RAMS

Regional Atmospheric Modeling System

Version 4.3/4.4

MODEL INPUT NAMELIST PARAMETERS

By Robert L. Walko and Craig J. Tremback

Table of Contents

Introduction

This document will describe the input namelist parameters for the RAMS atmospheric model 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 have the appearance of a standard FORTRAN namelist, and indeed they were patterned after the FORTRAN format. However, there have been several deficiencies in the FORTRAN standard, even in FORTRAN 90, concerning namelists. The biggest deficiency is a complete lack of, or different formats for, comments. Therefore, early in RAMS development, a replacement for the standard FORTRAN namelist reading capability was developed.

The atmospheric model component of RAMS contains 5 namelists, which have the names **$MODEL_GRIDS**, **$MODEL_FILE_INFO**, **$MODEL_OPTIONS**, **$MODEL_SOUND**, and **$MODEL_PRINT**, while the ISAN namelist are called **$ISAN_CONTROL**, and **$ISAN_ISENTROPIC**. These namelists are all contained in the data file called **RAMSIN** in the **etc** directory. This is a default name; the input namelist name can be set as a command line argument. Each namelist in this file begins with one of the above identifiers, and ends with the character string “$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. In a separate file, **rconfig.h**, in the **include** directory, the user sets values of 9 different parameters used in dimensioning the common blocks of the atmospheric model.

The following sections contain descriptions of all atmospheric model parameters and namelist variables. The descriptions include what the parameters and variables represent or control and provide acceptable or recommended settings for them. Cross-referencing to related variables provides additional information. For this reason, a complete read-through of all namelist variables is recommended. *Note as a matter of syntax that each value assigned to any namelist variable in the* ***RAMSIN*** *file must be followed by a comma.* An index of the configuration and namelist variables can be found at the end of this document.

Model configuration parameters - rconfig.h

The 9 parameters in the **rconfig.h** file are used as array dimensions for many variables in the atmospheric model global common blocks. For this reason, any time one of these parameters is changed, all files in the **model, isan**, and **revu** directories must be recompiled. The parameters are required for setting limits on common block array space. Often, the model run is well within those limits. 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 wasted. A suggestion is to set MAXGRDS=10, NXPMAX=200, NYPMAX=200, NZPMAX=150, NZGMAX = 20, and MAXSCLR = 150. The **rconfig.h** parameters are described in the following table.

MAXGRDS	maximum number of grids which 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	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	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	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	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	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	must be set to the largest of the values in NXPMAX, NYPMAX, NZPMAX+10, and NZGMAX.

MAXHP	maximum number of grid cells that overlap between the two hemispheric grids of a global simulation. Each hemispheric grid is square in polar stereographic space. The hemisphere of the earth projects onto the circle inscribed in the square. The four corner regions of the square extend into the opposite hemisphere. The points in these corner regions have their prognostic fields interpolated from values prognosed in the opposite hemispheric grid. The number of points in these corner regions depends on the grid spacing of each hemispheric grid. MAXHP should be set to at least 30% of the square of NNXP for grid 1 if NNXP is larger than 100, and at least 40% of the square of NNXP for smaller values of NNXP.
MAXMACH	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.

$MODEL_GRIDS Namelist

The **$MODEL_GRIDS** namelist provides information to the model primarily on the structure of the one or more nested grids used in a simulation, including location, mesh size, number of mesh points, spatial nesting relationships, time step length, and time and duration of the run. Most of these variables are arrays dimensioned to the parameter MAXGRDS, which is set in the file **rconfig.h**. Each value in the array applies to a different grid, several of which may be activated if the user wishes to employ grid nesting. In the **$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.

