# REVU

## RAMS/HYPACT Evaluation and Visualization Utilities

Version 2.5

# **User's Guide**

December 2004

*by*

Craig J. Tremback

Robert L. Walko

Martin J. Bell

ATMET LLC

ATmospheric, Meteorological, and Environmental Technologies

P.O. Box 19195

Boulder, CO 80308-2195

# *REVU User's Guide*

## *Contents*

# *About REVU*

This chapter describes the RAMS/HYPACT Evaluation and Visualization Utilities (***REVU***), which is the standard supported package for generating graphical representations and reformatting ***RAMS*** model output. ***REVU***'s function is to read "analysis" files written from a ***RAMS*** or ***HYPACT*** simulation, select user-specified fields and cross sections from the file data, and plot the field cross sections, or output the selected data in one of several available formats (e.g. Vis5D, GrADS, GRIB). ***REVU*** utilizes NCAR Graphics to perform most of the plotting functions. ***REVU*** can also interpolate point data from the analysis files.

Similar to ***RAMS***, ***REVU*** is in a state of continual evolution. This document is the fourth edition of the ***REVU*** User's Guide since it was redeveloped from ***VAN*** and split from the ***RAMS*** version 3b User's Guide. It describes ***REVU*** as of March 2004. ***REVU*** v2.5 is compatible with the new ***RAMS*** v5.x and contains the same basic functionality as ***REVU*** 2.3.1. UNIX/Linux make commands and a system of make files and makefile includes are used for compiling the code. The make files and build procedure are detailed in a separate document.

Execution of ***REVU*** is controlled by a set of namelist variables usually contained in a file named *REVU_IN*. In order to operate ***REVU***, the user should be acquainted with the two primary means of setting parameters that control its functions. These are:

1. The various configuration parameters contained in *vcomm2.h,* which define several array dimensions controlling the capacity of the model.
2. The variables in the *REVU_IN* namelist file.

The following sections of this document describe the function and use of each of the parameters and namelist variables, and how to set appropriate values for them.

This will be the last version of this series of ***REVU***. Development is progressing on a completely new ***REVU***, in which the various functionalities are divided into different applications.

# *REVU Configuration Parameters*

In *vcomm2.h*, there are a few parameters that are set to specify maximum dimensions of various arrays. These generally do not need to be modified except in exceptional situations.

| | |
|---|---|
| **MAXFORE**<br>integer | The maximum number of variables to plot or output (default set to 30). |
| **MAXFILS**<br>integer | The maximum number of input files (default set to 200). |
| **MAXLOC**<br>integer | The maximum number of observation locations (default set to 1000). |
| **MAXLEV**<br>integer | The maximum number of levels in a profile (default set to500). |

# *REVU_IN Namelists*

The namelist file *REVU_IN* contains four namelists, with the names **$CONTROL**, **$GRAB**, **$GLL** and **$STATS**. The **$CONTROL** namelist is required for all runs and specifies the general data extraction scenario and all plotting parameters if **ANATYPE** = 'SPACE'. The remaining namelists all control some portion of the non-plotting **ANATYPE** settings and are not always required.

## *$CONTROL Namelist*

In the following documentation a background or frame refers to the plotting or output of each **CFRAME_A**, **CFRAME_B** and **CFRAME_C** element, except in the case where multiple panels are plotted into a single frame, as specified by **IPANEL**.

| | |
|---|---|
| **ANPREF**<br>charachter | The UNIX path name and prefix of the names of ***RAMS*** or ***HYPACT*** analysis files to be read and used to drive ***REVU*** (up to 128 characters). Note that the path name is optional and can be either relative or absolute (no path is equivalent to './').<br>Depending on what was output by RAMS or HYPACT, ANPREF could be:<br>• <prefix>-A-<date> Standard ***RAMS*** analysis files<br>• <prefix>-L-<date> Lite ***RAMS*** analysis files<br>• <prefix>-M-<date> Averaged ***RAMS*** analysis files<br>• <prefix>-B-<date> Lite averaged ***RAMS*** analysis files<br>• <prefix>-C-<date> ***HYPACT*** analysis files<br><prefix> is the directory location and file prefix given by **AFILOUT** or **HYPPREF**, the ***RAMS*** and ***HYPACT*** output files name prefixes respectively.<br><date> is the output filename date string in which UNIX wild cards can be used, for example:<br><prefix>-A* will give all standard ***RAMS*** analysis files<br><prefix>-A-2000-10-??-000000* will give the analysis files for 0Z each day<br>***REVU*** finds all files in the specified directory that have that match **ANPREF** (it looks for the header files by listing **ANPREF***-head.txt) and determines their chronological order from the time suffix encoded into the filenames. The user selects some or all of these files to be plotted in the namelist variable **TVAR**. |
| **REVPREF**<br>charachter | The UNIX path name and prefix for the ***REVU*** output files (up to 128 characters). Note that the path is optional and can be either relative or absolute (no path is equivalent to './'). |

| | |
|---|---|
| **ANATYPE**<br>character | Designates what type of ***REVU*** run to make. Options are:<br>• SPACE — Produce NCAR Graphics spatial axis plots of the fields specified in **CFRAME_A**, **CFRAME_B** and **CFRAME_C**. No time axis plots are available in the current version.<br>• V5D — Create Vis5D files of the fields specified in **CFRAME_A**.<br>• GRADS — Create GrADS files of the fields specified in **CFRAME_A**.<br>• GRIB — Create GRIB files of the fields specified in **CFRAME_A**.<br>• DUMP — Dump the fields specified in **CFRAME_A** in a user defined format. The format is specified in *dumpout.f90* module, which may be modified by the user. The purpose of this option is to extract fields from analysis files that are written in packed format.<br>• GRAB — Output the fields specified in **CFRAME_A** at the points specified by latitude-longitude-height locations or vertical profiles at latitude-longitude locations in RALPH2 or a user defined format. The format is specified in *grabstat.f90* module, which may be modified by the user.<br>• STATS — Do statistical comparisons between ***RAMS*** data and observations. |
| **HEAD1**<br>character | A 24 character long string used to title all plots. |
| **IGRID**<br>integer array[†] | A background dependant parameter that specifies which grid is to be processed.<br>• If set to zero, all grids will be processed, on separate backgrounds if **ANATYPE** is set to 'SPACE'.<br>• If set to > 0, only the specified grid will be processed.<br>• If set to < 0, all grids finer than abs(IGRID) will be processed, on separate backgrounds if **ANATYPE** is set to 'SPACE'.<br>If **CFRAME_A**, **CFRAME_B** and **CFRAME_C** indicate multiple backgrounds and **IGRID** is not specified for a background, then the **IGRID** for the first background will be used for the unspecified background (i.e. in most cases you only need to specify **IGRID** for the first background). |
| **IZTRAN**<br>integer array[†] | A background dependant parameter that, if plotting a horizontal cross-section (see **ZVAR**, **XVAR**, and **YVAR**), determines the vertical coordinate.<br>• If set to 1, fields are output on the model's terrain-following coordinate surfaces.<br>• If set to 2, fields are interpolated to horizontal (Cartesian) surfaces which have the same heights as the terrain-following heights of a grid point at sea level.<br>• If set to 3, fields are interpolated to pressure surfaces. If plotting, the surface output is determined by **IPLEVEL**, otherwise the vertical extent of the field output is determined by **ZVAR**, **XVAR**, and **YVAR** acting on the standard pressure levels 1000, 925, 850, 700, 600, 500, 400, 300, 200 and 100 mb.<br>If **CFRAME_A**, **CFRAME_B** and **CFRAME_C** indicate multiple backgrounds and **IZTRAN** is not specified for a background, then the **IZTRAN** for the first background will be used for the unspecified background (i.e. in most cases you only need to specify **IZTRAN** for the first background). |

