on a given model grid. It is applicable only for namelist variable **ITOPTFLG** set to 1. Referring to the description of namelist variable **ITOPSFLG** above, the value of **TOPTWVL** controls the ratio of resolution between the Q and R grids. The shortest mode which any grid can resolve is that with a wavelength of twice the grid cell size. In general, the Q grid will contain all wavelengths of topographic data from its own 2 deltax scale and larger. Hence, if the ***RAMS*** 'R' grid, to which data is interpolated from the Q grid, has, for example, half the cell size of the model grid, the smallest mode that it will receive from the Q grid will be 4 deltax on the R grid. This smallest mode, in deltax units of the R grid, is the number specified for **TOPTWVL**. In other words, while the R grid spacing is set by the user (see **DELTAX**), the Q grid spacing will be **TOPTWVL** / 2 times the R grid spacing. This is how smoothing of topographic data is achieved in ***RAMS*** while allowing the variety of enhancing schemes described for ITOPSFLG. Because the numerical model does not properly handle the smallest modes resolvable on a grid, it is generally important not to force these modes into the meteorological fields through overly fine terrain height modes.
IZ0FLG integer array	A flag that controls how surface roughness length is computed for computing surface vertical fluxes of momentum, sensible heat, and latent heat. If **IZ0FLG** is set to 0, the standard roughness height evaluation of soil, snowcover, and/or vegetation surfaces is computed in the LEAF-2 submodel of ***RAMS***. If **IZ0FLG** is set to 1, roughness height is computed from the roughness of subgrid scale topography, and may reach several meters. An upper limit to roughness height may be imposed by the user (see **Z0MAX**). The influence of rough topography on vertical turbulent transport is most properly applied through several (lower) model layers, as well as the surface itself, but in the current implementation, its effect is concentrated solely on the surface fluxes. This tends to overestimate fluxes from the soil and vegetation to the lowest atmospheric level, and this option in ***RAMS*** should be used with caution (also see **Z0FACT**).
Z0MAX real array	Upper bound (in meters) imposed on surface roughness height that is evaluated from subgrid topography (see **IZ0FLG** and **Z0MAX**).
Z0FACT real array	Roughness factor used in computing surface roughness height from subgrid topography (see **IZ0FLG** and **Z0MAX**).

MKCOLTAB integer	Flag indicating whether or not a *collection table* is to be created in the present run. The collection table is required in the bulk microphysics package, and contains terms derived from the stochastic collection equation. These terms are computationally intensive to evaluate, and are thus pre-computed and stored in the collection table where they are efficiently accessed during model runtime. The values in the table depend on two sets of parameters. One is the set of gamma distribution shape parameters, defined in namelist variable GNU. The other is a list of parameters defined in a data statement in subroutine MICINIT in the file *mic_init.f90*, which describe the relation between mass and diameter, and between fall velocity and diameter for the hydrometeor categories. These latter parameters are not normally altered, unless a user has reason to change these relationships. The gamma distribution shape parameters defined in the namelist, however, are intended to be adjusted and experimented with. Any change in any parameter of either set requires a different collection table to be created and used. Setting **MKCOLTAB** to 1 causes the table to be created at the beginning of the present run, and to be stored in a file of the name given in namelist variable **COLTABFN**. The relevant parameters from both the namelist (the GNU variable) and the data statement in the code) are written in the first lines of the collection table. If **MKCOLTAB** is set to 0, the model will not create the table, but will instead attempt to read the file specified in the variable **COLTABFN**. This option avoids the need to generate the collection table each time a run is made. When the collection table is read in, the list of parameters in its first few lines will override the parameter settings in the namelist and the data statement. Only relevant for MCPHYS_TYPE = 0
COLTABFN character	Is a single-valued character variable giving the name (and path, if any) of the *collection table*, described above. Important: For MCPHYS_TYPE = 0 the name of the file is “ct2.0” For MCPHYS_TYPE = 1 the name of the file is “col_table”

MAPAOTFILE Character	Filename to read aerosol optical properties map over South America (infMapAOT.vfm). Only relevant for the CARMA radiation scheme.

$MODEL_OPTIONS Namelist

The **$MODEL_OPTIONS** namelist is where the majority of choices for specifying model parameterization options are made. The variables in this namelist are those which do not fit under the special categories of the **$MODEL_GRIDS** and **$MODEL_FILE_INFO** namelists, *i.e.*, those which neither control grid configuration, space and time domains, nor the reading or writing of data files.

ADVMNT Integer Range : 0,1,2	Flag that controls which advection scheme for scalars will be applied: 0 = Forward 2nd order (non-monotonic) 1 = Walcek monotonic advection scheme 2 = hybrid: forward 2nd order for thetail, microphysics species and TKE and Walcek scheme for tracers. Option 1 is strongly recommended for chemistry and explicit cloud simulations
GHOSTZONE LENGTH Integer Range: 1 to max(nxp,nyp)	Size of the ghostzone for the Walcek advection scheme. The recommended size is "3".

NADDSC integer array	Grid dependent parameter specifying the number of prognostic scalar fields to be added to the model simulation. Such scalar fields might be desired by the user for representing quantities such as chemical or aerosol pollutants. The added scalar fields are automatically advected, diffused, and marched forward in time, simply by setting **NADDSC** to a positive value. The new variable names that are added to model are SCLP(1), SCLP(2), etc. for the "past" time level and SCLT(1), SCLT(2), etc. for the tendency arrays. These, therefore, are referenced in the "A" array as A(ISCLP(1)), A(ISCLP(2)), etc. It is up to the user to add routines in the code to compute any initialization, source or sink terms required for the added scalars. The number of scalars that can be added in this way is unlimited (provided computer memory is large enough), but the value of the parameter **MAXSCLR** must be greater than or equal to **NADDSC**.
ICORFLG integer	Flag specifying whether the Coriolis force is to be activated, and, if the simulation is 2-D, whether the v-component of velocity is to be activated (0 = no, 1 = yes). In a 2-D simulation where only the u and w velocity components are in the grid plane, since the v-component would be uncoupled from u and w without Coriolis force, a value of 0 for **ICORFLG** deactivates prognosis or use of the v-component. It should be noted that in a 2-D simulation, which assumes that prognostic fields are homogeneous in the y-direction, many factors that control the v-component of wind are excluded from the equation set, and v can become unrealistic, especially where significant terrain exists in the model domain. Thus, 2d simulations should usually exclude consideration of the v-component altogether and use **ICORFLG =** 0. If the simulation is horizontally homogeneous (namelist variable **INITIAL**, set to 1), no horizontal pressure gradients are initially available to balance the Coriolis force, and it is useful to compute the Coriolis force only from the difference between the current model winds and a reference state wind (normally the initial model wind interpolated from the sounding). Then, the true horizontal distribution of pressure would be regarded as the sum of the prognosed model pressure and a gradient in geostrophic balance with the reference state wind.

VVELDAMP Integer Range: 0,1	Flag that controls if the damping scheme for vertical velocity will be applied or not: 0 = OFF 1 = ON Option 0 is strongly recommended. Option 1 is useful only for operational purposes.
IEXEV Integer Range: 1,2	Flag to determine the type of the Exner function tendency equation to be solved: 1 = simplest form. 2 = complete, mass conservative, solution based on Medvigy et al. (2005) - works only with sigma-z coordinate. The recommended option is "2".
IMASSFLX Integer Range: 0,1	Flag to determine if model will output advective and convective mass fluxes for the STILT Lagrangian Particle Dispersion Modelling. This option only works with sigma-z coordinate 0 = OFF 1 = ON

IBND integer	Flag controlling the lateral boundary conditions (in the x-direction) that are applied on only the coarse grid of a simulation. When **IBND =** 1, this condition applies only to the normal velocity component (i.e., U at the east and west boundaries), and activates the Klemp-Wilhelmson condition in which the normal velocity component specified at the lateral boundary is effectively advected from the interior assuming a propagation speed (intended to be similar to a dominant gravity wave phase speed). This phase speed is specified in the namelist in the variable **CPHAS**. This boundary condition is intended to allow most disturbances to propagate out of the model domain without strongly reflecting back into the interior. Setting **IBND** to 4 sets up a cyclic boundary condition between the two x-direction boundaries wherein all prognostic variables in the model, not just the normal velocity component (u), are assigned at the boundaries from values taken from the model interior just inside the opposite boundary. This makes the model periodic in the x-direction, effectively eliminating lateral boundaries from the solution. The number of grid points in the x-direction comprising one complete cycle in the periodic field is 3 less than **NNXP** for the coarse grid, if the second-order advection scheme is used (see descriptions for **IADVL** and **IADVF** in the ENGPARMS Section). For the 4th- and 6th-order advection schemes, the periodic cycle length is 5 and 7 points, respectively, less than **NNXP** for the coarse grid.
JBND integer	Similar to **IBND**, but applies to the y-direction, rather than x-direction boundaries. In a 2-D simulation, **JBND** is irrelevant.
CPHAS real	Used in conjunction with **IBND** and/or **JBND** set to 1, and specifies the characteristic propagation speed of internal gravity waves used in the Klemp-Wilhelmson lateral boundary condition. A value of 20 m/s is a common setting for this parameter. Use of larger values, even very much larger, still allows propagation of gravity waves out of the domain reasonably well, whereas much smaller values tend to lead to stronger reflection of the waves back into the domain interior. A very large value of **CPHAS** is equivalent to the zero-gradient boundary condition.