Variable name	Data 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.
VTABCUST	character	variable whose value is compared by the RAMS code against the value of a character string read from the file VTABLE. If the values are the same, the RAMS run will proceed, but if they are different, the run will output a message stating so and will stop. VTABLE is a data file which tells RAMS how each field variable in the "A" array will be used. The purpose of the VTABCUST is to help the user prevent accidentally using the wrong VTABLE file, in case a user has made a nonstandard version. Occasionally a user needs to modify VTABLE, for example, to add a new field variable or to instruct RAMS to output a variable to an analysis file that is not normally output. The value of the character string in VTABLE serves as an identifier for the specific version of VTABLE and should be changed for any new version made. Then, VTABCUST should be set in RAMSIN to the value on the desired VTABLE.
RUNTYPE	character	variable specifying one of six ways in which the model is to be run. In many situations, a RUNTYPE of '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, and ISOILFLG, 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 be defined 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. Setting RUNTYPE to ‘MAKELAND’ will process only the surface files (containing topography, soil textural class, vegetation type, and subgrid distribution). Setting RUNTYPE to ‘MAKESST’ will process only the SST files (containing sea surface temperatures). If RUNTYPE is either ‘MAKESFC’, ‘MAKELAND’, or ‘MAKESST’, 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 set to 1 in this case), or are read from a varfile prepared in an earlier ISAN run (INITIAL set to 2). 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. We take care here to define the terms *simulation* and *run*. A *simulation* is an entire integration of the atmospheric model from initial conditions to a 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.
IMONTH1	integer	month of the year when the simulation begins. It is used in conjunction with the namelist parameters IYEAR1, IDATE1, and IHOUR1 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 and IMONTH1 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 IHOUR1 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.
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 LEAF2, 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 LEAF2 is used.
NZS	integer	maximum number of snowcover layers allowed to occur in a

		simulation. It is relevant only when the LEAF2 submodel is activated In RAMS, NZS must be at least 1, and has no upper limit other than a practical one. If no snowcover is expected in a simulation, there is no need for NXS to be greater than 1. 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. If RAMS is run on a global domain, it uses two coarse grids each covering just over a hemisphere, and thus requires two grids to have NXTNEST=0. In fact, setting two NXTNEST values to zero is the way to specify that the model is to be run globally. GRID 1 must be one of the coarse grids, but the other may be numbered 2, 3, or any larger integer that obeys the above rule.
IHTRAN	integer	type of horizontal grid transformation to be used in a model simulation. If it is set to 0, the 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. Setting IHTRAN to 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 which 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. If RAMS is run on a global domain, the value of DELTAX specified in the namelist is ignored, and the model instead computes DELTAX such that grid 1 projects onto a hemisphere of the earth, given the specified value for NNXP. In order to approximately compute what spacing the model will use for the global domain, divide 26000 km by the value of NNXP for grid 1.
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. If RAMS is run on a global domain, the value of DELTAY specified in the namelist is ignored, and the model instead computes DELTAY such that grid 1 projects onto a hemisphere of the earth, given the specified value for NNYP. In order to approximately compute what spacing the model will use for the global domain, divide 26000 km by the value of NNYP for grid 1.
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	similar to NSTRATX, but applies to the grid resolution ratio in the

	array	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	grid-dependent parameter used to control east-west movement of a nested grid. The value assigned to it will be the velocity in meters per second that the grid will be moved. The velocity specified for each grid is always with respect to grid 1 (which cannot move) or equivalently to the fixed ground coordinates. This model feature is used to keep a nested grid surrounding a feature of interest, such as a thunderstorm, which is traveling. For example, if at some time it is found that the storm is centered within grid 3, but the storm is traveling westward at 20 m/s and will soon reach the western boundary of grid 3, GRIDU for grid 3 can be set to -20 m/s which will keep the storm centered within the grid. When a nested grid moves, it does so in discrete horizontal distance increments equal to the grid spacing on the next coarser mesh in which the moving grid is nested. This is done, because the allowable positions the moving grid may have within the coarser parent grid are themselves discrete. When a grid moves, its interior prognostic variables are simply shifted toward the grid's trailing edge the correct number of points (equal to NSTRATX for that grid), the values near the trailing edge are discarded, and the values near the leading edge are newly interpolated from the coarser parent grid. Special care must be taken by the user to make sure that grid 3 will not itself move too close to the boundary of the coarser grid in which it is nested. One remedy if this happens would be to simultaneously move the coarser grid too, as long as it is not grid 1. Set GRIDU to 0 to not activate grid movement. An alternative to moving grids, which has been widely used but which is not as widely applicable, is to Doppler shift the meteorological features by adding a constant wind vector to all wind values in the domain. However, this alternative cannot be used where the ground surface is not horizontally homogeneous. In using the moving grid feature, it is not permissible, due to requirements of the terrain-following coordinate transformation, to continually fill topography from the standard RAMS files on any of the moving grids or any of the grids nested within them. These fields on moving grids must be interpolated from the respective parent grids instead. Thus, in the namelist, set ITOPTFLG to zero for moving grids. Also, in order to make sure the interpolated values of topography and other surface fields are not overwritten by assignments in **ruser.f90** each time a grid moves, comment out the sections of all relevant IF blocks in **ruser.f90** that fill these values. Moving nested grids are not currently implemented for model runs using parallel processing.
GRIDV	real array	similar to GRIDU, but applies to north-south movement of a nested grid.