| | |
|---|---|
| **IPLEVEL**<br>integer array‡ | A background dependant parameter that, if plotting a horizontal cross-section (see **ZVAR**, **XVAR**, and **YVAR**) and **IZTRAN** is set to 3, **IPLEVEL** denotes a pressure surface (in mb) on which to generate the plot. Only standard pressure levels 1000, 925, 850, 700, 600, 500, 400, 300, 200 and 100 mb may be specified.<br>If **CFRAME_A**, **CFRAME_B** and **CFRAME_C** indicate multiple backgrounds and **IPLEVEL** is not specified for a background, then the **IPLEVEL** for the first background will be used for the unspecified background (i.e. in most cases you only need to specify **IPLEVEL** for the first background). |
| **MAPFILL**<br>integer | If plotting a horizontal cross-section (see **ZVAR**, **XVAR**, and **YVAR**), specifies whether a map projection will be plotted and if so, whether it will be color-filled.<br>• If set to 0, no map will be drawn.<br>• If set to 1, draw a map outline (filled contours and tiles will be drawn on top on the map).<br>• If set to 2, draw a map outline and fills the land in green and water in blue.<br>• If set to 3, as with 2, except the map outline will be drawn on top of all filled contours and tiles.<br>• If set to <0, as with any setting of **MAPFILL**>0, but with the map outline drawn in a highlighted shadow (enhances the geographic boundaries).<br>Users can modify the *mkmap.f90* module to draw other geographic and demographic information (the user will also need to supply the data). |
| **IBACKGND**<br>integer | If plotting, specifies the plot background color.<br>• If set to 1, the plot background is set to black and the foreground to white. This setting is intended for display on electronic media.<br>• If set to 2: the plot background is set to white and the foreground to black. This setting is intended for display on electronic media.<br>• If set to 3: the plot background is set to white and the foreground to black. This setting is intended for display on white paper.<br>• If set to <0, as with any setting of **IBACKGND**>0, but all colors are set to the foreground color (titles, plot scales, etc) and grayscale (map fills, filled contours and tiles).<br>Users can modify the *rcolors.f90* module to specify their own color schemes (especially relevant to setting up the color tables for printing on specific printers), noting that some user color customization is available with **COLORS** without any modifications to the code. |
| **IPLTINFO**<br>integer | If plotting, specifies whether to draw the plot information table that appears at the bottom of each plot.<br>• If set to 0, do not draw the information table (this maximizes the plot space for the actual plot)<br>• If set to 1, draw the information table.<br>• If set to 2, draw the reduced plot information (not in table). |

| | |
|---|---|
| **IPANEL**<br>integer | If plotting, specifies the number of plots drawn per frame (1 to 4).<br>• If set to 1, one plot is drawn on the full frame and the number of frames drawn equals the number of plots specified by **CFRAME_A**, **CFRAME_B** and **CFRAME_C**.<br>• If set to 2, 3 or 4, that number of plots are drawn per frame and the number of frames drawn equals the number of plots specified by **CFRAME_A**, **CFRAME_B** and **CFRAME_C** divided by the IPANEL setting. The size of each plot is one quarter the size of the plots when IPANEL is set to 1. Also, a reduced set of plot information is included for each plot and the axis appear without any labels. You can use this setting to create classic four panel plots.<br>• If set to 0, not plotting is done. Instead a series of tables are output with the colors that are going to be used in the first frame (i.e. taking into account the settings of **CFRAME_A**(1), **IBACKGND** and **COLORS**). |
| **LANDMARK**<br>character array[†] | If plotting a horizontal cross-section (see **ZVAR**, **XVAR**, and **YVAR**), controls the drawing of the landmarks specified in the *LANDMARKS* file. **LANDMARK** is a text string that has two slash delimited components, the first controlling the drawing of the markers and the second controlling the labeling of the markers. Each of these has further colon-delimited components. The maker component:<br>• **M** component (integer): Marker control (M0 does not mark the landmarks and M1 marks the landmarks).<br>• **b** component (real): Landmark separation or buffer - min separation between markers (b.03 gives a spacious separation and b.01 gives a tight packing).<br>• **t** component (integer): Controls the marker type (t1: dot, t2: plus, t3: asterisk, t4: circle, t5: cross).<br>• **s** component (real): Controls the scale of the markers (s1. is a good setting).<br>• **x** component (character string): The color of the marker. This can be any in the color table (below), or any user defined color specified by **COLORS**.<br>The Label components:<br>• **L** component (integer): Label control (L0 does not label the landmarks and L1 labels the landmarks).<br>• **s** component (real): Controls the size of the label characters (s.01 is a good setting).<br>• **x** component (character string): The color of the label. This can be any in the color table (below), or any user defined color specified by **COLORS**.<br>For example, a **LANDMARK** setting of<br>`LANDMARK(1)='/M1:b.03:t2:s1.:xyellow/L1:s.01:xred/',`<br>Draws yellow + makers and red labels with sufficient buffer as to avoid a clutter of markers on the plots.<br>If **CFRAME_A**, **CFRAME_B** and **CFRAME_C** indicate multiple backgrounds and **LANDMARK** is not specified for a background, then the **LANDMARK** for the first background will be used for the unspecified background (i.e. in most cases you only need to specify **LANDMARK** for the first background). |

**COLORS**
character array[†]

If plotting, specifies the re-mapping of colors in the color table. **COLORS** is a slash delimited text string. Each slash delimited component has a further two or four colon delimited components as illustrated below:

```
COLORS(1)='/axis:yellow/title0:rgb:0.0:0.4:0.0/'
```

In the first example of **COLORS(1)**, the color table element 'axis' (defined below) is redefined from its default of white on a black background, or back on a white background, to the color table element yellow (also defined below).

In the second example of **COLORS(1)**, the color table element 'title0' (defined below) is redefined in red-green-blue (RGB) color space as a dark green, a shade of green not previously defined in the color table.

```
COLORS(2)='/red:rgb:0.8:0.0:0.0/dkred:rgb:1.0:0.0:0.0/'
```

In the first example of **COLORS(2)**, the color table element 'red' (defined below) is redefined from its default RBG value to a slightly darker shade of red.

In the second example of **COLORS(2)**, a new color table element, 'dkred', defined in RGB color space.

You can redefine up to around 8 colors (the string has a maximum length of 128 characters) in either RGB of HLS (hue-lightness-saturation) color space, each as a slash delimited item.

Background independent colors:

| *Color* | *RGB Values* | | | *Color* | *RGB Values* | | |
|---|---|---|---|---|---|---|---|
| white | 1.0 | 1.0 | 1.0 | purple | 1.0 | 0.0 | 1.0 |
| black | 0.0 | 0.0 | 0.0 | cyan | 0.0 | 1.0 | 1.0 |
| grayblack | 0.3 | 0.3 | 0.3 | tan | 0.86 | 0.58 | 0.44 |
| darkgray | 0.65 | 0.65 | 0.65 | sienna | 0.63 | 0.32 | 0.18 |
| gray | 0.5 | 0.5 | 0.5 | brown | 0.65 | 0.16 | 0.16 |
| lightgray | 0.8 | 0.8 | 0.8 | orangered | 1.0 | 0.0 | 0.2 |
| darkred | 0.5 | 0.0 | 0.0 | orange | 1.0 | 0.5 | 0.0 |
| red | 1.0 | 0.0 | 0.0 | gold | 1.0 | 0.85 | 0.0 |
| midred | 1.0 | 0.65 | 0.65 | greenyellow | 0.70 | 1.0 | 0.2 |
| lightred | 1.0 | 0.8 | 0.8 | forestgreen | 0.14 | 0.56 | 0.14 |
| darkgreen | 0.0 | 0.5 | 0.0 | aqua | 0.1 | 1.0 | 0.85 |
| green | 0.0 | 1.0 | 0.0 | deepskyblue | 0.0 | 0.75 | 1.0 |
| midgreen | 0.65 | 1.0 | 0.65 | skyblue | 0.2 | 0.56 | 0.8 |
| lightgreen | 0.8 | 1.0 | 0.8 | royalblue | 0.25 | 0.45 | 0.95 |
| darkblue | 0.0 | 0.0 | 0.5 | slateblue | 0.4 | 0.35 | 0.8 |
| blue | 0.0 | 0.0 | 1.0 | bluemagenta | 0.5 | 0.0 | 1.0 |
| midblue | 0.65 | 0.65 | 1.0 | darkviolet | 0.6 | 0.0 | 0.8 |
| lightblue | 0.8 | 0.8 | 1.0 | magenta | 1.0 | 0.0 | 1.0 |
| yellow | 1.0 | 1.0 | 0.0 | lavender | 0.8 | 0.8 | 1.0 |
| yellowgreen | 0.5 | 1.0 | 0.0 | | | | |