LSFLG integer	Complements **IBND** and **JBND** as a lateral boundary condition flag, applying to all variables other than the normal velocity component. It deals with all these remaining variables in the same way. **LSFLG** applies only for **IBND** = 1 and for **JBND** = 1. A value of 0 for **LSFLG** sets the lateral boundary value of each variable to the value in the field immediately adjacent to the boundary in the interior. This is a zero-gradient condition. This option is the most commonly used for horizontally-homogeneous simulations (the **INITIAL** variable, set to 1), and is recommended as a first-try option. Occasionally at inflow boundaries, this zero-gradient option has been found to cause a runaway trend to a field variable, and must then be replaced by another option. A value of 1 for **LSFLG** is very similar to the 0 value, using a zero-gradient condition at inflow boundaries, but setting values at outflow boundaries such that the second spatial derivative in the direction normal to the boundary is zero. This is simply an extrapolation from two interior points. This condition is superior for some cases to the simpler zero-gradient option at the outflow boundary, but usually makes no significant difference. A value of 2 for **LSFLG** is identical with the value of 1 for the outflow boundary condition, but holds variables constant in time at inflow boundaries. This is a possible remedy to the runaway problem sometimes encountered at inflow boundaries, as described above, but is not without its own shortcomings. In cases where flow reverses in time at a lateral boundary, alternating between inflow and outflow, option 2 for **LSFLG** will cause the boundary condition to alternate between constant in time and time-variant. This may generate an undesirable model response in some cases. Another problem with holding the variables constant in time at an inflow boundary is that those constant values are continuously advected into the model interior, and may become incompatible with the evolving situation in the domain. For example, if a boundary layer is growing in the model in response to solar heating, the constant-in-time inflow condition will cause the initial cooler values of temperature to flow into the model from the boundary, artificially cooling the boundary layer. A good solution for this situation is to directly specify an appropriate temporal evolution of the boundary value, still using the **LSFLG =** 2 option. This is not presently a standard feature in the code, but can easily be implemented by the user in subroutine LATSET, located in the file *rbnd.f90*. A value of 3 for **LSFLG** leaves values at all boundaries, inflow and outflow, unchanged in time. This option can be used only for variable initialization simulations (the **INITIAL** variable, set to 2), and has sometimes, but not always, been found to give better results than **LSFLG** = 0 for **INITIAL** = 2.

NFPT integer	Number of grid levels on the coarse grid, starting from the top of the model domain, in which the Rayleigh friction absorbing layer will be activated (also see **DISTIM**). A value of 0 leaves Rayleigh friction inactive. Rayleigh friction must be applied only in stable air, and thus requires the model domain to extend sufficiently high, most commonly in the stratosphere. The bottom layer of the Rayleigh friction zone should be above the top of whatever meteorological system is important to the simulation, so as not to directly interfere with its behavior. This normally requires adding more model levels in the vertical than would otherwise be used in a simulation not employing Rayleigh friction. Rayleigh friction is a relaxation of all 3 velocity components and potential temperature toward the undisturbed, horizontally homogeneous reference state values. Its purpose is to damp gravity wave and other disturbances which approach the top model boundary, so that they will not be reflected back downward. Because prognostic fields which are not initialized horizontally homogeneously cannot sensibly be forced toward a horizontally-homogeneous state, Rayleigh friction must not be used with namelist variable **INITIAL** set to 2. However, top boundary nudging (which damps gravity waves) for horizontally-inhomogeneous initialization may be activated elsewhere (see **TNUDTOP** and **ZNUDTOP**.)
DISTIM real	Used in conjunction with namelist variable **NFPT** in activating Rayleigh friction in the upper region of the model. **DISTIM** specifies a dissipation time scale in seconds, representing the e-folding decay time of a disturbance being damped by the Rayleigh friction layer in the absence of any other source or sink terms. This decay rate strictly applies at the top boundary of the model only, and the strength of the damping decreases linearly with height, reaching zero at the lowest of the **NFPT** levels in which the damping is activated. **DISTIM** should be related to the rate at which the model is generating upward propagating gravity waves, and normally ranges from 60 to 200 seconds. **DISTIM** must be at least twice the value of **DTLONG** for numerical stability. The model solution proves to be rather insensitive to **DISTIM**, provided it is in or near this range.

ISWRTYP Integer	Flag specifying options for evaluating shortwave radiative transfer in the model 0 = no radiation, 1 = Chen and Cotton parameterization, 2 = Mahrer and Pielke parameterization, 3 = Two-stream parameterization developed by Harrington, 4 = Radiation scheme from the CARMA aerosol model, 5 = reserved, 6 = Rapid Radiative Transfer Model (RRTMG). The Mahrer and Pielke scheme is the simplest and by far the least expensive computationally. The main reason for this is that it ignores liquid and ice in the atmosphere, although it does account for water vapor. This scheme should not be used, for example, if attenuation of solar radiation by clouds is important to the simulation. The Chen and Cotton scheme does account or condensate in the atmosphere, but not whether it is cloud water, rain, or ice. This is a major limitation. The Harrington parameterization accounts for each form of condensate (cloud water, rain, pristine ice, snow, aggregates, graupel, and hail) as well as water vapor, and even utilizes information on ice crystal habit. In addition, the scheme adds upper atmospheric levels for radiation computation for cases where the model domain does not extend up to at least 25 km (roughly the height of the ionosphere). This is the most sophisticated parameterization and is recommended over the Chen and Cotton scheme. The radiation option 4 is based on the scheme of a modified version of the Community Aerosol and Radiation Model for Atmosphere (CARMA, Toon et al., 1988). The original CARMA version considered simultaneously an aerosol microphysics scheme and two-stream radiative transfer module for both solar and terrestrial spectral regions (Toon et al., 1989). The major modification from the original version refers to the prescription of aerosol intensive optical properties, specifically extinction efficiency, single scattering albedo and asymmetry parameter. These parameters are obtained from previous off-line Mie calculations. Especially for the South American continent, the prescription of smoke optical properties derives from the use of climatological size distribution and complex refractive index from several AERONET sites measurements in the southern of the Amazon Basin. These properties are used as input to an off-line Mie code to calculate the spectral optical properties required by CARMA (Procopio et al., 2003; Rosario et al., 2011). The spatial distribution of the optical models in the model is based on the concept of anisotropic areas of influence described by Hoelzemann et al. (2009). Elsewhere in the model domain not covered by the areas of influence, a reference optical model was adopted. The option 6 implements the RRTMG radiation scheme. It is a radiative transfer model that utilizes the correlated-k approach to calculate fluxes and heating rates efficiently and accurately for application to GCMs. See http://rtweb.aer.com/rrtm_frame.html for details. It is strongly recommended to apply options 4 or 6.
ILWRTYP integer	Similar to **ISWRTYP**, but applies to longwave, rather than shortwave, radiation. The same values for **ILWRTYP** apply with the same meaning as for **ISWRTYP**.
RADDATFN Character	Radiation parameterization data file to be read only if CARMA radiation is defined (**ISWRTYP**/ **ILWRTYP** = 4).

RADFRQ real	Specifies how often during a model run the radiative parameterization is to be exercised to compute updated values for the radiative contribution to tendencies of both atmospheric and land surface temperatures. The tendencies themselves are applied every timestep of the model integration, but the tendencies are updated only at time intervals specified by **RADFRQ**. Because computing the radiative contribution to temperature tendencies is a relatively computationally expensive process, it is strongly suggested that this computation be done only at intervals of 600 to 1200 seconds (by specifying such values for **RADFRQ**). In most situations, this is often enough to account for changes in radiative transfer.
LONRAD integer	flag specifying whether the longitudinal variation of solar hour angle is to be accounted for in the computation of shortwave solar radiation, or whether these angles are to be assumed constant over the model domain. A value of 0 for **LONRAD** assumes the hour angle is longitudinally constant, and a value of 1 accounts for its longitudinal variation. The latter choice is the one normally made in a simulation of a real event, while the former choice is used when a certain level of idealism is required in a simulation, such as horizontal homogeneity of external forcing.

NNQPARM integer array Range: 0 to 6	Grid-dependent flag used to control whether the convective parameterization is to be activated or not. A value greater than 1 for this flag activates the respective parameterization, and a value of 0 leave it inactive. Convective parameterization is used to vertically redistribute heat and moisture (as if by convection) in a grid column when the model generates a region which is superadiabatic or convectively unstable and when the horizontal grid resolution is too coarse for the model to develop its own convective circulation. Ideally, resolving a convective circulation would require at least a few grid cells to horizontally span an updraft, which for deep convection would normally require the horizontal cell size to be less than 1 or 2 kilometers. Coarser resolution than this would make realization of sufficiently strong vertical motion difficult or impossible to adequately bring about the required vertical exchange of heat and moisture so as to convert the convective available potential energy into other forms. Thus, it is on coarser grids where a parameterized convective adjustment becomes necessary. The current parameterizations implement in BRAMS are: nnqparm= 1- Tremback scheme nnqparm= 2- Grell-Deveny scheme nnqparm= 3- Grell-3d formulation nnqparm= 4- Grell-Freitas scheme as implemented in FIM/NOAA model nnqparm= 5- Grell-Freitas scale and aerosol aware scheme version 2014 nnqparm= 6- Grell-Freitas scale and aerosol aware scheme version 2015
G3D_SPREAD Integer Range: 0,1	Flag to activate the lateral spreading of environmental subsidence in grey zone for deep convection applications (grid spacing < 10 km). Only relevant for Grell 3d scheme (NNQPARM =3). 0=OFF, 1=activated