$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.

Variable name	Data type	Description
INITIAL	integer	flag specifying how atmospheric fields in the model are to be initialized. If INITIAL is set to 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. If INITIAL is set to 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. Further details regarding the use of variable initialization are given in descriptions of several variables in this Section.
VARFPFX	character	prefix that begins the names of all initialization files, also called varfiles, (including path names if applicable) to be read into the model, for cases when the INITIAL variable is set to 2. These files are output from a previous model run with RUNTYPE set to 'MAKEVFILE'. (See IFNVAR) If INITIAL is set to 1, this parameter is ignored.
VWAIT1	real	used only for automated operational model runs in which an attempt is made to acquire initialization data as input to ISAN. If this attempt is unsuccessful and a varfile has not yet been generated from ISAN, the model is intructed to wait a time interval given by VWAIT1 (in 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.
NUDLAT **TNUDLAT** **TNUDCENT** **TNUDTOP** **ZNUDTOP**	integer real real real real	parameters that control the option of four dimensional data assimilation (4DDA) by Newtonian relaxation (nudging). This 4DDA option requires that the INITIAL variable is set to 2, so that the model is initialized from varfiles. The varfiles 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.
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. The character string placed in this variable must be the actual name of the history 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 and HFUNITS .
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.
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 which 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 4.3 or 4.4.
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 (versions 4.3 or 4.4) 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.
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 applits 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 which 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** which 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	***Not currently implemented -*** 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 (At the time of this writing, these files are not yet prepared; until they are available, avoid ISOILFLG = 1.) If ISOILFLG = 2, soil textural class for the grid is filled horizontally homogeneously to the value specified by NSLCON. As for for vegetation class values described for IVEGTFLG, soil textural class 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.
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).
ITOPTFN	character array	grid-dependent character variable specifying the family name of the terrain height data file(s), with path names if applicable, to be read in conjunction with namelist variable ITOPTFLG set to 1. This variable is not the entire file name, but is only set to the prefix denoting the set of files. These 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 files that contain 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 file containing vegetation type data for initializing each grid. It is similar to namelist variable ITOPTFN (see ITOPTFN and IVEGTFLG).
ISOILFN	character array	grid dependent character variable in which the file name prefix (and pathname if applicable) is specified for the soil textural class files to be input.. (At the time of this writing, these files are not yet prepared.)
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 which 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 perserve 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.
IZOFLG	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 IZOFLG 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 IZOFLG 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 ZOMAX). The inflence 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 ZOFACT).
ZOMAX	real array	Upper bound (in meters) imposed on surface roughness height that is evaluated from subgrid topography (see IZOFLG and ZOMAX).
ZOFACT	real array	Roughness factor used in computing surface roughness height from subgrid topography (see IZOFLG and ZOMAX).
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.
COLTABFN	character	is a single-valued character variable giving the name (and path, if any) of the *collection table*, described above.

$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.

Variable name	Data type	Description
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.
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). 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 for 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. It is, however, still the most computationally expensive scheme, although work is planned to improve its performance. (See also ILWRTYP and RADFRQ .)
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.
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	grid-dependent flag used to control whether the convective parameterization is to be activated or not. A value of 1 for this flag activates the parameterization, and a value of 0 leaves 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. Unfortunately, the convective parameterization schemes currently available assume the grid size in the horizontal to be around 20 kilometers or greater. This means that convective parameterization may be activated on any grid of this resolution, but that at resolutions between about 2 and 20 kilometers, no adequate convective adjustment scheme exists.
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 to 1 for a particular grid.
WCLDBS	real	used 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.
NPATCH	integer	number of subgrid patches used in LEAF2, 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	flag specifying which of 2 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 LEAF2 parameterization available in the model. A value of 1 for ISFCL activates the LEAF2 submodel (described in Section 1c). LEAF2 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.)