**COLORS**
(cont)

Background dependent colors:

| *Name* | *Black Backgnd* | | | *White Backgnd* | | | *Description* |
|---|---|---|---|---|---|---|---|
| title0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | information box |
| title1 | 1.0 | 0.7 | 0.7 | 0.1 | 0.0 | 0.0 | first info line |
| title2 | 0.0 | 0.95 | 0.8 | 0.0 | 0.5 | 0.2 | second info line |
| axis | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | axis and labels |
| roads0 | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 | roads (filled) |
| roads1 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | roads (outline) |
| land | 0.0 | 0.5 | 0.0 | 0.8 | 1.0 | 0.8 | land |
| water | 0.0 | 0.0 | 0.5 | 0.8 | 0.8 | 1.0 | water |
| bound0 | 0.0 | 0.3 | 0.0 | 0.45 | 1.0 | 0.45 | map lines (filled) |
| bound1 | 0.0 | 0.5 | 0.0 | 0.6 | 1.0 | 0.6 | map lines (outline) |
| shadow0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | map shadow (filled) |
| shadow1 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | map shadow (outline) |

If **CFRAME_A**, **CFRAME_B** and **CFRAME_C** indicate multiple backgrounds and **COLORS** is not specified for a background, then the **COLORS** for the first background will be used for the unspecified background (i.e. in most cases you only need to specify **COLORS** for the first background).

**TVAR**
**ZVAR**
**YVAR**
**XVAR**
character array[†]

Character strings that specify the orientation, location, and size of the two-dimensional slab to be plotted or 3 dimensional field to be extracted. Each is a slash delimited text string with two components, the second of which has further colon delimiting.

The first component is a single character that describes how this direction will appear on the background and is only relevant if plotting. The characters and their meanings are described as follows:

- H : direction will be plotted horizontally
- V : direction will be plotted vertically
- F : fix the direction to the following value

Note that the option of plotting variables with respect to time is not yet implemented.

The second component specifies the range information for the directions and time and is applicable to all types of ***REVU*** runs. The syntax is:

```
left_value : right_value : increment
```

where the values, if positive, are actual grid point values. If the left or right values are negative or zero, they are interpreted as an offset from the boundaries of the grid. For the time direction, the values refer to the analysis file number found according to the filtering of **ANPREF**. If **CFRAME_A**, **CFRAME_B** and **CFRAME_C** indicate multiple backgrounds and **TVAR** indicate multiple times, then the settings for each background apply to each time. If plotting, the increment is only applied to T in **TVAR**. If the increment is not specified, it defaults to 1.

**TVAR**
**ZVAR**
**YVAR**
**XVAR**
(cont)

For example, a series of vertical Y-Z slabs may be specified as:

```
TVAR(1)='/F/1:5:1/',
ZVAR(1)='/V/0:0:1/',
YVAR(1)='/F/8:8:1/',
XVAR(1)='/H/0:0:1/',
```

This means that this background will have X horizontally on the plots and Z vertical (both will have the full domain of grid points for the grid specified by **IGRID** above). The Y direction will be fixed to the 8th grid point from the southern edge of the domain and time T will range from the 1st to the 5th file found after filtering with **ANPREF**.

Or a series of horizontal X-Y slabs of may be specified as:

```
TVAR(1)='/F/ 1:25:2/',
ZVAR(1)='/F/ 2: 2:1/',
YVAR(1)='/V/-1:-2:1/',
XVAR(1)='/H/ 3:15:1/',
```

This means that this background will have X horizontally on the plots and Y vertical. For the grid specified by **IGRID** X will span from the 3rd to the 15th grid point from the western edge of the domain. Y will span from the 2nd grid point from the southern edge of the domain to the 3rd grid point from the northern edge of the domain. Time T will range from the 1st to the 25th file found after filtering with **ANPREF**, skipping every 2nd file (so 13 plots will be drawn).

If not plotting, the following settings may be appropriate (noting that the F, V and H values are ignored):

```
TVAR(1)='/F/ 1:25:1/',
ZVAR(1)='/F/ 2: 0:1/',
YVAR(1)='/V/-1:-1:2/',
XVAR(1)='/H/-1:-1:2/',
```

Which will output all 25 times for a horizontal (X-Y) domain that does not include the boundary grid points for the grid specified by **IGRID** and skips every other grid point in between, and includes all vertical levels other that the lowest which is below the terrain surface.

If **CFRAME_A**, **CFRAME_B** and **CFRAME_C** indicate multiple backgrounds and **TVAR**, **ZVAR**, **YVAR** and **XVAR** are not specified for a background, then the settings for the first background will be used for the unspecified background (i.e. in most cases you only need to specify them for the first background).

***Note**: RAMS v5.x has an option to specify different output frequencies for different grids. Therefore, all grids may not be available at all times. TVAR must be specified according to the entire set of file times that were generated for all grids, not just the grid that is being requested. Or to put this another way, TVAR is specified according to the header files (*-head.txt) as opposed to a grid's data files (*-gx.vfm).*

*For example, assume you made a 2 grid RAMS run for 6 hours and output grid 1 every hour and grid 2 every half hour. You would have a total of 13 header files. If you want to plot each hour of grid 1, you would set TVAR='/1:13:2/'. Note that in this case, this is also equivalent to TVAR='/1:13:1/', since any grid that is not present at a requested time will be skipped.*

**CFRAME_A**
character array[‡]

**CFRAME_A** specifies the variables to plot if **ANATYPE** is set 'SPACE', or extract for other **ANATYPE** settings. All available variables are detailed below (default is 'none').

If not plotting, only the variable is required. For example, a set of **CFRAME_A** settings for **ANATYPE** = 'V5D' might be:

```
CFRAME_A(1)='/u/',
CFRAME_A(2)='/v/',
CFRAME_A(3)='/w/',
CFRAME_A(4)='/tempk/',
CFRAME_A(5)='/dewptk/',
```

If plotting, a series of slash-delimited components containing plotting options can follow the variable. These control the way in which the variable is represented in the plot. Order is important and, although you can leave components at the end of the list unspecified, you must specify each component (at least with placeholders) up to the last component you wish to specify (an example of this is shown below). Options that are left unspecified are given sensible defaults. Note that the subscript of **CFRAME_A** denotes the background number (equals the frame number if **IPANEL** is set to 0). The subscript on **CFRAME_A** is tied to those on **CFRAME_B** and **CFRAME_C** in that sets of these parameters with the same subscript appear on the same plots. Describing each slash delimited component following the variable:

2. A 1 or 2 character string that defines the method of plotting (first character), and whether or not to include a color scale bar (second character) if the viable is to be plotted with a range of colors. First character setting may be:
   - **c** : draw contour lines
   - **f** : draw and fill contours
   - **t** : fill tiles

   Second character settings may be:
   - **n** or blank : no color bar
   - **b** : draw color bar

   For example:

   ```
   CFRAME_A(n)='/tempc/fb/',
   ```

   will result in a filled contour plot of temperature in degrees centigrade and include color bar (default is **fb**).
3. Minimum contour level for the field named in component 1. A value of zero allows the code to choose its own minimum.
4. Maximum contour level for the field named in component 1. A value of zero allows the code to choose its own maximum.