CLOSURE_ TYPE character(2)	If the Grell's based parameterization for deep convection is used (**NNQPARM>1**), this parameter define the type of closure for the parameterization. Setting **'EN'** defines an ensemble of the closures, which the mass flux is defined by the mean of each mass flux determined by the closures described below. Setting **'GR'** would define the standard Grell closure (quasi-equilibrium assumption). Setting **'LO'** defines a closure based on the omega velocity (bar/s) at cloud base. Setting **'MC'** the closure will be based on the moisture convergence at atmospheric column. Setting **'SC'** defines a closure type like Fritsch-Chappel or Kain-Fritsch, in which the instability is simply removed by convection at some timescale defined in the code. Setting **'AS'** the scheme will use the Arakawa-Schubert closure. Finally, setting "PB" the Betchold et al., 2014 closure will be applied.
NNSHCU Integer Range: 0,1,2,3	Grid-dependent flag used to control whether the Shallow Cumulus parameterization is to be activated or not. A value greater than 0 for this flag activates the parameterization and a value of 0 leaves it inactive. The shallow scheme available in BRAMS are: nnshcu= 1- Souza's scheme nnshcu= 2- Grell-Deveny scheme nnshcu= 3- Grell-Freitas scheme
CONFRQ real	Time interval in seconds during a simulation at which contributions to atmospheric tendencies in temperature and moisture by convective parameterization are to be computed. These tendencies are applied to the temperature and moisture fields every timestep of the model integration, but are updated only at the time interval specified for **CONFRQ**. There are two reasons why those contributions should be updated less frequently than every timestep. One is that a computational expense is involved in exercising the convective parameterization, and frequent exercise thus slows down the model. The change in the tendencies would not be large from one timestep to the next anyway. A second reason has an important physical basis. For example, consider a case of a squall line moving through a grid volume. As the squall line enters the grid volume, it will begin to stabilize the air that it has moved through. However, the air in front of the squall line is still convectively unstable and will contribute to the maintenance of the squall line. The mean atmosphere *i.e.*, that represented on a coarse grid volume, soon may be stabilized by the convection. If the convective tendencies were to be re-evaluated in the model each 5 minutes or so, the result for the above example would be that the original convection might be shut down in its early stages. A more accurate representation of the squall line, however, would be to let it run through the entire grid volume. This is achieved by setting **CONFRQ** to a time scale comparable to the length of that life cycle, such as 1200 to 1800 seconds. **CONFRQ** is relevant only if the parameter **NNQPARM** is set greater than 0 for a particular grid.
SHCUFRQ real	Time interval in seconds during a simulation at which contributions to atmospheric tendencies in temperature and moisture by shallow cumulus parameterization are to be computed. These tendencies are applied to the temperature and moisture fields every timestep of the model integration, but are updated only at the time interval specified for **SHCUFRQ**. **SHCUFRQ** is relevant only if the parameter **NNSHCU** is set greater than 0 for a particular grid. (Exclusive to BRAMS)
WCLDBS real	Use in conjunction with **NNQPARM** and **CONFRQ** , and specifies the threshold value of vertical velocity in m/s in the model at the diagnosed level of cloud base required to initiate convection. The more positive this parameter, the more difficult

	is the initiation of convection. Values for this parameter should normally range from about -0.1 to 0.2. This parameter is relevant only for NNQPARM=1.

NPATCH integer	Number of subgrid patches used in LEAF-3, which is the ***RAMS*** submodel for energy and moisture in soil, snowcover, vegetation, canopy air, and for surface fluxes from these and from water surface areas. LEAF2 is described more fully in section 1c. **NPATCH** must be at least 2, to allow for water surface areas and at least one land surface area in each grid cell. Larger values of **NPATCH** will allow more land surface areas, each with a different vegetation type, soil textural class, and/or wetness index. A value of 2 is often used where the extra detail is not required, while larger values are used for greater accuracy where multiple landuse types may coexist in the same grid cell or where it is desired to model temperature and moisture content of individual, varied ecosystems within the same grid cell.
NVEGPAT integer	Used for any grid whose landuse type is initialized from a standard ***RAMS*** file, i.e., for which **IVEGTFLG =** 1. For any surface grid cell, especially one covering a relatively large geographic area, many landuse vegetation types will often occur in that area and will be read in from the file. ***RAMS*** determines the fractional area of the surface grid cell that is covered by water (oceans, lakes, rivers) and divides the land portion of the grid cell among the N most frequently occurring landuse types read from the file for the cell area, where N is the value specified for NVEGPAT. NVEGPAT should thus usually be set to NPATCH-1 in order to allow for the 1 water patch and fill all land patches for the data file. For special applications, one may wish to set NVEGPAT to a value less than NPATCH-1 to reserve additional land patches for special use. Then, the model will still fill the water patch and the NVEGPAT land patches such that they collectively comprise the entire grid cell area. The user must then reduce the fractional areas of 1 or more of the patches to allow the desired fractional area of the special patch(es) whose number is NPATCH-NVEGPAT-1. NVEGPAT may not be larger than NPATCH-1.
ISFCL Integer Range: 0, 1, 2 and 5.	Flag specifying which of 3 possible options for obtaining surface fluxes of heat and moisture to or from the atmosphere will be used in a simulation. If **ISFCL** is set to 0, the user specifies values for namelist variables **DTHCON** and **DRTCON** representing the result of subtracting a fictitious ground temperature and moisture mixing ratio, respectively, from the surface atmospheric values of the two variables. These differences are then converted to vertical fluxes of heat and moisture to or from the lowest atmospheric level. This is the simpler of the 2 choices, and **ignores** the LEAF3 parameterization available in the model. A value of 1 for ISFCL activates the LEAF3 submodel (described in Section 1c). LEAF3 prognoses soil, vegetation, snowcover, and canopy air temperature and moisture based on vertical diffusion and exchange with the atmosphere of both quantities. (Also see **NZG**, **PCTLCON**, **NSLCON**, **SOILDZ**, **SLZ**, and **STGOFF**). In the current version, SiB submodel is not available. An alternative option is applying the Joint UK Land Environment Simulator (JULES) model. JULES is a state-of-the-art land surface model that includes carbon cycle. JULES is activated setting ISFCL = 5 and it has its own namelist "JULES.in".

NVGCON integer	Flag specifying the landuse class to be used in the LEAF-3 submodel of ***RAMS***, for the case where a simple horizontally homogeneous initialization of landuse type is used (see **IVEGTFLG**). The landuse class is mostly characterized by its vegetation type, but landuse class also defines whether the surface is water, ice cap, wetland, or unvegetated land. By default, the value of this flag is used to fill all surface grid cells. For horizontally inhomogeneous initialization of landuse type, data are normally input from a standard dataset (see **IVEGTFLG**). For direct customizing of inhomogeneous landuse type, the default homogeneous value specified here may be overridden in the file *ruser.f90*. Values for **NVGCON** ranging from 0 to 30 have the following definitions in the standard ***RAMS*** code, but new classes are easily added where information is available: Classes from BATS: 0 Ocean 1 Lakes, rivers, streams (inland water) 2 Ice cap/glacier 3 Evergbreen needleleaf tree 4 Deciduous needleleaf tree 5 Deciduous broadleaf tree 6 Evergreen broadleaf tree 7 Short grass 8 Tall grass 9 Desert 10 Semi-desert 11 Tundra 12 Evergreen shrub 13 Deciduous shrub 14 Mixed woodland 15 Crop/mixed farming 16 Irrigated crop 17 Bog or marsh Classes from LDAS: 18 Evergreen needleleaf forest 19 Evergreen broadleaf forest 20 Deciduous needleleaf forest 21 Deciduous broadleaf forest 22 Mixed cover 23 Woodland 24 Wooded grassland 25 Closed shrubland 26 Open shrubland 27 Grassland 28 Cropland 29 Bare ground 30 Urban and built up Each of these landuse classes is assigned a set of land surface parameter (LSP) values, including leaf area index, vegetation fractional coverage, vegetation height, albedo, and root depth. The LSP values are defined in a data statement in subroutine SFCDATA in the file *leaf2_init.f90*.

PCTLCON **real**	Fraction of a surface grid cell, ranging from 0 to 1, that is covered by land surface, implying that the remainder is covered by water. **PCTLCON** is the default value used when **IVEGTFLG** is set to 2 (see above). If **IVEGTFLG** is 0 or 1, **PCTLCON** is ignored. The distinction between land and water surfaces in the model influences the values of surface fluxes of momentum, heat, and moisture that are exchanged between the atmosphere and surface. Separate calculations of these fluxes are made for each grid cell for the soil and water surfaces, and the resultant net flux is weighted between these two according to the relative area covered by each type of surface. By default, the value assigned to **PCTLCON** is assigned to each surface grid cell in the model. If, as is usually the case, the user wishes to define a horizontally variable distribution of land percent values, subroutine SFCINIT_USER in the file *ruser.f90* needs to be modified, or land percentage values can be read in from a standard file (see **IPCTLFLG**).
NSLCON integer	Soil textural class used in the LEAF-3 submodel of ***RAMS***. **NSLCON** is the default value used when **ISOILFLG** is set to 2 (see above). If **ISOILFLG** is set to 0 or 1, **PCTLCON** is ignored. The soil type controls important soil properties that ultimately affect the influence by the soil on the atmosphere. These properties are the thermal diffusivity, specific heat, moisture capacity (porosity), and the diffusivity of moisture. The thermal properties are themselves highly influenced by the moisture content of the soil, but the particular relation between them is dependent on the soil class. A total of 12 different soil textural classes are parameterized in the model. The following is a list of each soil class and the value of NSLCON that activates it: 1 Sand 2 Loamy sand 3 Sandy loam 4 Silt loam 5 Loam 6 Sandy clay loam 7 Silty clay loam 8 Slay loam 9 Sandy clay 10 Silty clay 11 Clay 12 Peat As in the case for **PCTLCON**, the value assigned to **NSLCON** in the namelist is in turn assigned to the soil type for each surface grid cell in the model. If it is desired to make the soil type horizontally inhomogeneous, the user may modify subroutine SFCINIT_USER in the file *ruser.f90*.