NVGCON	integer	flag specifying the landuse class to be used in the LEAF-2 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-2 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 LEAF2 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 Kelvins 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.
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** .
IDIFFK	integer array	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.
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.
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. Setting NLEVEL to 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. A value of 1 for NLEVEL 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. NLEVEL equal to 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. Setting NLEVEL to 3 activates

		the bulk microphysics parameterization, which includes cloud water, rain, pristine ice, snow, aggregates, graupel, and hail, or certain subsets of these as specified by ICLOUD, IRAIN, IPRIS, ISNOW, IAGGR, IGRAUP, and IHAIL. This parameterization includes the precipitation process.
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 either 5 or 7. ICLOUD = 5 nucleates cloud droplets from a constant concentration of cloud condensation nuclei (CCN) that is specified by the user in CPARM (see below) in units of number per kilogram of air. This is roughly equal to number per cubic meter in the lower troposphere but is less than this at higher levels. When ICLOUD is set to 7, a three-dimensional prognostic CCN field is activated in RAMS, and cloud droplet number is computed from nucleation from this field as a function of CCN and environmental properties. The CCN field is initialized by default from the concentration specified in CPARM, but evolves in time afterward. There is currently (in RAMS version 4.4) no scavenging removal or source functions of CCN, but there are advective and diffusive transport. Further development of CCN prediction in RAMS is in progress. ICLOUD is relevant only when LEVEL is set to 3.
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. Additional values of 6 or 7 are valiid for IPRIS and control number concentration prediction due to heterogeneous nucleation. If IPRIS = 5, RAMS assumes that ice forming nuclei (IFN) have a constant concentration in terms of number per kilogram of air (implying a reduction with height in the number per cubic meter). If IPRIS = 6, a horizontally homogeneous and constant-in-time vertical profile of IFN is assumed where number per kilogram of air decreases with height. If IPRIS = 7, a prognostic IFN field is activated in RAMS, and the current local predicted IFN concentration in each grid cell is used in determining the number of ice crystals nucleated.
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.
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 CCN in number per kilogram of air. If ICLOUD is

		set to 7, a prognostic cloud concentration nuclei (CCN) field is activated in RAMS, and CPARM then represents the initial value (in number per kilogram of air) of the CCN concentration field.
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 since pristine ice number concentration must always be prognosed (if pristine ice is active).
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.
GNU	real array	parameter with 7 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	Data type	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	Data type	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.

ISAN Configuration Parameters – ascoms.h

The following configuration parameters are defined in the **ascoms.h** file. 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	The maximum number of vertical levels that can be used in the pressure data.
MAXISN	The maximum number of vertical levels that can be used in the isentropic analysis.
MAXX	The maximum number of grid points (in the "x" or east-west direction) in any of the RAMS or pressure grids.
MAXY	The maximum number of grid points (in the "y" or north-south direction) in any of the RAMS or pressure grids.
MAXTIMES	The maximum number of data analysis times that can be processed in a single run.
MAXAGRDS	The maximum number of RAMS grids that can have varfiles generated.
MAXSIGZ	The maximum number of vertical levels that can be used in the σ_z analysis.
MAXLEV	The maximum number of levels in an input rawinsonde.

$ISAN_CONTROL Namelist

	Data type	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, $\sigma_{z,}$ 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/σ_z Stage: $ISAN_ISENTROPIC Namelist

Variable name	Data type	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.
NONLYS	integer	sometimes just one or a few stations are accessed, for instance, to print a sounding. In cases where not all stations are to be accessed, this is the number of stations to access. If NONLYS is set to 0, then all stations found in the domain are accessed. Since the surface station information is not very long, all of these are printed by default. Therefore, NONLYS only refers to the rawinsondes.
IDONLYS	character array	is a character array of station IDs to access in the case where NONLYS (see above) is not set to zero. The number of entries defined for IDONLYS should equal the value specified for NONLYS.
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 which assumes a 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.
ISTAPLT	integer	flag of value 0(no) or 1(yes) indicating whether to call the graphical routines to plot skew-T's of the rawinsondes. If this is set to 1, the code will only plot the rawinsondes and surface observations and will not do any objective analysis. **Note: this flag is unsupported in version 4.3 and is no longer present in version 4.4. See comment at the beginning of the next section: ISAN Graphical Stage.**
ISTAREP	integer	flag of value 0(no) or 1(yes) indicating whether to print rawinsonde values. If this is set to 1, the code will only access and print the

		observations and will not do any objective analysis.
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.

ISAN Graphical Stage: $ISAN_GRAPH Namelist

The capability of plotting directly from ISAN is marked for obsolescence. It is planned to be replaced with a post-processing utility (such as GrADS) that will work directly from the intermediate isentropic/σ_z files. While the namelist variables described in this section remain in version 4.3, they have been removed entirely from version 4.4. Their use is not recommended and has not been supported since version 3b.