**CFRAME_A** (cont)

5. If positive, this is the contour interval for the field named in component 1. If negative, then it indicates about how many contour lines will be drawn between the lowest and highest values. If set to zero, a value will be chosen in the code.

   For example:

   ```
   CFRAME_A(n)='/tempc/fb/10.0/30.0/2.0/',
   ```

   will set the contour levels for temperature from 10.0 to 30.0 degrees with a contouring interval of 2.0 degrees centigrade. If values exist outside of this range they will not be contoured.

6. Options for plotting the field named in component 1. This is a string of colon-delimited options. Only those options you wish to specify need be included. An array of examples follow the definitions:

   - **m** : Graduated color method (integer). There are a variety of ways to specify the color increments with contour or tile values (all methods interpolate between colors in HLS space):

     **m0** : Default color table (mixture of predefined and easily distinguished colors).

     *Single color schemes:*

     **m1** : Single color scheme with component 7 (first specified color) at the maximum slab value, fading to the background color at the minimum slab value.

     **m2** : Single color scheme that is the same as **m1**, but fades to the foreground color.

     **m3** : Single color scheme with component 7 (first specified color) at the middle value, lightening above to white at the maximum slab value and darkening to black at the minimum slab value.

     *Two color schemes:*

     **m4** : Two color scheme with component 7 (first specified color) covering slab values greater than the central color value (specified in option '**c**') and component 8 (second specified color) covering slab values less than the central color value. The maximum saturation for the specified colors occurs at the maximum and minimum slab values, fading to the background color at the interface (the central color value) (default).

     **m5** : Two color scheme that is the same as **m4**, but fades to the foreground color.

     **m6** : Two color blending scheme with component 7 (first specified color) at the maximum slab value and component 8 (second specified color) at the minimum slab value. Colors are blended between the maximum and minimum without going through either the foreground or background colors. This can lead to some odd color ranges if the user is not careful with the bounding color selections.

**CFRAME_A** (cont)

*Three color schemes:*

**m7** : Three color blending scheme with component 7 (first specified color) at the maximum slab value, component 8 (second specified color) at the minimum slab value and component 9 (third specified color) at the central slab value. Colors are blended between the maximum and minimum and the center without going through either the foreground or background colors. This can lead to some odd color ranges if the user is not careful with the bounding color selections.

Negative values for m switches the order of color graduation (what appeared at the maximum values appear at the minimum values and vise versa).

- **b** : Graduated color bias (real) that gives a departure from linear the interpolation between colors since bias towards lighter colors is preferred as this gives more intervals in the human eyes sensitive range (e.g. b1. = linear, b1.5 = lighten and b.75 = darken) (default is 1.5).
- **c** : Central color value (real) for splitting colors and contour lines in the scheme described above (default = 0.).
- **i** : Major contour line interval (integer) (default is 5).
- **g** : Graduated contour line color offset (real).

  **>**0.0 : Graduate contour line colors> the value is used for the lightness color offset (default is 30).

  0.0 : Use component 7 (first specified color) and component 8 (second specified color) for positive and negative contours respectively.

  <0.0 : Use contour and high/low labels color parameter (option **x**) for the contour line color.
- **t** : Contour line thickness (real). t1.0 is the standard line thickness. Major contour lines are drawn at 1.5*t (default is 1.0)
- **d** : Dashed / solid line control (integer)

  **d0** : Solid lines for all contours.

  **d1** : Dashed lines for contours with values less that the central color values (option **c**) and solid lines for all other contours (default).

  **d2** : Dashed lines for all contours.
- **o** : Overlay lines on top of any filled contours and tiles (integer).

  **o0** : Do not overlay contour lines.

  **o1** : Overlay contour lines (default)
- **l** : Contour line label controls (integer).

  **l-1**: No labels.

  **l0** : Draw labels with the color defined in **x** below, but without filling a background behind the label or outlining the labels with boxes (default).

  **l1** : Draw labels in box outlined with the foreground color.

  **l2** : Draw labels in box filled with the background color.

  **l3** : Draw labels in box outlined with the foreground color and led with the background color.

**CFRAME_A** (cont)

- **h** : High and low label controls (integer). The options for h are identical to those for '**l**'.
- **x** : Color of contour and high/low labels (character), and contour lines if option **g** is set to less that zero (default is lightgray).
- **s** : Size of the contour and high/low labels (real) (default is 0.01).

7. First specified color used for color contours and fills (character). Use of this color is determined by the **m** and **g** options above (default is red).
8. Second specified color used for color contours and fills (character). Use of this color is determined by the **m** and **g** options above (default is blue).
9. Third specified color used for color contours and fills (character). Use of this color is determined by the **m** and **g** options above (default is green).

The default setting for CFRAME_A is:

```
CFRAME_A(1)='/none/fb/0./0./0./m4:c0.:b1.5:i5:d1:g30.:
             t1.:o1:l0:h0:s.01:xlightgray/red/blue/green/'
```

***Examples:***

```
CFRAME_A(1)='/tempc/fb/10./30./2.0/m5:c20.:xgray/red/blue/'
```

will fill contours of temperature from 10 to 20 degrees centigrade at 2 degree intervals with the 28 to 30 degree area filled in red and the 10 to 12 degree area in blue. The color fill dividing line is set to 20 degrees, so on a black background with color scheme 5, the two areas either side to 20 degrees (18 to 20 and 20 to 22) will be filled in white. The colors will graduate from white to red above 20 and white to blue below. The contour lines will follow the same color pattern, but each line will have a lightness offset of 30% from the fill color. Contour line and high/low labels will be drawn in gray without boxes or fill. A color scale bar will also be drawn.

```
CFRAME_A(2)='/relhum/f/50./100./5./m1:g20.:h-1:xblue/midblue/'
```

will fill contours of relative humidity from 50 to 100% at 5% intervals. The 95 to 100% interval will be filled with midblue. From there the fill colors will fade to the background color in the 50 to 55% interval. Contour lines will be drawn with a 20% lightness offset from the adjacent fill color and contour line labels will be drawn in blue. No high/low labels or color scale bar will be drawn.

```
CFRAME_A(3)='/relhum/c////c50.:g-1.:h-1/yellow/purple/'
```

will draw contours of relative humidity from the minimum value in the plotted slab to the maximum value with some convenient interval (still centered around a central color value of 50%). With the central color set at 50%, yellow contour lines of the same color will be drawn above 50% and purple below. Not high/low labels or color fill scale bar will be drawn. Note that the /'s after the /c/ are placeholders for the contour range and interval settings. They will take on the default values. The placeholders are required as order and occurrence of each component in the slash-delimited string is important.