ZROUGH real	Roughness length, a parameter important in representing the effects of ground roughness on the surface fluxes of momentum, heat, and moisture. The parameterization of these fluxes is designed according to surface similarity theory, in which the roughness length is a crucial parameter. A roughness length of 0.01 meter or less represents a smooth surface, while a roughness length of 1.0 meter represents a very rough surface such as a forest of tall trees. As in the case of **PCTLCON**, the value assigned to **ZROUGH** in the namelist is in turn assigned to each grid point in the model. Should the user wish to make roughness length a location-dependent parameter, subroutine SFCINIT_USER in file *rsurf.f90* must be modified. **ZROUGH** only refers to the soil areas of the model surface; the vegetation roughness lengths vary with type and are set in subroutine SFCDATA.
ALBEDO real	Albedo or reflectivity of the ground surface to solar radiation. This parameter only applies when the LEAF-3 submodel is not activated, *i.e.*, for **ISFCL** set to 0. In this case, the albedo is not involved in the computation of surface sensible and latent heat fluxes; it is only used to compute solar radiation reflected upward from the ground, which is needed in the atmospheric radiation budget.
SEATMP real	Temperature in Kelvin of the surface water temperature. **SEATMP** is the default value used only if **ISSTFLG** is set to 2. If **ISSTFLG** is 0 or 1, **SEATMP** is ignored. Horizontally homogeneous initialization of sea surface temperatures by SEATMP may be overridden by modifying subroutine SFCINIT_USER in the file *ruser.f90*. Surface water temperature is relevant for any surface grid cell which is not completely covered by land (see **PCTLCON**), and strongly influences the fluxes of heat and moisture between the surface and atmosphere. If **ISSTFLG** is 2 and SEATMP is therefore used, the model assumes that the surface water temperature is constant in time during a simulation.
DTHCON real	Parameter representing the temperature difference in degrees Kelvin obtained by subtracting a fictitious soil temperature from the temperature of the lowest level of the atmosphere. It is used as a simple means for imposing a surface source or sink of heat to the atmosphere in the case where the soil model is not activated, that is for ISFCL set to 0. For example, setting **DTHCON** to -10 would impose a flux of heat into the atmosphere equivalent to that which would occur over a ground surface 10 K warmer than the air a height around one half **DELTAZ** above the ground.
DRTCON real	Similar to **DTHCON**, being relevant only for ISFCL set to 0, but applies to a difference in water vapor mixing ratio, rather than to a temperature difference. For example, setting **DTHCON** to -0.002 would cause an upward flux of moisture to the atmosphere equivalent to that which would occur if the equilibrium vapor mixing ratio in the soil were 2 grams per kilogram higher than the ambient vapor mixing ratio at low levels in the atmosphere.

SOIL_MOIST character(1)	Set the type of soil moisture initialization. If set to **'n'**, initialize with the standard homogeneous soil moisture defined at parameter **SLMSTR**. If the user desires to use a heterogeneous soil moisture, defined at a file, it can use one of the possible options: **'i', 'h', 'a'**. At any case, the program will initially look for a pre-processed file defined in **USMODEL_IN**. If it is not found then it looks for the file defined in **USDATE_IN** with the raw data. Setting the parameter to **'i'**, the files defined in **USDATA_IN** and **USMODEL_IN** will be used to define a heterogeneous soil moisture only when **RUNTYPE = 'INITIAL'**. Setting to **'h'**, the same will occur when **RUNTYPE = 'HISTORY'** only. Setting to **'a'**, both **RUNTYPE = 'INITIAL'** or **'HISTORY'** will use the file with heterogeneous soil moisture data. If the required files are not found, the next parameter **SOIL_MOIST_FAIL** will define what to do. (Exclusive to BRAMS)
SOIL_MOIST_FAIL character(1)	Define what to do in case of the files defined at **USDATA_IN** and **USMODEL_IN**, necessary if a heterogeneous soil moisture initialization should be used, but are not found. Setting the parameter to **'s'** would cause a stop in the program. Setting to **'h'** standard homogeneous soil moisture initialization will be used. Setting **'l'**, the program will look for until 5 early days, and if they are not found again then the program will stop. (Exclusive to BRAMS)
USDATA_IN character	Soil moisture file prefix. This file represents the raw data. If the program only found this file and the file defined in **USMODEL_IN** do not exist, the file defined in **USMODEL_IN** will be generated. (Exclusive to BRAMS)
USMODEL_IN character	Soil moisture file prefix with the soil moisture content interpolated to the grid(s) model by BRAMS. If this(these) file(s) already exist, the raw data file defined by **USDATA_IN** no longer needs more. (Exclusive to BRAMS)
SLZ real array	Depth of the levels in the soil model in meters. There needs to be as many values specified as **NZG** beginning with the deepest level. For the **SLZ** values to be used, the namelist parameter **SOILDZ** must be set to 0. The **SLZ** values pertain to the *bottom* of the soil layers. Unlike previous ***RAMS***' versions, **SLZ**(**NZG**) should not be set to 0.

SLMSTR real array	Used to initialize the moisture content of the soil. The number of values specified must equal the number of soil levels **NZG** with the values must be separated by commas. The first value specified for **SLMSTR** applies to the deepest soil level, with the last value representing the topmost layer of soil. Allowable values for this parameter range from 0.0, representing totally dry soil, to 1.0 which represents totally saturated soil. A more realistic lower limit is about .15 since soil cannot lose all of its moisture, even when exposed to direct sunlight for extended periods, and ***RAMS*** will in fact impose a minimum value on soil moisture to prevent its initialization too dry. The values represent fraction of total water capacity that the soil can hold. The equilibrium water vapor mixing ratio of the soil is the quantity with which the ambient atmospheric water vapor mixing ratio interacts in determining the flux of moisture between the atmosphere and soil. This soil equilibrium value is a highly nonlinear function of the soil moisture content (represented initially by **SLMSTR**). The equilibrium vapor pressure is essentially constant for soil moisture below about 0.2 and above about 0.7 [depending on the soil type (see **NSLCON**)], and varies sharply in between those values. The **SLMSTR** values are by default used to fill all soil grid points horizontally-homogeneously, but horizontal variations in initial soil moisture may be specified by the user by altering the code in subroutine SFCINIT_USER in the file *ruser.f90*.
STGOFF real array	Used to define a vertical temperature profile in the initial conditions for the soil model, and represents a deviation from the temperature in degrees K in the lowest atmospheric level in the model. The values must be equal in number to NZG and must be separated by commas. The first value of STGOFF applies to the deepest level of soil, while the last value applies to the topmost soil level. For example, if the first value of **STGOFF** is set to 5.0, the deepest soil level will initially have a temperature 5 degrees K. warmer than the initial temperature of the lowest atmospheric level. As for many other soil variables in the namelist, the values specified in **STGOFF** are by default used to horizontally homogeneously initialize all soil model grid points. A user may impose horizontal variations in the initial soil temperature by modifying the code in subroutine SFCINIT_USER in model source code file *ruser.f90*.
IF_URBAN_ CANOPY integer	Flag to activate an experimental urban canopy scheme. This scheme currently requires the pre-computation of drag coefficients defined for a specific urban area. This scheme was not working properly in the current model version.

IDIFFK integer array Range : 1,2,3,4 and 7,8.	Grid-dependent flag that controls the type of parameterization to be used for computing both horizontal and vertical diffusion coefficients. Allowable values for this flag are 1, 2, 3, or 4. A value of 1 or 2 is appropriate to model grids in which the horizontal grid spacing is large compared to the vertical spacing such that dominant convective motions are not resolved. Horizontal diffusion in such cases is normally required to be stronger (for numerical damping) than justifiable on physical grounds, and the model assumes complete decoupling of the horizontal and vertical diffusion in all aspects, including computation of deformation rates, length scales, and stress tensor components. With **IDIFFK** set to either 1 or 2, the horizontal diffusion coefficients are computed as the product of horizontal deformation rate (horizontal gradients of horizontal velocity) and a length scale squared, based on the original Smagorinsky formulation. The length scale is the product of x-direction grid spacing **DELTAX** and namelist parameter **CSX** . When **IDIFFK** is set to 1, vertical diffusion is parameterized according to the Mellor and Yamada scheme, which employs a prognostic turbulent kinetic energy (TKE). When IDIFFK is set to 2, vertical diffusion is computed from a 1-dimensional analog of **the** Smagorinsky scheme in which vertical deformation is evaluated from vertical gradients of horizontal wind, and vertical length scale is the local vertical grid spacing times namelist parameter **CSZ**. In addition, modifications of the vertical diffusion coefficient due to static stability or instability are used, based on formulations of Hill and Lilly. The Lilly modification is in the form of a multiplying factor, equal to sqrt(1-R kh/km) where R is the Richardson number and kh/km is the ratio of the scalar to momentum vertical diffusion coefficients specified by the user in namelist variable **ZKHKM** . The multiplying factor is greater than 1 in unstable cases (*i.e.*, where wind shear and/or unstable lapse rates make the Richardson number sufficiently low, and is less than 1 in stable cases. The Hill modification applies only to regions of unstable lapse rates (having negative squared Brunt-Vaisala frequencies), and consists of adding the absolute value of the Brunt-Vaisala frequency squared to the deformation rate, to obtain a modified inverse time scale for the diffusion coefficient computation. The Lilly and Hill modifications were originally designed for use without each other, although we have found that the added vertical diffusion in unstable air obtained by using both together is usually desirable. Values of 3 or 4 for **IDIFFK** are usually appropriate on grids having comparable horizontal and vertical spacings, and are intended for use in situations where dominant 3-dimensional motions, such as cumulus or boundary layer convection, are resolved. In these cases, ***RAMS*** computes diffusion coefficients from the 3-dimensional rate-of-strain tensor, and uses the same or similar diffusion coefficients for horizontal and vertical diffusion. When **IDIFFK** is set to 3, horizontal and vertical diffusion coefficients are computed as the product of the 3-D rate-of-strain tensor and a length scale squared. The length scale is the product of the vertical grid spacing and namelist parameter **CSZ**. The Hill and Lilly stability-dependent modifications described above also apply to **IDIFFK** set to 3. When **IDIFFK** is set to 4, vertical and horizontal diffusion are parameterized according to the Deardorff scheme, which employs a prognostic subgrid turbulent kinetic energy. This scheme is intended only for the specific purpose of performing a large eddy simulation (LES) in which it is assumed that resolved eddy motions in the model perform most of the eddy transport. The parameterized diffusion only represents the subgrid turbulent mixing. Thus, the Deardorff scheme is not appropriate for horizontal grid cell sizes larger than a couple hundred meters. In both the Deardorff and Mellor and Yamada schemes, the prognostic kinetic energy is generated by means of shear and buoyancy processes, and a parameterized pressure-work term, and is destroyed by potentially all of the above plus a parameterized dissipation term. The kinetic energy is also advected and diffused. The result of these processes yields a field of kinetic energy from which the diffusion coefficients are locally diagnosed. Options 5 and 6 are reserved and must no be applied. Option 7 activates the Nakanishi and Niino (2004) turbulence scheme. The implementation is intended to find the TKE tendency by using a 2.5-level model based on Mellor-Yamada scheme. It's actually an update of the existent M-Y code in RAMS, outputting also the Sig-W (vertical velocity standard-deviation) and the Lagrangian time scale, Obukhov Length and PBL depth, the latter following the method by Vogelezang and Holtslag (1996). This option is required when IMASSFLX is set to 1. Option 8 activates the Taylor's theory based formulation for KZZ (Campos Velho, 1998)