Variable name	Data type	Description
IPLTPRS	integer	main switch for plotting pressure data. 0 - Don’t plot anything from the pressure stage. 1 - Plot things from the pressure stage.
IPLTISN	integer	main switch for plotting isentropic data. 0 - Don’t plot anything from the isentropic stage. 1 - Plot things from the isentropic stage.
IPLTSIG	integer	main switch for plotting σ_z data. 0 - Don’t plot σ_z data. 1 - Plot σ_z data
IPLTSTA	integer	main switch for plotting station values. These are horizontal plots with values of variables placed at the station locations. Only the surface and isentropic level values may be plotted at this time. 0 – Don't plot station data. 1 - Plot station data.
ILFT1I	integer	grid point value that will be the boundary window on the western (left) side of the plots. If any of these window values are set to 0, the window defaults to the entire domain.
IRGT1I	integer	grid point value that will be the boundary window on the eastern (right) side of the plots. If any of these window values are set to 0, the window defaults to the entire domain.
IBOT1J	integer	grid point value that will be the boundary window on the southern (bottom) side of the plots. If any of these window values are set to 0, the window defaults to the entire domain.
ITOP1J	integer	grid point value that will be the boundary window on the northern (top) side of the plots. If any of these window values are set to 0, the window defaults to the entire domain.
NPLEV	integer	number of pressure levels to plot.
IPLEV	integer array	pressure levels to be plotted. The number of values specified should equal the value assigned to NPLEV and the pressure level values specified must be also specified for LEVPR (described above).
NFLDU1	integer array	number of fields to be plotted on each pressure level. Each field will be on a separate frame. The total number of frames for the pressure plots is then NPLEV*NFLDU1.
IFLDU1	character array	field names to plot. The number of character strings specified must equal the value assigned for NFLDU1 (described above). The field names are listed at the end of this chapter.

CONU1	real array	contour intervals for each field. If set to 0., a default value will be chosen by CONREC. The number of values specified must equal the value assigned for NFLDU1 (described above).
IVELU1	integer array	velocity vector flags specifying if vectors should be overlayed on the contoured fields. The number of values specified must equal the value assigned for NFLDU1 (described above). 0 - No vectors should be drawn. 1 - Vectors should be drawn at every grid point. 2 - Vectors should be drawn at every other grid point. 3 - Vectors should be drawn at every third grid point.
ILFT3I	integer	grid point value that will be the boundary window on the western (left) side of the plots. If any of these window values are set to 0, the window defaults to the entire domain.
IRGT3I	integer	grid point value that will be the boundary window on the eastern (right) side of the plots. If any of these window values are set to 0, the window defaults to the entire domain.
IBOT3J	integer	grid point value that will be the boundary window on the southern (bottom) side of the plots. If any of these window values are set to 0, the window defaults to the entire domain.
ITOP3J	integer	Grid point value that will be the boundary window on the northern (top) side of the plots. If any of these window values are set to 0, the window defaults to the entire domain.
IUP3BEG	integer	lowest isentropic level to plot.
IUP3END	integer	uppermost isentropic level to plot.
IUP3INC	integer	increment of isentropic levels to plot.
NFLDU3	integer	number of fields to be plotted on each isentropic level. Each field will be on a separate frame.
IFLDU3	character array	field names to plot. The number of character strings specified must equal the value assigned for NFLDU3 (described above). The field names are listed at the end of this chapter.
CONU3	real array	contour intervals for each field. If set to 0., a default value will be chosen by CONREC. The number of values specified must equal the value assigned for NFLDU3 (described above).
IVELU3	integer array	velocity vector flags specifying if vectors should be overlayed on the contoured fields. The number of values specified must equal the value assigned for NFLDU3 (described above). 0 - No vectors should be drawn. 1 - Vectors should be drawn at every grid point. 2 - Vectors should be drawn at every other grid point. 3 - Vectors should be drawn at every third grid point.
		Variables controlling surface plots: The surface plots use the windowing information from the isentropic plots (ILFT3I, IRGT3I, IBOT3J, ITOP3J).
NFLDS3	integer	number of fields to be plotted on the surface. Each field will be on a separate frame.
IFLDS3	character array	character array of field names to plot. The number of character strings specified must equal the value assigned for NFLDS3 (described above). The field names are listed at the end of this chapter.
CONS3	real array	real array of contour intervals for each field. If set to 0., a default value will be chosen by CONREC. The number of values specified must equal the value assigned for NFLDS3 (described above).
IVELS3	integer array	velocity vector flags specifying if vectors should be overlayed on the contoured fields. The number of values specified must equal the value assigned for NFLDS3 (described above). 0 - No vectors should be drawn. 1 - Vectors should be drawn at every grid point.