| | |
|---|---|
| **CFRAME_B**<br>character array[‡] | If plotting, **CFRAME_B** specifies what wind information should be plotted. It is a slash-delimited string with up to 10 components. Order is important and, although you can leave components at the end of the list unspecified, you must specify each component (at least with placeholders) up to the last component you wish to specify (an example of this is in **CFRAME_A**). Options that are left unspecified are given sensible defaults. The subscript on **CFRAME_B** is tied to those on **CFRAME_A** and **CFRAME_C** in that sets of these parameters with the same subscript appear on the same plots. Describing each slash delimited component:<br>1. Character value to denote how to depict the wind field:<br>**n** : none (default)<br>**s** : draw streamlines.<br>**b** : draw wind barbs.<br>**v** : draw wind vectors.<br>**t** : draw turbulence with vectors.<br>**r** : draw relative vorticity vectors.<br>2. Variable indicating the number of horizontal grid points between adjacent wind vector or barb plots (integer ***intindi***) (default is 1).<br>3. Variable indicating the number of vertical grid points between adjacent wind vector or barb plots (integer ***intindj***) (default is 1).<br>4. Variable specifying the wind barb flags, staff and half staff values (integer) (default is (not used for vectors, but place holder still required):<br>**1** : 50, 10, 5 m/s<br>**2** : 20, 4, 2 m/s<br>**3** : 10, 2, 1 m/s (default)<br>**4** : 5, 1, 0.5 m/s<br>5. Wind marker color (test)<br>6. Stem length for vector or barb (scales the whole barb) (real) (default is ***intindi***). The default behavior for the size of the vectors and barbs is to double in size when ***intindi*** is doubled, in which case they tend to dominate the picture. You can prevent this by halving the stem length when ***intindi*** is doubled.<br>7. Maximum wind vector scale (real) (default is –1. which get recomputed to the maximum wind speed).<br>8. Wind vector head length (real) (default is –1. which get recomputed to 0.1* ***intindi***).<br>9. Wind vector head length for 0 wind speed (real) (default is –1. which get recomputed to 0.2* ***intindi***).<br>Wind vector head angle (real) (default is 25.0 degrees). |

| | |
|---|---|
| **CFRAME_B** (cont) | The default setting for **CFRAME_B** is:<br>`CFRAME_B(1)='/n/1/1/3/yellow/1./0./-1./-1./-1./'`<br>Examples:<br>`CFRAME_B(1)='/b/2/2/4/green/.5/ '`<br>will draw green wind barbs on every other grid point in both the horizontal and vertical with flag, staff and half staff values of 5, 1 and 0.5 m/s respectively. The size of the barbs will be maintained the same as for drawing them at every grid point with the 0.5 stem length setting. |
| **CFRAME_C** character array[†] | If plotting, **CFRAME_C** is a second variable that can be contoured or tiled. It can also be used to color wind vectors with or without a color scale bar (second component setting of 'vb' or 'vn'). The subscript on **CFRAME_C** is tied to those on **CFRAME_A** and **CFRAME_B** in that sets of these parameters with the same subscript appear on the same plots.<br>The default setting for **CFRAME_C** is:<br>`CFRAME_C(1)='/none/cn/0./0./0./m2:c0.:b1.5:i5:d1:g30.:`<br>`t1.:o1:l0:h0:s.01:xlightgray/darkgreen/purple/'` |

[†] Each array may contain entries for each background. In general, if array elements for a background are missing, the first array element is used. If nothing is specified, the default value is used. If nothing is specified in CFRAME_A, CFRAME_B and CFRAME_C for a background, nothing is plotted on that background.

## *$GRAB Namelist*

Required only for run types **ANATYPE** = 'GRAB' and described the I/O of the grabbed data.

| | |
|---|---|
| **GRABIN** character | Specifies the filename that contains the locations where we want to grab the data. Each latitude-longitude location can be specified as either a point in space (requires a height AGL to be given), or a profile of the model levels. The output data for each latitude-longitude-height location is then interpolated from the surrounding ***RAMS*** grid points on the finest possible grid specified by **IGRID**. The format of this file is specified below. |
| **IGRABFMT** integer | Specifies the output data format:<br>• If set to 1, output selected **CFRAME_A** variables in an ASCI file (format is specified below).<br>• If set to 2, output the data in the RALPH2 ASCI format (requires that **CFRAME_A** be set, in order, to the following fields 'u', 'v', 'tempc', 'dewptc' and 'press'). |

## *$GLL Namelist*

Required only for run types **ANATYPE** = 'GRIB', 'V5D' or 'GrADS' and described the latitude-longitude grid that the output data is interpolated to.

| | |
|---|---|
| **IGRIDLL**<br>integer | For output that is interpolated to a latitude-longitude grid (e.g. GRIB, GrADS and Vis5D), this specifies the method by which the size and resolution of the output grid are determined.<br>• If set to 0, the remaining variables in this namelist specify the size and resolution of the latitude-longitude grid.<br>• If set to 1, the maximum sized latitude-longitude grid that fits within the RAMS grid specified by **IGRID** is determined in the code.<br>• If set to 2, the minimum sized latitude-longitude grid that contains the RAMS grid specified by **IGRID** is determined in the code. |
| **GLLDLLAT**<br>real | If **IGRIDLL** = 0, specifies the latitude grid spacing of the latitude-longitude grid. |
| **GLLDLLON**<br>real | If **IGRIDLL** = 0, specifies the longitude grid spacing of the latitude-longitude grid. |
| **GLLWLON**<br>real | If **IGRIDLL** = 0, specifies the western edge of the latitude-longitude grid |
| **GLLELON**<br>real | If **IGRIDLL** = 0, specifies the eastern edge of the latitude-longitude grid |
| **GLLSLAT**<br>real | If **IGRIDLL** = 0, specifies the southern edge of the latitude-longitude grid |
| **GLLNLAT**<br>real | If **IGRIDLL** = 0, specifies the northern edge of the latitude-longitude grid |

## *$STATS Namelist*

Required only for run types **ANATYPE** = 'STATS' and describes the run mode and I/O.

| | |
|---|---|
| **CMODE**<br>character | Specifies the observations to process:<br>• If set to 'all', process both surface and upper air observations.<br>• If set to 'sfc', process surface only.<br>• If set to 'rawin', process upper air observations only. |
| **SFCPREF**<br>character | Prefix for the surface observational file names (file must be in RALPH2 format). |
| **RWNPREF**<br>character | Prefix for the upper air observational file names (file must be in RALPH2 format). |
| **NOQ**<br>character | Quality control suffix for the file names. -**NOQ** is tagged to the original data file if quality control has been run on it. |
| **IPDIFF**<br>**IPVALS**<br>**IPWIND**<br>**IPHIST**<br>**IPCONT**<br>integer | Switches for plotting statistical results (not fully functional):<br>• If set to 0, switch is off.<br>• If set to 1, switch is on.<br>**IPDIFF** specifies whether to plot predicted-observed difference values.<br>**IPVALS** specifies whether to plot actual predicted and observed values.<br>**IPWIND** specifies whether to plot winds comparison.<br>**IPHIST** specifies whether to plot histograms of predicted-observed.<br>**IPCONT** specifies whether to plot contours of predicted-observed differences. |

# *REVU Variables for Plotting or Output*

The list of variables that are currently configured in ***REVU*** version 2.5 (for ***RAMS*** v5.x) is described below. These may be selected for plotting or output into the GrADS, GRIB or Vis5D files (except for the LEAF patch variables).

The field name is what you may use in **CFRAME_A** or **CFRAME_C** in your *REVU_IN* file. Note that any particular field will require one or more model variables to be present on the analysis files. Therefore, if particular model options are not selected, you cannot plot a particular field. For example, in order to plot *totpcp*, the run must be made with microphysics activated.