CSX real array	Grid-dependent parameter serving as a coefficient in computing horizontal diffusion coefficients, in the case where **IDIFFK** is set to either 1 or 2. It is used in computing the horizontal diffusive length scale, defined as the product of **CSX** and the x-direction grid spacing. Doug Lilly and others have proposed specific values for **CSX** based on theory, although the philosophy on which such a theory is based has been a matter of controversy. Setting **CSX** to 0.135 is effectively the value proposed by Lilly to yield diffusion coefficients of a strength that just provides adequate damping of the smallest resolvable features on the grid (to prevent an unrealistic accumulation of energy at that small scale). However, justification for a considerably larger of **CSX** of 0.32, has been argued by Paul Mason for more strongly filtering features represented on the grid, so that the only features with wavelengths longer than about 4 or 5 times the grid spacing will exist at any significant amplitude on the grid. This choice is desirable since the the smaller wavelengths of 2 and 3 times the grid spacing are not properly handled by numerical finite differencing schemes. The price paid for this stronger diffusive filter is the requirement of more grid cells for simulating a particular feature. The user must make a judgment here as to how strong to make **CSX**. This is often done on a case-by-case basis based on a subjective decision as to whether the model prognostic fields are "sufficiently smooth".
CSZ real array	Similar to **CSX**, but it applies to the computation of vertical, rather than horizontal, diffusion coefficients, and is relevant only for **IDIFFK** set to 2 or 3.
XKHKM real array	Grid-dependent parameter in which the user specifies the ratio of strength of horizontal diffusion coefficients between scalars and velocity to be used in a simulation. It is used to multiply the momentum diffusion coefficient to obtain the scalar coefficient, and applies to cases where **IDIFFK** is set to 1 or 2. Deardorff found that the scalar coefficient had to be around 3 times larger than the momentum coefficient in strongly convective boundary layers for adequate mixing of all fields. For stable situations, the ratio may decrease to 1. It has been our usual practice to set **XKHKM** to 3, a value which applies throughout the model domain regardless of local stratification. The Deardorff parameterization for large eddy simulations (activated by setting **IDIFFK** to 4) includes an automatic computation of this ratio for each grid cell in the domain based on the local conditions.
ZKHKM real array	Similar to **XKHKM**, but applies to the computation of vertical, rather than horizontal, diffusion coefficients. It is applicable only in cases where **IDIFFK** is set to 2 or 3.

AKMIN real array	Is a multi-valued, grid-dependent parameter used to impose a minimum value on the value of the horizontal diffusion coefficients throughout the model domain. It is only relevant for cases where **IDIFFK** is set to 1 or 2. The minimum value is sometimes required when the local fluid deformation rate happens to be close to zero in a region, resulting in very weak diffusion there. Setting **AKMIN** to a value greater than 0 will prevent the horizontal diffusion coefficients everywhere from falling below the minimum. **AKMIN** is scaled such that if set to 1.0, the minimum value placed on the diffusion coefficients should be about the same as normally obtained from the basic deformation scheme itself (without buoyancy effects) regardless of grid spacing. It is suggested that **AKMIN** be set to a low value, say 0.1, so that the standard horizontal diffusion parameterization be given a chance to do its job. If it is found that horizontal diffusion is inadequate (for example, as seen by considerable grid noise appearing in a contour plot of model fields), first try increasing **CSX** up to as much as .32, and if more diffusion is still required, then increase **AKMIN** to as much as 1 or 2, as required.
MCPHYS_TYPE Integer Range: 0 to 3	Flag that specifies the type cloud microphysics scheme will be applied in the model: 0 = previous RAMS/BRAMS version 1 = double-moment from CSU RAMS version 6.0^{+} and aerossol aware 2 = Thompson cloud water single-moment 3 = Thompson cloud water double-moment and aerossol aware Important: all microphysics flags below are relevant only for mcphys_type = 0 or 1.

LEVEL integer	Flag that specifies the level of moisture complexity to be activated in the model. This flag is the primary means by which the user tells the model whether to consider the effects of moisture in the simulation, and if so, to what degree. LEVEL = 0 - causes the model to run dry, completely eliminating any process which influences or is influenced by any phase of moisture. With this option, radiation parameterizations (controlled by ISWRTYP and ILWRTYP) must be turned off. LEVEL = 1 - activates advection, diffusion, and surface flux of water, where all water substance in the atmosphere is assumed to occur as vapor even if supersaturation occurs. The value of 1 also activates the buoyancy effect of water vapor in the vertical equation of motion, as well as the radiative effects of water vapor if radiation is activated elsewhere. LEVEL = 2 - activates condensation of water vapor to cloud water wherever supersaturation is attained. The partitioning of the total water substance into vapor and cloud water is purely diagnostic in this case. No other forms of liquid or ice water are considered. Both the positive buoyancy effect of water vapor and the liquid water loading of cloud water are included in the vertical equation of motion. Radiative effects of both water vapor and cloud water are activated, if the radiation parameterization is itself activated. LEVEL = 3 - activates the bulk microphysics parameterization, which includes cloud water, drizzle, rain, pristine ice, snow, aggregates, graupel, and hail, or certain subsets of these as specified by ICLOUD, IDRIZ, IRAIN, IPRIS, ISNOW, IAGGR, IGRAUP, and IHAIL. This parameterization includes the precipitation process.

IRIME **integer**	Choice of standard bulk riming (IRIME=0) or hybrid-bin riming of cloud droplets via method of moments (IRIME=1). (Saleeby and Cotton 2008, JAMC).

IPLAWS **integer**	IPLAWS=0 uses original hydrometeor species fall speed power laws from RAMS 4.3. IPLAWS=1 uses non-banded, but newer power laws via Mitchell (1996). IPLAWS=2 uses size banded newer power laws via Mitchell (1996) and modified by Rob Carver and Jerry Harrington (PSU).

ICLOUD **integer**	Is similar to IRAIN, but applies to the cloud water category. In RAMS version 4.3, prediction of cloud droplet number concentration is not implemented, so ICLOUD must be less than 5. Values of 1 or 4 are normally very preferable over ICLOUD = 2. In RAMS version 4.4, cloud droplet number prediction is an option, and is used by setting ICLOUD to 5, 6 or 7. ICLOUD = 5 nucleates cloud droplets from a constant concentration of cloud condensation nuclei (CCN). ICLOUD = 6 nucleates cloud droplets from CCN in which exponential decrease with height. When ICLOUD is set to 7, a three-dimensional CCN field must be set in mic_init.f90. The CCN field is initialized by default from the concentration specified in CPARM (see below) in units of #/cm3. Aerosols evolve in time after initiation. ICLOUD is relevant only when LEVEL is set to 3.

IDRIZ **integer**	Similar to ICLOUD, but controls the activation of drizzle rather than cloud. IDRIZ cannot = 5 unless ICLOUD=5. If IDRIZ > 0, then cloud droplets generally pass through the drizzle mode during growth before becoming rain. This slows autoconversion a bit. If IDRIZ = 5, then aerosols can nucleate directly to the drizzle mode if the median radius of the activated aerosol distribution > ~1.0 microns.

IRAIN Integer	Flag that controls the model's treatment of rain water, and is relevant only if LEVEL is set to 3. If IRAIN is set to 0, rain is not activated, and any process involving the interaction of rain with other water species is not performed. Rain is activated when IRAIN is set to a value from 1 to 5. Activation means that the mixing ratio of rain is prognosed from conservation equations which include advective, diffusive, and precipitation tendencies, and source terms resulting from interactions between rain and other forms of water substance. The choice of values for IRAIN between 1 and 5 controls the manner in which the mean rain droplet diameter and number concentration are determined. With the flag equal to 1, the mean diameter is specified from a default value in the code, and the number concentration is diagnosed automatically from this mean diameter and the prognosed mixing ratio. When the flag is set to 2, the user specifies the mean diameter (in meters) in the namelist variable RPARM, and the number concentration is diagnosed as before. When the flag is set to 3, the user specifies the y-intercept value of the number concentration (this is the number concentration per unit diameter increment, (i.e., number per m^3 per m), evaluated at zero diameter, assuming a Marshall-Palmer (exponential) size distribution. The total number concentration (i.e., the total number of droplets per cubic meter) and the mean diameter are then both diagnosed from the y-intercept value and the prognosed mixing ratio. With a value of 4 for the flag, the user specifies the total number concentration of rain droplets (number per kg of air) directly in the namelist variable RPARM, and the mean diameter is automatically diagnosed from this concentration and the prognosed mixing ratio. When the flag is set to 5, the model activates a prognostic equation for rain droplet number concentration, and a special array for this quantity is added. Mean rain droplet diameter is then diagnosed from the prognosed mixing ratio and number concentration. Note that if any of the flags IRAIN, ISNOW, IAGGR, IGRAUP, or IHAIL is set to 5, the model will override settings of 1, 2, 3, or 4 for ALL of the others and make them 5. (Any that are set to zero will remain zero.) Thus, the model requires that if number concentration is predicted for any of these, it must be predicted for all of these species that are active. IPRIS is not in this list; it must ALWAYS be set to 5 (if pristine ice is activated). Independently of how flags for the other species are set. The choice of which value of IRAIN to use will depend to a large extent on how the user wishes to constrain the rain field, i.e., whether mean diameter, total number concentration, or both should be allowed to vary. For example, if the user is simulating an observed and documented precipitation event in which, say, mean droplet diameter was measured, it would make sense to specify that diameter as a fixed quantity. Independent selection of microphysical species to be activated versus not activated is controlled through the setting of IRAIN and the related parameters IPRIS, ISNOW, IAGGR, IGRAUP, and IHAIL described below. However, not all possible combinations are intended as valid nor would they function properly without code modification. Activation of all categories simultaneously is the normal practice, although activation of the limited sets {cloud}, {cloud, rain}, {cloud, pristine ice}, {pristine ice}, {pristine ice, snow}, {pristine ice, aggregates}, and {pristine ice, snow, aggregates} are reasonable when only a limited range of microphysical processes need to be considered.