		2 - Vectors should be drawn at every other grid point. 3 - Vectors should be drawn at every third grid point.
		Variables controlling σ_z plots: The plots use the windowing information from the isentropic plots (ILFT3I, IRGT3I, IBOT3J, ITOP3J).
ISZBEG	integer	index of the lowest σ_z level to plot. These levels pertain to RAMS grid levels.
ISZEND	integer	index of the uppermost σ_z level to plot.
ISZINC	integer	increment of integer indexes of σ_z levels to plot.
NFLDSZ	integer	number of fields to be plotted on each σ_z level. Each field will be on a separate frame.
IFLDSZ	character array	field names to plot. The number of character strings specified must equal the value assigned for NFLDSZ (described above). The field names are listed at the end of this chapter.
CONSZ	real array	contour intervals for each field. If set to 0., a default value will be chosen by CONREC. The number of values specified must equal the value assigned for NFLDSZ (described above).
IVELSZ	integer array	velocity vector flags specifying if vectors should be overlayed on the contoured fields. The number of values specified must equal the value assigned for NFLDSZ (described above). 0 - No vectors should be drawn. 1 - Vectors should be drawn at every grid point. 2 - Vectors should be drawn at every other grid point. 3 - Vectors should be drawn at every third grid point.
		Variables controlling station plots: The surface plots use the windowing information from the isentropic plots (ILFT3I, IRGT3I, IBOT3J, ITOP3J).
NPLTRAW	integer	number of rawinsonde plots per page. As each rawinsonde is accessed, a two panel graph is made of the basic variables (u, v, potential temperature, relative humidity). The purpose of these plots is to check for bad data in the report. NPLTRAW is a target number of how many rawinsondes to graph on each frame. For instance, a value of 16 will plot a 4 by 4 array of 16 rawinsondes.
NSTIS3	integer	number of fields to be plotted from the station data. Each field will be on a separate frame.
ISTIS3	character array	field names to plot. The number of values specified must equal the value assigned for NSTIS3 (described above). The field names are listed at the end of this chapter
		Variables controlling vertical cross-sections:
NCROSS3	integer	Number of cross-sections to plot.
		The following 4 variables must have as many values specified as the value assigned to NCROSS3:
ICRTYP3	integer array	cross-section type: 1 - East-west cross-section. 2 - North-south cross-section.
ICRA3	integer array	grid point values of the left boundary.
ICRB3	integer array	grid point values of the right boundary.
ICRL3	integer array	grid point values of the cross-section locations.
NCRFLD3	integer array	number of fields to plot on each cross-section. There will be NCROSS3*NCRFLD3 frames each with isentropes and another field contoured.
ICRFLD3	character array	field names to plot. The field names are listed at the end of this chapter. The number of character strings specified must equal the value assigned to NCRFLD3 (described above).

THCON3	real array	contour increment of the isentropes. The number of values specified must equal the value assigned to NCRFLD3 (described above).
ACON3	real array	contour increment of the other field. The number of values specified must equal the value assigned to NCRFLD3 (described above).

Field names for the graphical stage

Pressure plots:	
U	East-west wind component
V	North-south wind component
TEMP	Temperature
GEO	Geopotential (height)
RELHUM	Relative humidity
MIXRAT	Vapor mixing ratio
THETA	Potential temperature
SPEED	Wind speed
ENERGY	Moist static energy
THETAE	Equivalent potential temperature
SPRESS	Sea-level pressure

Isentropic plots:	
U	East-west wind component
V	North-south wind component
PRESS	Pressure
MONTSTR	Montgomery streamfunction
RELHUM	Relative humidity
THETA	Potential temperature
SPRESS	Sea-level pressure

Surface station plots:	
U	East-west wind component
V	North-south wind component
PRESS	Pressure
RELHUM	Relative humidity
MIXRAT	Vapor mixing ratio
TEMP	Temperature

σ_z plots:	
U	East-west wind component
V	North-south wind component
PRESS	Pressure
THETA	Potential temperature
RELHUM	Relative humidity
MIXRAT	Vapor mixing ratio
TEMP	Temperature
SPEED	Wind speed
ENERGY	Moist static energy
THETAE	Equivalent potential temperature