## *3-D Atmospheric Variables*

The following variables are defined on the 3-D atmospheric grid and may be plotted in either horizontal or vertical cross section. Obviously, many of these variables are dependent on which options were activated for a particular run.

| ***Field Name*** | ***Description [units]*** |
|---|---|
| *u* | x-direction wind component [m/s] |
| *v* | y-direction wind component [m/s] |
| *u_avg* | eastward wind component averaged to T point [m/s] |
| *u_avg* | northward wind component averaged to T point [m/s] |
| *ue* | earth rotated eastward wind component [m/s] |
| *ve* | earth rotated northward wind component [m/s] |
| *ue_avg* | eastward wind component earth rotated and averaged to T point [m/s] |
| *ve_avg* | northward wind component earth rotated averaged to T point [m/s] |
| *w* | z-direction wind component [m/s] |
| *wcms* | z-direction wind component [cm/s] |
| *w_avg* | z-direction wind component averaged to T point [m/s] |
| *speed* | horizontal wind speed averaged to T point [m/s] |
| *speed_mph* | horizontal wind speed averaged to T point [mph] |
| *direction* | horizontal wind direction averaged to T point [deg] |
| *relvortx* | x-component of relative vorticity [rad/s] |
| *relvorty* | y-component of relative vorticity [rad/s] |
| *relvortz* | z-component of relative vorticity [rad/s] |
| *absvortz* | z-component of absolute vorticity [rad/s] |
| ***Field Name*** | ***Description [units]*** |
| *potvortz* | z-component of potential vorticity [rad/s] |
| *horiz_div* | horizontal divergence [s^-1] |
| *pi* | Exner function [J/(kg K)] |
| *press* | pressure [mb] |
| *theta* | potential temperature [K] |

| | |
|---|---|
| *dn0* | reference state density [kg/m^3] |
| *pi0* | reference state Exner function [J/(kg K)] |
| *th0* | reference state virtual potential temperature [K] |
| *pert_pressure* | perturbation pressure [mb] |
| *tempk* | temperature [K] |
| *tempc* | temperature [deg C] |
| *tempf* | temperature [deg F] |
| *theta_e* | equivalent potential temperature [K] |
| *theta_v* | virtual potential temperature [K] |
| *vapor* | water vapor mixing ratio [g/kg] |
| *cloud* | cloud water mixing ratio [g/kg] |
| *rain* | rain mixing ratio [g/kg] |
| *pristine* | pristine ice mixing ratio [g/kg] |
| *snow* | snow mixing ratio [g/kg] |
| *aggregates* | aggregates mixing ratio [g/kg] |
| *graupel* | graupel mixing ratio [g/kg] |
| *hail* | hail mixing ratio [g/kg] |
| *liquid* | liquid water mixing ratio [g/kg] |
| *ice* | ice mixing ratio [g/kg] |
| *total_cond* | total condensate mixing ratio [g/kg] |
| *r_total* | total water mixing ratio [g/kg] |
| *rtotal_orig* | total water mixing ratio (original method) [g/kg] |
| *dewptk* | dew point temperature [K] |
| *dewptf* | dew point temperature [deg F] |
| *dewptc* | dew point temperature [deg C] |
| *relhum* | relative humidity [percent] |
| *relhum_frac* | relative humidity [fraction] |
| *cloud_concen_mg* | cloud droplet number concentration [#/mg] |
| *rain_concen_kg* | rain number concentration [#/kg] |
| *pris_concen_kg* | pristine ice number concentration [#/kg] |
| *snow_concen_kg* | snow number concentration [#/kg] |
| *agg_concen_kg* | aggregates number concentration [#/kg] |
| *graup_concen_kg* | graupel number concentration [#/kg] |
| *hail_concen_kg* | hail number concentration [#/kg] |
| *cloud_concen_cm3* | cloud droplet number concentration [#/cm^3] |
| *rain_concen_m3* | rain number concentration [#/m^3] |
| ***Field Name*** | ***Description [units]*** |
| *pris_concen_m3* | pristine ice number concentration [#/m^3] |
| *snow_concen_m3* | snow number concentration [#/m^3] |
| *agg_concen_m3* | aggregates number concentration [#/m^3] |
| *graup_concen_m3* | graupel number concentration [#/m^3] |
| *hail_concen_m3* | hail number concentration [#/m^3] |
| *ccn_concen* | CCN number concentration [#/mg] |
| *ifn_conc* | IFN number concentration [#/kg] |

| | |
|---|---|
| *cloud_diam* | cloud droplet mean-mass diameter [microns] |
| *rain_diam* | rain mean-mass diameter [mm] |
| *pris_diam* | pristine ice mean-mass diameter [microns] |
| *snow_diam* | snow mean-mass diameter [mm] |
| *agg_diam* | aggregates mean-mass diameter [mm] |
| *graup_diam* | graupel mean-mass diameter [mm] |
| *hail_diam* | hail mean-mass diameter [mm] |
| *q2* | rain internal energy parameter [J/kg] |
| *q6* | graupel internal energy parameter [J/kg] |
| *q7* | hail internal energy parameter [J/kg] |
| *rain_temp* | rain temperature [deg C] |
| *graup_temp* | graupel temperature [deg C] |
| *hail_temp* | hail temperature [deg C] |
| *rain_air_tempdif* | rain-air temperature difference [K] |
| *graup_air_tempdf* | graupel-air temperature difference [K] |
| *hail_air_tempdif* | hail-air temperature difference [K] |
| *graup_fracliq* | liquid fraction in graupel [ ] |
| *hail_fracliq* | liquid fraction in hail [ ] |
| *geo* | geopotential height [m] |
| *tke* | turbulent kinetic energy [m^2/s^2] |
| *scalar1* | scalar # 1 mixing ratio [units/kg] |
| *scalar2* | scalar # 2 mixing ratio [units/kg] |
| *cuparm_thetasrc* | cumulus parameterization heating rate {K/s] |
| *cuparm_rtsrc* | cumulus parameterization moistening rate [kg/kg/s] |
| *rad_thetasrc* | radiative transfer heating rate [K/s] |
| *khh* | horizontal scalar mixing coefficient [m^2/s] |
| *khv* | vertical scalar mixing coefficient [m^2/s] |

## 2-D Variables

The following variables are defined as a function of horizontal coordinates only and may only be plotted in horizontal cross section.

| *Field Name* | *Description [units]* |
|---|---|
| *tempf2m* | 2-meter-height air temperature [deg F.] |
| *speed10m* | 10-meter-height wind speed [m/s] |
| *clear_frac* | algorithmic clear sky fraction [fraction] |
| *cloud_frac* | algorithmic cloud cover fraction [fraction] |
| *pbl_ht* | planetary boundary layer height [m] |
| *accpr* | surface accumulated rain [kg/m$^2$] |
| *accpp* | surface accumulated pristine ice [kg/m$^2$] |
| *accps* | surface accumulated snow [kg/m$^2$] |
| *accpa* | surface accumulated aggregates [kg/m$^2$] |
| *accpg* | surface accumulated graupel [kg/m$^2$] |
| *accph* | surface accumulated hail [kg/m$^2$] |
| *totpcp* | surface accumulated resolved precipitation [mm liquid equivalent] |
| *totpcp_in* | surface accumulated resolved precipitation [inches liquid equivalent] |
| *precip* | surface accumulated resolved plus convective precipitation [mm liquid equivalent] |
| *precip_in* | surface accumulated resolved plus convective precipitation [inches liquid equivalent] |
| *pcprr* | surface precipitation rate of rain [mm/hr liquid equivalent] |
| *pcprp* | surface precipitation rate of pristine ice [mm/hr liquid equivalent] |
| *psprs* | surface precipitation rate of snow [mm/hr liquid equivalent] |
| *pcpra* | surface precipitation rate of aggregates [mm/hr liquid equivalent] |
| *pcprg* | surface precipitation rate of graupel [mm/hr liquid equivalent] |
| *pcprh* | surface precipitation rate of hail [mm/hr liquid equivalent] |
| *pcpg* | total surface precipitation falling this timestep [kg/m^2] |
| *qpcpg* | total internal energy of surface precipitation falling this timestep [J/m$^2$] |
| *dpcpg* | total added depth of surface precipitation falling this timestep [m] |
| ***Field Name*** | ***Description [units]*** |
| *pcprate* | resolved surface precipitation [mm/hr liquid equivalent] |
| *pcprate_in* | resolved surface precipitation [inches/hr liquid equivalent] |
| *precipr* | resolved plus convective surface precipitation [mm/hr liquid equivalent] |
| *precipr_in* | resolved plus convective surface precipitation [inches/hr liquid equivalent] |
| *conpcp* | cumulus parameterization precipitation rate [mm/hr] |
| *acccon* | cumulus parameterization accumulated surface precipitation [mm] |
| *vertint_rt* | vertically-integrated total water mixing ratio [mm liquid equivalent] |
| *vertint_cond* | vertically-integrated total condensate mixing ratio [mm liquid equivalent] |