IPRIS integer	Similar to IRAIN, but controls the activation of pristine ice rather than rain. The descriptions of the various settings given for IRAIN apply here, except that if pristine ice is activated at all its number concentration must be predicted as well. Hence, values of 1, 2, 3, and 4 are not permitted for IPRIS. With IPRIS = 5, BRAMS assumes that ice forming nuclei (IFN) have its concentration initially based on PPARM.
ISNOW integer	Similar to **IRAIN**, but controls the activation of snow rather than rain. The descriptions of the various settings given for **IRAIN** apply here.
IAGGR integer	Similar to **IRAIN**, but controls the activation of aggregates rather than rain. The descriptions of the various settings given for **IRAIN** apply here.
IGRAUP integer	Similar to **IRAIN**, but controls the activation of graupel rather than rain. The descriptions of the various settings given for **IRAIN** apply here.
IHAIL integer	Similar to **IRAIN**, but controls the activation of hail rather than rain. The descriptions of the various settings given for **IRAIN** apply here.

ICCNLEV integer	Flag for aerosol treatment complexity. First note that when multiple aerosol categories are turned on, they will compete for nucleation based on the surface area of the fraction of particles that may nucleate. (see Saleeby and van den Heever 2013, JAMC) ICCNLEV=0 allows aerosol activation for nucleation but only nucleates additional aerosol in excess of the number of cloud droplets present in a given grid cell. Aerosols are not removed from the domain by nucleation. ICCNLEV=1 allows aerosol removal upon nucleation of new cloud droplets. Removal from each category starts at the large end of the distribution first. Largest aerosols are removed first and the median radius is then recalculated before the next timestep and will be smaller since the larger aerosols were removed. ICCNLEV=2 is the same as ICCNLEV=1 but now we store the nucleated and removed aerosol mass in tracking arrays. A tracking array exists for each hydrometeor species. First, aerosol mass will enter the cloud category tracking array. As cloud is collected or undergoes any transfer to another category, a proportional amount of aerosol mass-within-cloud will be transferred as well. All possible transfers among hydrometeor species is accounted for. Upon hydrometeor evaporation, aerosols will be restored to a regenerated aerosol category. For regeneration, the mass of regenerated aerosols is proportional to the mass of evaporated hydrometeor mass. The number of restored aerosols equals the number of fully evaporated hydrometeors in each category. If the median radius of the regenerated aerosols is < 1 micron, then the aerosols enter the sub-micron regenerated category. If > 1 micron, they enter the super-micron regenerated category. NOTE: ICCNLEV=2 will function for single moment microphysics (except ICLOUD must = 5) but any regeneration will be suspect since number concentration may be diagnosed rather than predicted. This could lead to regeneration of unrealistic numbers of aerosols.

CPARM real	Similar to RPARM below, but applies to cloud water rather than rain. If ICLOUD is set to 5 (see above), CPARM represents the concentration of cloud concentration nuclei (CCN) in number per kilogram of air. If ICLOUD is set to 7, a prognostic CCN field is activated in RAMS, and CPARM then represents the initial value (in number per kilogram of air) of the CCN concentration field. Similar to RPARM below, but applies to cloud water rather than rain. If ICLOUD is set to 5 (see above), CPARM represents the concentration of cloud concentration nuclei (CCN) in number per kilogram of air. If ICLOUD is set to 7, a prognostic CCN field is activated in RAMS, and CPARM then represents the initial value (in number per kilogram of air) of the CCN concentration field.
DPARM real	Similar to RPARM, but applies to drizzle rather than rain. DPARM is linked to giant cloud condensation nuclei (GCCN) as CPARM is linked to CCN when IDRIZ is set to 5, 6 or 7.
RPARM real	Specifies either a fixed mean diameter (in meters), a fixed y-intercept number concentration per unit diameter increment (number per m^3 per meter), or a fixed total number concentration (in number per kg of air) to be imposed on the rain droplet field, in conjunction with settings 2, 3, and 4, respectively, for the flag **IRAIN** described above. If **IRAIN** is set to 0, 1, or 5, **RPARM** is ignored.
PPARM real	Obsolete as pristine spectrum parameter (mean diameter, y-intercept number concentration or total number concentration) since pristine ice number concentration must always be prognosed (IPRIS = 5). PPARM is an initialization parameter for ice forming nuclei (IFN) concentration. IFN is a prognostic variable.
SPARM real	Similar to **RPARM**, but applies to snow rather than rain.
APARM real	Similar to **RPARM**, but applies to aggregates rather than rain.
GPARM real	Similar to **RPARM**, but applies to graupel rather than rain.
HPARM real	Similar to **RPARM**, but applies to hail rather than rain.

CNPARM real	Median radius of the CCN spectrum in terms of distribution (cm)
GNPARM real	Median radius of the GCCN spectrum in terms of distribution (cm)
EPSIL real	Aerosol solubility fraction (0 to 1)

GNU real array	Parameter with 8 values, one for each hydrometeor category. It specifies the shape parameter of the gamma distribution that all categories in the bulk microphysics submodel are assumed to follow. **GNU =** 1 indicates the Marshall-Palmer, or exponential, distribution, in which number concentration decreases monotonically with diameter throughout the size spectrum. Larger values of **GNU** indicate more general gamma distributions, in which the size distribution peaks at a positive diameter. The larger the value of **GNU**, the more narrowly distributed the spectrum is. The intent in using this variable is to specify a shape based on observation. The shape may in general depend on the type of precipitating system (deep convective, wintertime cyclone, etc.), as well as on the hydrometeor category (rain versus hail, for example). Little observational guidance is available to date on appropriate values of **GNU**. A Marshall Palmer distribution has been used in many models, although it may not be the closest of the gamma distribution family to real hydrometeor spectra. Sensitivity experiments with **GNU** in ***RAMS*** have demonstrated that it can sometimes have a significant effect on model results. The user is encouraged to experiment with the values of **GNU**, and to use observational guidance where possible. Ongoing research is aimed at determining appropriate values of **GNU** for each hydrometeor category in different weather situations. Values of 2 are suggested as middle-of-the-road values to start from, but it is sometimes argued that cloud spectra are generally narrower and should have gnu equal to at least 5.

$MODEL_SOUND Namelist

The **$MODEL_SOUND** namelist consists of a set of variables for specifying a sounding to be used in initializing a simulation. This method of initialization is performed horizontally homogeneously, and is performed only when the flag **INITIAL** is set to 1. The variables in the **$MODEL_SOUND** namelist consist mainly of (1) arrays containing the actual values of pressure or height, velocity, temperature, and moisture, and (2) a set of flags specifying how the array values are to be interpreted. An alternative form of specifying five of the variables in this namelist, **PS**, **TS**, **RTS**, **US**, and **VS**, all described below, is to enter them into a file called *SOUND_IN* in tabular form. This option is selected by setting the first value of PS to 0. In this case, the model opens and reads the file *SOUND_IN*, which must reside in the same directory as the model executable. The code that reads from this file is contained in the file *rhhi.f90* in subroutine **ARRSND** in the first DO loop. Free format is assumed with one sounding level per record starting with the lowest sounding level at the top of the file, and the five variables are read from each record in the order shown above.

Variable name	Description
IPSFLG integer	Is a flag specifying how the values given for **PS** are to be interpreted. When **IPSFLG** is set to 0, all values in PS are interpreted as pressures in millibars, and when **IPSFLG** is set to 1, all values in PS are interpreted as heights in meters. In either case, the **PS** values are used to define the heights in the sounding at which all sounding data applies.
ITSFLG integer	Is a flag specifying how the values given for **TS** are to be interpreted. When **ITSFLG** is set to 0, all values in **TS** are interpreted as temperatures in degrees C, when **ITSFLG** is set to 1, all values in **TS** are interpreted as temperatures in degrees K, and when **ITSFLG** is set to 2, all values in **TS** are interpreted as potential temperatures in degrees K. The options allow the model to read sounding temperature data in any of these forms.
IRTSFLG integer	Flag specifying how the values given for **RTS** are to be interpreted. When **IRTSFLG** is set to 0, all values in **RTS** are interpreted as dew point temperatures in degrees C, when **IRTSFLG** is set to 1, all values in **RTS** are interpreted as dew point temperatures in degrees K, when **IRTSFLG** is set to 2, all values in **RTS** are interpreted as water mixing ratio values in grams per kilogram, when **IRTSFLG** is set to 3, all values in **RTS** are interpreted as relative humidities in percentage, and if **IRTSFLG** is set to 4, all values in **RTS** are interpreted as dew point depressions in degrees K. The options allow the model to read sounding moisture data in any of these forms.
IUSFLG integer	Flag specifying how the values given for **US** and **VS** or for **USNDG** and **VSNDG** are to be interpreted. When **IUSFLG** is set to 0, all values in **US** (or **USNDG**) are interpreted as components of velocity in meters per second in the x-direction, and all values in **VS** (or **VSNDG**) are interpreted as components of velocity in meters per second in the y-direction. When **IUSFLG** is set to 1, all values in **US** (or **USNDG**) are interpreted as wind directions in degrees azimuth clockwise from true north, and all values in **VS** (or **VSNDG**) are interpreted as wind speeds in meters per second in the x-direction.