| | |
|---|---|
| *tfz* | surface sensible heat flux [K m/s] |
| *qfz* | surface water vapor flux [kg/kg m/s] |
| *uw* | surface x-component momentum flux [$m^2/s^2$] |
| *vw* | surface y-component momentum flux [$m^2/s^2$] |
| *wfz* | surface y-component momentum flux [$m^2/s^2$] |
| *sens_flux* | surface sensible heat flux [$W/m^2$] |
| *lat_flux* | surface latent heat flux [$W/m^2$] |
| *etrans* | evapotranspiration rate [mm/hr] |
| *etrans_in* | evapotranspiration rate [in/hr] |
| *umom_flx* | surface x-component momentum flux [Pa] |
| *vmom_flx* | surface y-component momentum flux [Pa] |
| *wmom_flx* | surface x-component momentum flux [Pa] |
| *bowen* | Bowen ratio [ ] |
| *rshort* | incident surface flux of shortwave radiation [$W/m^2$] |
| *rlong* | incident surface flux of longwave radiation [$W/m^2$] |
| *rlongup* | upward surface flux of longwave radiation [$W/m^2$] |
| *albedt* | grid-cell-averaged surface albedo [ ] |
| *topo* | topography height [m] |
| *topo_ft* | topography height [ft] |
| *lat* | latitude [deg] |
| *lon* | longitude [deg] |
| *sea_press* | sea level pressure [mb] |
| *sfc_div* | horizontal divergence at surface [s^-1] |
| *land* | land fractional area [ ] |
| *ctprof* | cloud top height [m] |
| *sst* | water temperature [deg C] |

## LEAF Variables

These variables are from the LEAF2 parameterization and comprise all the soil and vegetation parameters.

The following variables are defined in ***RAMS*** for land subgrid patches only. They are averaged over all land patches and are defined for the entire grid cell regardless of how little area of the grid cell is occupied by land. *In grid cells that contain no land, these field values will retain their initial values in the RAMS simulation.* These fields may be plotted only in horizontal cross section.

| ***Field Name*** | ***Description [units]*** |
|---|---|
| *soil_z0_ps* | soil roughness height by grid cell [m] |
| *veg_fracarea_ps* | vegetation fractional area by grid cell [ ] |
| *veg_lai_ps* | vegetation leaf area index by grid cell [ ] |
| *veg_z0_ps* | vegetation roughness height by grid cell [m] |
| *veg_disp_ps* | vegetation displacement height by grid cell [m] |
| *grnd_mixrat_ps* | ground mixing ratio by grid cell [g/kg] |
| *soil_mixrat_ps* | soil mixing ratio by grid cell [g/kg] |
| *veg_moist_ps* | vegetation surface moisture by grid cell [kg/m^2] |
| *veg_temp_ps* | vegetation temperature by grid cell [deg C] |
| *snow_depth_ps* | snow depth by grid cell [m] |
| *snowcover_ps* | snowcover content by grid cell [kg/m^2] |

The following variables are defined in ***RAMS*** for all subgrid patches. They are averaged over all patches here. They may be plotted only in horizontal cross section.

| ***Field Name*** | ***Description [units]*** |
|---|---|
| *net_z0_ps* | net roughness height by grid cell [m] |
| *canopy_mixrat_ps* | canopy mixing ratio by grid cell [g/kg] |
| *canopy_temp_ps* | canopy temperature by grid cell [deg C] |
| *ustar_ps* | ustar by grid cell [m/s] |
| *tstar_ps* | tstar by grid cell [K] |
| *rstar_ps* | rstar by grid cell [kg/kg] |
| *sens_heat_flux_ps* | surface sensible heat flux by grid cell [W/m^2] |
| *lat_heat_flux_ps* | surface latent heat flux by grid cell [W/m^2] |
| *5050_temp_ps* | average of k = 2 air and canopy air temperatures by grid cell [deg C] |
| *5050_tempf_ps* | average of k = 2 air and canopy air temperatures by grid cell [deg F] |

The following variables are defined in ***RAMS*** for all grid cells, but they are not quantities that can be averaged. A single value is taken here for the entire grid cell from the subgrid patch having the largest area in the cell. They may be plotted only in horizontal cross section.

| ***Field Name*** | ***Description [units]*** |
|---|---|
| *veg_class_bp* | vegetation class of the biggest patch in the grid cell [ ] |
| *qveg_class_bp* | "q" vegetation class of the biggest patch in the grid cell [ ] |

The following variables are defined in ***RAMS*** for subgrid patches and remain defined by patch here. They may be plotted only in horizontal cross section and only with the tile plot function. *These variables cannot be output to GrADS, GRIB or Vis5D.*

| ***Field Name*** | ***Description [units]*** |
|---|---|
| *patch_area* | fractional area of patches [ ] |
| *soil_z0_p* | soil roughness height by patch [m] |
| *veg_class_p* | vegetation class by patch [ ] |
| *qveg_class_p* | "q" vegetation class by patch |
| *veg_fracarea_p* | vegetation fractional area by patch [ ] |
| *veg_lai_p* | vegetation leaf area index by patch [ ] |
| *net_z0_p* | net roughness height by patch [m] |
| *veg_z0_p* | vegetation roughness height by patch [m] |
| *veg_disp_p* | vegetation displacement height by patch [m] |
| *patch_wetind* | wetness index by patch [ ] |
| *snowlevels* | number of snow levels by patch [ ] |
| *grnd_mixrat_p* | ground mixing ratio by patch [g/kg] |
| *soil_mixrat_p* | soil mixing ratio by patch [g/kg] |
| *veg_moist_p* | vegetation surface moisture by patch [kg/m^2] |
| *canopy_mixrat_p* | canopy mixing ratio by patch [g/kg] |
| *veg_temp_p* | vegetation temperature by patch [deg C] |
| *canopy_temp_p* | canopy temperature by patch [deg C] |
| *ustar_p* | ustar by patch [m/s] |
| *tstar_p* | tstar by patch [K] |
| *rstar_p* | rstar by patch [kg/kg] |
| *sens_heat_flux_p* | surface sensible heat flux by patch [W/m^2] |
| *lat_heat_flux_p* | surface latent heat flux by patch [W/m^2] |
| *snow_depth_p* | snow depth by patch [m] |
| *snowcover_p* | snowcover content by patch [kg/m^2] |

The following variables are defined in ***RAMS*** for land subgrid patches only and as a function of depth. They are averaged over land patches here and are expanded to cover the entire grid regardless of how little area of the grid cell is occupied by land. *In grid cells that contain no land, these field values will retain* their initial values in the ***RAMS*** simulation. These fields may be plotted either in horizontal or vertical cross section. *These variables cannot be output to GrADS, GRIB or Vis5D.*

| ***Field Name*** | ***Description [units]*** |
|---|---|
| *sltex_bp* | soil textural class of biggest patch in grid cell [ ] |
| *soilq_ps* | soil internal energy parameter by grid cell [j/m^2] |
| *soil_temp_ps* | soil/sea temperature by grid cell [deg C] |
| *soil_moist_ps* | soil moisture by grid cell [m^3/m^3] |
| *soil_moistf_ps* | soil moisture fraction by grid cell [m^3/m^3] |