HS real array	Sounding heights, but only the first value is read from the namelist. This first value indicates the absolute height (relative to sea level) of the first or lowest sounding level. It must be at or below the lowest topography height in the ***RAMS*** model domain. This may require adding one or more levels to the bottom of a real sounding.
PS real array	Pressures in millibars or the heights in meters of the sounding levels starting at the ground and proceeding upward. The choice of meaning is controlled by the value of **IPSFLG**. If **PS** represents heights, the first value of **PS** is still specified as a pressure in millibars, and its corresponding height is read from namelist variable HS. The primary requirement of the sounding data is that it extend to a height greater than the top of the model domain, and at least as low as the lowest topography in the domain. Thus, the top sounding level must contain either a height or a pressure higher in the atmosphere than the model domain top, which is itself determined by the combination of namelist parameters **NNZP**, **DELTAZ**, **DZRAT**, and **DZMAX**, or by **ZZ**. A maximum of 200 values may be specified for **PS**, although with code modification this number could be increased if necessary. The values given for **PS** will determine the vertical spatial resolution of the sounding, and should be chosen to properly represent any important significant levels. **Note the special option of setting the first value of PS to 0., as described at the beginning of this Section.**
TS real array	Sounding temperature values. The way in which the values are interpreted is specified by namelist variable **ITSFLG**. The number of values specified for **TS** must equal the number of values specified for **PS**. The user should use caution in specifying sounding temperatures. Superadiabatic profiles in the sounding are strongly discouraged (which does not mean that the model simulation may not successfully develop superadiabatic regions after initialization). High vertical resolution is sometimes desirable in a sounding because the model fields of temperature and moisture mixing ratio are interpolated linearly from it in the vertical direction, and relative humidity is a very nonlinear function of these two interpolated fields. This can result in supersaturated layers in the model initial fields where none exist in the sounding.
RTS real array	Sounding moisture values. The way in which the values are interpreted is specified by namelist variable **IRTSFLG**. The number of values specified for **RTS** must equal the number of values specified for PS. (Also see **TS**.)
US real array	Sounding wind values. The way in which the values are interpreted is specified by namelist variable **IUSFLG**. The number of values specified for **US** must equal the number of values specified for **PS**.
VS real array	Sounding winds values. The way in which the values are interpreted is specified by namelist variable **IUSFLG**. The number of values specified for **VS** must equal the number of values specified for **PS**.

$MODEL_PRINT Namelist

The **$MODEL_PRINT** namelist provides a means for obtaining a quick look at model fields. It is used to specify selected data from the model to be written to the standard output file generated with a model run. This data is then examined by displaying the contents of the standard output file.

Variable name	Description
NPLT integer	Total number of data subsets to be written to the standard output file. Most of these data subsets consist of a 2-D cross section of values from a 3-D model field. Different cross sections from the same model field are considered as separate data subsets. (If the model itself is run in 2-D, the 2-D cross section normally encompasses the entire model domain.) The exact specification of the cross section to be output is made in namelist variables **IPLFLD**, **IXSCTN**, and **ISBVAL**. Fields from the LEAF2 submodel of ***RAMS*** are output in their entirety rather than as a specified 2-D slab.
IPLFLD character array	Variables to be written to the standard output file. Each value specified for **IPLFLD** must be followed by a comma, and the total number of values specified must be at least the value specified for **NPLT**. The authentic complete list of character strings that can be used as values for **IPLFLD** is provided in FUNCTION OPTLIB and in subroutine SFCPRT located in the file *rprnt.f90*. Other choices are easily added to OPTLIB.
PLFMT character array	FORTRAN format to be used in writing a data subset (see **NPLT**) to the standard output file. Although a default format exists in the model for every field variable, defining alternative formats in **PLFMT** is a way of overriding the default when necessary. Any element of **PLFMT** that the user wishes to define a value for is listed individually. We illustrate the use of this variable by means of the following example: **PLFMT**(2) = '3PF6.1'. The subscript 2, denoting the second element in the **PLFMT** array, corresponds to the second element specified in the **IPLFLD** array. The character string assigned to **PLFMT**(2) denotes the format to be used in writing that second field in **IPLFLD**. The format statement itself contains the F6.1 part referring to the well-known floating point format, and the less-well-known 3P part which scales (multiplies) each value by 10**3 before writing it. This latter feature is sometimes necessary when values in a given field become unusually large or small, and are thus incompatible with the default format. For example, the default format for vertical velocity involves a multiplication of the model values by 100 so that output values are expressed in cm/sec. This is usually best for mesoscale simulations, but not appropriate for simulations of deep convection which generate vertical velocities of tens of meters per second.
IXSCTN integer array	Controls the orientation of the 2-D cross section extracted from the given model field for writing to the standard output file. A value of 1 for **IXSCTN** specifies the data subset to be an X-Z cross section, a value of 2 denotes a Y-Z cross section, and a value of 3 denotes an X-Y cross section. Fields from the LEAF2 submodel of ***RAMS*** (**IPLFLD =** 'TGP', 'TGPT', 'WGP', 'SCHAR', or 'GSF') are output in their entirety. Each element in the **IXSCTN** array refers to the corresponding element in the **IPLFLD** and **ISBVAL** arrays. A number of values equal to or greater than NPLT must be specified.

ISBVAL integer array	Selects a specific 2-D slab from a 3-D model field among all slabs having the orientation specified by **IXSCTN**. Each number specified for **IXSCTN** refers to the grid location index in the direction perpendicular to the slab, with the value 1 denoting either the westernmost, the southernmost, or the lowest slab, depending on the orientation. Each element in the **IXSCTN** array refers to the corresponding element in the **IPLFLD** and **IXSCTN** arrays. A number of values equal to or greater than **NPLT** must be specified.

RAMS ISAN Configuration Parameters – Fortran 90 Module: "isan_coms .f90"

The following configuration parameters are defined in the *isan_coms.f90* file (encountered at <BRAMS_ROOT_DIRECTORY>/src/brams/modules/). They function as dimensions for several arrays and are responsible for the maximum values that can be set for some namelist variables during execution. These are mentioned for completeness and would not normally need to be changed except in extreme circumstances. They may be set to fairly large values in a standard model installation, as their settings do not affect total memory usage very much; most of the memory is allocated dynamically. Note that an index of configuration and namelist variables for chapters 6, 7, and 8 is located at the end of chapter 8.

MAXPR integer	Maximum number of vertical levels that can be used in the pressure data.
MAXISN integer	Maximum number of vertical levels that can be used in the isentropic analysis.
MAXX integer	Maximum number of grid points (in the "x" or east-west direction) in any of the ***RAMS*** or pressure grids.
MAXY integer	Maximum number of grid points (in the "y" or north-south direction) in any of the ***RAMS*** or pressure grids.
MAXTIMES integer	Maximum number of data analysis times that can be processed in a single run.
MAXAGRDS integer	Maximum number of ***RAMS*** grids that can have varfiles generated.
MAXSIGZ integer	Maximum number of vertical levels that can be used in the σ_z analysis.
MAXLEV integer	Maximum number of levels in an input rawinsonde.
MAXSNAME	Maximum number of input observations
MAXISFILES	Maximum number of input data times

RAMS ISAN Namelists

Variable name	Description
ISZSTAGE integer	Switch with a value of 0 (no) or 1 (yes), indicating whether the isentropic/σ_z stage of ISAN is to be run. This stage inputs observational upper air datasets, as prepared separately in 'RALPH2 format' and objectively analyzes the data onto both isentropic and σ_z surfaces. It also inputs surface observations and objectively analyzes them onto a surface that is itself an objectively analyzed height surface based on surface elevations of the surface stations. This stage and the subsequent 'varfile' stage (see **IVRSTAGE**) can be run at once or each can be executed individually. The isentropic stage outputs the isentropic and σ_z files which the varfile stage will input if run separately.
IVRSTAGE integer	Similar to **ISZSTAGE** (see above) but applies to the 'varfile' stage, which produces varfiles for model initialization from isentropic and σ_z fields generated by the isentropic/σ_z stage of ISAN. (Also see
ISAN_INC integer	Desired interval between consecutive data processing times. Certain files may be available at more frequent times than others, as, for example, surface observations in contrast with rawinsonde observations. Setting **ISAN_INC** to the standard 12-hour interval between rawinsondes will cause surface observations to be processed at only the rawinsonde times. The beginning and duration of the time period to processes observational data are specified in **IYEAR1**, **IMONTH1**, **IDATE1**, **ITIME1**, and **TIMMAX** in the **$GRIDS** namelist. The format for **ISAN_INC** is *hhmm*.
GUESS1ST character	Set to 'PRESS' or to 'RAMS' to specify whether the first guess field in the objective analysis of observational data is to be interpolated from pressure level data or taken from a ***RAMS*** analysis file.
I1ST_FLG integer	What choice the model should make if the first guess field is not found but was requested to be used. This situation may arise, for example, in an operational setting where observational data are expected periodically from an outside source, but where something has prevented the data from being available. If **I1ST_FLG** is set to 1, it instructs ISAN to skip all processing of this data time and proceed with the next time. This will result in a particular varfile not being generated. If **I1ST_FLG** is set to 2, it instructs ISAN to stop. IF **I1ST_FLG** is set to 3, it instructs ISAN to generate a first guess field by interpolating between the previous and next available data times. If either expected datafile likewise does not exist, ISAN will stop. ***I1ST_FLG = 3 is not implemented yet.***
IUPA_FLG integer	Similar to **I1ST_FLG**, but applies to input rawinsonde data instead of the first guess field. Another difference: if IUPA_FLG is set to 3, ISAN is instructed to process the given data time without rawinsonde data.
ISFC_FLG integer	Similar to **I1ST_FLG**, but applies to input surface data instead of the first guess field. Another difference: If **ISFC_FLG** is set to 3, ISAN is instructed to process the given data time without surface data.