The following variables are defined in ***RAMS*** for all subgrid patches and as a function of depth. They remain defined in that way here. They may be plotted either in horizontal or vertical cross section, but only with the tile plot function. *These variables cannot be output to GrADS, GRIB or Vis5D.*

| ***Field Name*** | ***Description [units]*** |
|---|---|
| *sltex_p* | soil textural class by patch [ ] |
| *soilq_p* | soil internal energy parameter by patch [J/m^3] |
| *soil_temp_p* | soil/sea temperature by patch [deg C] |
| *soil_moist_p* | soil moisture by patch [m^3/m^3] |
| *soil_moistf_p* | soil moisture fraction by patch [m^3/m^3] |

The following variables are made up from a combination of leaf2 variables defined by both patch and depth and those defined by patch only, plus atmospheric variables from the lowest model level. They remain defined in that way here. They may be plotted only in vertical cross section, and only with the tile plot function. *These variables cannot be output to GrADS, GRIB or Vis5D.*

| ***Field Name*** | ***Description [units]*** |
|---|---|
| *leaf2_moisture* | moisture of all leaf2 components |
| *leaf2_temp* | temperature of all leaf2 components |

## HYPACT Variables

REVU will also plot the particle and concentration files from HYPACT. The concentration files are defined on the RAMS grid structure that served as the input fields for HYPACT. Therefore, both the particle and concentration plots will be plotted relative to the RAMS grids domain size and resolution. ANPREF should be specified as the prefix of the concentration files (e.g., hypout/a-C-), even if you wish to plot particles only.

The field names that can be specified in CFRAME_A and CFRAME_C are listed in the table below. The designation "*#spec*" and "*#src*" refer to an integer identifier for the species and the source that you wish to plot. The species number must be specified as an integer greater than 0; the source number may be specified as 0 to designate all sources.

If you output time-averaged files from HYPACT, use the field names with the "*avg*" designation.

| *Field Name* | *Description [units]* |
|---|---|
| *part_lag:#spec:#src* | particle depictions of a particular species |
| *conc_lag:#spec* | Lagrangian concentration (instantaneous) |
| *prob_lag:#spec* | Lagrangian probability (instantaneous) |
| *conc_lag_avg:#spec* | Lagrangian concentration (time-averaged) |
| *conc_eul:#spec* | Eulerian concentration (instantaneous) |
| *conc_eul_avg:#spec* | Eulerian concentration (time-averaged) |
| *conc_hyb:#spec* | Total concentration (instantaneous) |
| *conc_hyb_avg:#spec* | Total concentration (time-averaged) |

# *REVU Input Files*

The *REVU_IN* namelist file, required for all ***REVU*** runs, is described in full above. Two other input files are described below.

## *LANDMARKS*

If plotting a horizontal cross-section (see **ZVAR**, **XVAR**, and **YVAR**) and the **LANDMARK** parameter is set to draw landmarks, the locations available for plotting are specified in the *LANDMARKS* file which must reside in the run directory. Landmark locations are specified by a name (up to 16 characters in length) and its latitude-longitude coordinates (as either degree minute second integer qualities, or degree real quantities), noting that line beginning with # are ignored. For example:

```
# ------- NY --------------------------
ISP               40 47 38   -073 06 06
MTP               41 04 23   -071 55 24
LGA               40 46 45   -073 52 48
POU               41 37 32   -073 52 55
```

Is equivalent to:

```
# ------- NY --------------------------
ISP               40.7939    -73.1017
MTP               41.0731    -71.9233
LGA               40.7792    -73.8800
POU               41.6256    -73.8819
```

Since not all the locations specified in the *LANDMARKS* file are plotted when doing so will cause label text to overlap or the plot to be cluttered (controlled by the **LANDMARK** '**b**' component setting), those locations that appear nearer the beginning of the file may get plotted at the expense of those later in the file. Thus, to ensure you see the locations you wish to see, ensure those locations are at the top of the file, in order of drawing priority.

## *grab_in*

The **$GRAB** namelist parameter **GRABIN** by default points to the file *grab_in*. This file specifies the locations where we want to grab the data for an **ANATYPE** 'GRAB'. Each latitude-longitude location can be specified as either a point in space (requires a height AGL to be given), or a profile of the model levels. The first line contains the file format version number (integer). The remaining lines are either comments or space delimited station location data (one station per line):

```
stat lat lon elev type
```

where:

```
stat = station id number (integer)
lat  = latitude [degrees] (real)
lon  = longitude [degrees] (real)
elev = height AGL in m (real) [=0. for itype=2]
type = 1, get point data at lat,lon,elev (integer)
     = 2, get profile data at lat,lon (RAMS levels)
```

Noting that anything after a # or ! is ignored. For example:

```
1
# some test data (! and # indicate comments)
! st id   lat         lon        elev  itype
  1    40.79388889 -73.10166667   10.    1   ! point data at 10m AGL
  2    41.07305556 -71.92333333   10.    1
  3    41.07305556 -71.92333333   0.     2   ! profile data
  4    40.77916667 -73.88000000   500.   1   ! point data at 500m AGL
  5    41.62555556 -73.88194444   10.    1
```

# *REVU Output Files*

If the type of run defined by **ANATYPE** is 'SPACE', an NCAR Graphics NCGM file named *gmeta* will be output in the current directory (not the directory specified by **REVPREF**). For all other run types the file format:

**REVPREF**<analpref>[SCP]-<date>-g<grid>.<format>

where:

- <analpref> is the file prefix given by **AFILOUT** or **HYPPREF**, the ***RAMS*** and ***HYPACT*** output filename prefixes respectively, without the path component.
- [SCP] is determined by **IZTRAN**, i.e.
  S – for **IZTRAN**=1, output on sigma surfaces.
  C – for **IZTRAN**=2, output on Cartesian surfaces.
  P – for **IZTRAN**=3, output on pressure surfaces.
- <date> is the standard date string of the analysis files for the beginning of the ***RAMS*** simulation.
- <grid> is the ***RAMS*** grid number of the data. A grid number of 0 indicates that the file contains information on all grids.
- <format> is the type of output file, as determined by **ANATYPE** below, i.e.
  v5d – for **ANATYPE**=V5D, Vis5d files.
  gra – for **ANATYPE**=GRADS, GrADS files.
  grb – for **ANATYPE**=GRIB, GRIB files.
  gbr – for **ANATYPE**=GRAB, grabber files.
  dmp – for **ANATYPE**=DUMP, dumped files.

If the types of run defined by **ANATYPE** is 'GRAB' and **IGRABFMT** is 0, the selected **CFRAME_A** variables are output in an ASCI file with the following format:

```
date time istat lat lon elev x y z grd (var(i),(i=1,nvars))
```

where:

```
date = YYYYMMDD (integer, i10)
time = HHMMSS (integer, i10)
stat = station id number (integer, i10)
lat  = latitude [degrees] (real, e15.6)
lon  = longitude [degrees] (real, e15.6)
elev = height AGL [m] (real, e15.6) [=0. for itype=2]
x    = RAMS x coordinate [m] (real, e15.6)
y    = RAMS y coordinate [m]  (real, e15.6)
z    = RAMS z coordinate [m AGL] (real, e15.6)
grd  = RAMS grid used (integer, i10)
var  = variable values  (real, e15.6) [up to 30]
```

If the types of run defined by **ANATYPE** = 'GRAB' and **IGRABFMT** = 1, the selected **CFRAME_A** are output in a RALPH2 format ASCI file.