IAPR character	Filename prefix, with directory path if applicable, of the input pressure files that were generated by dataprep. All files that have this prefix (and that are within the specified processing parameters) will be processed.
IARAWI character	Filename prefix, with directory path if applicable, of the input rawinsonde files that were generated by dataprep. All files that have this prefix (and that are within the specified processing parameters) will be processed.
IASRFCE character	Filename prefix, with directory path if applicable, of the input surface observation files that were generated by dataprep. All files that have this prefix (and that are within the specified processing parameters) will be processed.
VARPFX character	Filename prefix, with directory path if applicable, of two classes of files that are output from ISAN. The first file type is the 'varfile', which is the final stage of processing by ISAN and contains velocity, pressure, potential temperature, and vapor mixing ratio fields interpolated to the model grid(s) and ready for initialization and/or time-dependent data assimilation. The other file type is the 'isentropic/σ_z file', which contains the output from the isentropic, σ_z, and surface analyses that are performed before the varfile stage of ISAN (see **ISZSTAGE**). While both file types are assigned names that begin with the prefix plus the year, month, date, and time automatically appended, the varfile name is additionally given a 'V' character, while the isentropic/σ_z file name is given an 'I' character. One run of ISAN may process data at one or many analysis times, and files for all times are named with the same prefix but different time designations.
IOFLGISZ character	Flag indicating whether to write out the isentropic/σ_z files: 0 – no, 1 – yes. (See **ISZSTAGE** and **VARPFX**).
IOFLGVAR character	Flag indicating whether to write out the varfiles: 0 – no, 1 – yes. (See **ISZSTAGE** and **VARPFX**).

$ISAN_ISENTROPIC Namelist - ISAN Isentropic/σ_z Stage

Variable name	Description
NISN integer	Number of isentropic levels on which to perform objective analysis. A value around 40 is suggested for **NISN** to provide adequate vertical resolution.
LEVTH integer array	Isentropic levels in integer degrees Kelvin of the isentropic grid on which objective analysis is performed. The spacing that normally provides sufficient vertical resolution is 1-2 K near the ground, 3-5 K in the remainder of the troposphere, 20-30 K in the lower stratosphere, and 50 K in the middle stratosphere. It is now common practice in ***RAMS*** to use the NCEP reanalysis pressure level data as the first guess field in ***RAMS***, and the top level of this dataset is 10 mb. Thus, isentropic levels may be specified to values as high as 800 K, in case the model domain is unusually high.
NIGRIDS integer	Number of ***RAMS*** grids to analyze. This must be less than or equal to **NGRIDS** in the **$MODEL_GRIDS** namelist. Although the ability exists to analyze atmospheric data and generate a varfile independently for each ***RAMS*** grid, this option does not yet produce analyses that match smoothly across nested grid boundaries. Thus varfiles for nested grids should be used only for model initialization and not 4DDA (Newtonian nudging during model runtime) (see **TNUDCENT**). Gradients of prognostic variables initialized at grid boundaries will tend to weaken with time if they are not continuously restored by nudging.
TOPSIGZ real	σ_z analysis will be done to all model levels under this height (in meters).
HYBBOT real	The hybrid analysis between the isentropic and σ_z datasets is accomplished in this version of ISAN by a weighted blending of the data over a specified, terrain-following layer using a linear (in height above the surface) weighting profile. At the bottom of the layer, the data is completely from the σ_z analysis; at the top of the layer, the data is completely from the isentropic analysis. **HYBBOT** is the approximate height above the surface of the bottom of this layer. The actual bottom is found as the closest model level to **HYBBOT**.
HYBTOP real	Approximate height above the surface of the top of the isentropic/σ_z blending layer (see **HYBBOT** above). The actual height of the top is found as the closest model level to **HYBTOP**.
SFCINF real	In addition to the blending of the isentropic and σ_z analysis in the creation of the varfile, the surface data analysis is also blended in a layer near the ground. This only occurs if there are actual surface observations close enough to a grid point and the model level is within a certain vertical distance from the actual station height. This distance is **SFCINF** (meters). The actual weight given to the surface analysis varies from a full weight if the difference between the model level and station height is 0. to no weight for the surface (full weight for the upper air data) when the difference reaches **SFCINF**.

SIGZWT real	Optional weight given to the σ_z analysis in the blending process. If **SIGZWT** is set to 0., then the varfile will only contain information from the isentropic analysis. If **SIGZWT** is set to 1., then the varfile will contain full information from the σ_z analysis under **HYBBOT**. Any value between 0. and 1. may be used. For most applications, this should be set to 1.
NFEEDVAR integer	In the creation of the varfiles when more than one grid is analyzed, the data can optionally go through the nesting "feed back" process. The feed back must be done if the 4-dimensional data assimilation options are to be used. Set **NFEEDVAR** to 1 to do the feed back. Set **NFEEDVAR** to 0 usually only if you are doing a data analysis on different grid resolutions, will not be running the model, and you want to see the unmodified results of the different resolution analyses. Normally, leave this set to 1.
MAXSTA integer	Maximum number of rawinsondes expected in the data access. Memory will be allocated using this number, so don't set it too big.
MAXSFC integer	Maximum number of surface observations expected in the data access. Memory will be allocated using this number, so don't set it too big.
NOTSTA integer	Number of stations in the domain area to be excluded if they are found. The specific stations to be excluded are indicated by **NOTID** (see below). Excluding stations in this way is the means for eliminating stations with bad data.
NOTID character array	Station IDs to be excluded from any further processing (see **NOTSTA** above). Prefix the station ID with an 'r' for rawinsonde or an 's' for surface observation.
STASEP real	Surface observations will be objectively analyzed with the Barnes scheme that assumes uniform data coverage. Frequently, there are 2 surface stations reporting in the same city. As the surface stations are processed, any station within a distance STASEP (in degrees of latitude) of a previously processed station will be discarded unless the new one has less missing data than the previous one, in which case the other previous station will be discarded.
IOBSWIN integer	
IGRIDFL integer	Flag which controls blending of gridded pressure level data and observations. If **IGRIDFL =** 0, no pressure grid point data is used, only observations. If **IGRIDFL =** 1, all observations (those not discarded as specified by the **STASEP** and **NOTSTA** parameters - see above) and all pressure grid point data are used. If **IGRIDFL =** 2, all observations (those not discarded) are used, but pressure data are used only from geographic locations that (1) have no sounding closer than the distance specified in **GOBRAD** and (2) do not have soundings in 3 or 4 quadrants (defined by intersecting east-west and north-south lines) all closer than the distance specified by **GOBSEP** (see **GOBRAD** and **GOBSET**). If **IGRIDFL =** 3, only gridded pressure level data are used, no observations. If **IGRIDFL =** 4, the gridded pressure data is taken as a first guess field. The observations are objectively analyzed and applied to this first guess field as deviations.

GRIDWT integer array	The implementation of the Barnes scheme in ***RAMS*** allows differential weighting of any station or pressure data point to give it more or less influence compared to other stations. **GRIDWT** has been implemented to reduce the weight of the gridded pressure data set relative to the actual observations. For example, if a rawinsonde observation were made at the same location as a gridded pressure point, a setting of **GRIDWT** = .1, would give the rawinsonde 10 times more weight than the pressure data. However, remember that the gridded data now are interpolated to the actual ***RAMS*** grids, so that there are many more pressure grid points than soundings. Also, there is a pressure grid point at the exact location of the grid to be analyzed. Therefore, a much lower setting of **GRIDWT** will usually be optimal. The actual value will take some experimentation but a value in the .1 to .001 may be appropriate. Note also that the value is grid dependent and must have as many values specified as NIGRIDS. In general, **GRIDWT** will become larger as the grid spacing of the ***RAMS*** grid increases.
GOBSEP real	If **IGRIDFL** (described above) is set to 2, a pressure data grid point will not be used in the objective analysis if it is within GOBSEP degrees of an upper air observation.
GOBRAD real	If **IGRIDFL** (described above) is set to 2, a pressure data grid point will not be used in the objective analysis if there are observations in three quadrants of a circle of **GOBRAD** radius (degrees). The circle is divided into quadrants by north-south and east-west diameters.
WVLNTH real array	In the Barnes objective analysis scheme, a degree of smoothing to be applied is determined from two parameters. The first, **WVLNTH**, is the wavelength of the data on the isentropic and σ_z surfaces (upper air) specified in kilometers to be retained. The second, **RESPON** (see below) is the fractional amplitude at which to retain that wavelength. Responses of other wavelengths are determined given a value of .3 set for the "gamma" parameter in the Barnes scheme. Note that **WVLNTH** is grid dependent and there must be as many values specified as **NIGRIDS**.
SWVLNTH real array	Similar to **WVLNTH** (described above) but pertains to surface rather than upper air data and analysis. It uses the same **RESPON** value as **WVLNTH**. Note that **SVLNTH** is grid dependent and there must be as many values specified as **NIGRIDS**. For the surface objective analysis. Note that it is grid dependent and there must be as many values specified as **NIGRIDS**.
RESPON real array	Percentage of specified wavelength amplitudes to be retained in both the upper air and surface analyses (see **WVLNTH** and **SWVLNTH** above). Note that **RESPON** is grid dependent and there must be as many values specified as **NIGRIDS**.

$DIGITALFILTER Namelist

The **$DIGITALFILTER** namelist provides a means for …

Variable name	Description
applyDigitalFilter boolean	Flag that controls digital filter feature activation. If **applyDigitalFilter = .**TRUE**.,** turns on digital filter and if **applyDigitalFilter = .**FALSE**.,** turns it off.
digitalFilterTimeWindow integer	Specifies the time, in seconds, that the model should run before applies the digital filter.

$METEOGRAM Namelist

The **$METEOGRAM** namelist provides a means for …

Variable name	Description
applyMeteogram boolean	Flag that controls digital filter feature activation. If **applyDigitalFilter = .**TRUE**.,** turns on digital filter and if **applyDigitalFilter = .**FALSE**.,** turns it off.
meteogramFrequency integer	Specifies the time, in seconds, that the model should write a meteogram result output.

meteogramMap: filename of special file that contains polygons representing cities or user defined areas.
string

meteogramDir: direcoty path to meteogramMap file.
string